

Computational Physics: Hartree-Fock methods and introduction to Many-body Theory

Morten Hjorth-Jensen^{1,2}

¹National Superconducting Cyclotron Laboratory and Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

²Department of Physics, University of Oslo, Oslo, Norway

Spring 2017

Why Hartree-Fock?

Hartree-Fock (HF) theory is an algorithm for finding an approximative expression for the ground state of a given Hamiltonian. The basic ingredients are

- Define a single-particle basis $\{\psi_\alpha\}$ so that

$$\hat{h}^{\text{HF}}\psi_\alpha = \varepsilon_\alpha\psi_\alpha$$

with the Hartree-Fock Hamiltonian defined as

$$\hat{h}^{\text{HF}} = \hat{t} + \hat{u}_{\text{ext}} + \hat{u}^{\text{HF}}$$

- The term \hat{u}^{HF} is a single-particle potential to be determined by the HF algorithm.
- The HF algorithm means to choose \hat{u}^{HF} in order to have

$$\langle \hat{H} \rangle = E^{\text{HF}} = \langle \Phi_0 | \hat{H} | \Phi_0 \rangle$$

that is to find a local minimum with a Slater determinant Φ_0 being the ansatz for the ground state.

- The variational principle ensures that $E^{\text{HF}} \geq E_0$, with E_0 the exact ground state energy.

Why Hartree-Fock?

We will show that the Hartree-Fock Hamiltonian \hat{h}^{HF} equals our definition of the operator \hat{f} discussed in connection with the new definition of the normal-ordered Hamiltonian (see later lectures), that is we have, for a specific matrix element

$$\langle p|\hat{h}^{\text{HF}}|q\rangle = \langle p|\hat{f}|q\rangle = \langle p|\hat{t} + \hat{u}_{\text{ext}}|q\rangle + \sum_{i \leq F} \langle pi|\hat{V}|qi\rangle_{AS},$$

meaning that

$$\langle p|\hat{u}^{\text{HF}}|q\rangle = \sum_{i \leq F} \langle pi|\hat{V}|qi\rangle_{AS}.$$

The so-called Hartree-Fock potential \hat{u}^{HF} brings an explicit medium dependence due to the summation over all single-particle states below the Fermi level F . It brings also in an explicit dependence on the two-body interaction (in nuclear physics we can also have complicated three- or higher-body forces). The two-body interaction, with its contribution from the other bystanding fermions, creates an effective mean field in which a given fermion moves, in addition to the external potential \hat{u}_{ext} which confines the motion of the fermion. For systems like nuclei, there is no external confining potential. Nuclei are examples of self-bound systems, where the binding arises due to the intrinsic nature of the strong force. For nuclear systems thus, there would be no external one-body potential in the Hartree-Fock Hamiltonian.

Quantum Many-particle Methods

- Large-scale diagonalization (Iterative methods, Lanczo's method, dimensionalities 10^{10} states)
- Coupled cluster theory, favoured method in quantum chemistry, molecular and atomic physics. Applications to ab initio calculations in nuclear physics as well for large nuclei
- Perturbative many-body methods
- Density functional theories/Mean-field theory and Hartree-Fock theory
- Monte-Carlo methods (Only in FYS4411, Computational quantum mechanics)
- Green's function theories
- and other. The physics of the system hints at which many-body methods to use.

Selected Texts and Many-body theory

- Blaizot and Ripka, *Quantum Theory of Finite systems*, MIT press 1986
- Negele and Orland, *Quantum Many-Particle Systems*, Addison-Wesley, 1987.
- Fetter and Walecka, *Quantum Theory of Many-Particle Systems*, McGraw-Hill, 1971.
- Helgaker, Jorgensen and Olsen, *Molecular Electronic Structure Theory*, Wiley, 2001.
- Mattuck, *Guide to Feynman Diagrams in the Many-Body Problem*, Dover, 1971.
- Dickhoff and Van Neck, *Many-Body Theory Exposed*, World Scientific, 2006.

Definitions

An operator is defined as \hat{O} throughout. Unless otherwise specified the number of particles is always N and d is the dimension of the system. In nuclear physics we normally define the total number of particles to be $A = N + Z$, where N is total number of neutrons and Z the total number of protons. In case of other baryons such isobars Δ or various hyperons such as Λ or Σ , one needs to add their definitions. Hereafter, N is reserved for the total number of particles, unless otherwise specified.

Definitions

The quantum numbers of a single-particle state in coordinate space are defined by the variable

$$x = (\mathbf{r}, \sigma),$$

where

$$\mathbf{r} \in \mathbb{R}^d,$$

with $d = 1, 2, 3$ represents the spatial coordinates and σ is the eigenspin of the particle. For fermions with eigenspin $1/2$ this means that

$$x \in \mathbb{R}^d \oplus \left(\frac{1}{2}\right),$$

and the integral $\int dx = \sum_{\sigma} \int d^d r = \sum_{\sigma} \int d\mathbf{r}$, and

$$\int d^N x = \int dx_1 \int dx_2 \dots \int dx_N.$$

Definitions

The quantum mechanical wave function of a given state with quantum numbers λ (encompassing all quantum numbers needed to specify the system), ignoring time, is

$$\Psi_\lambda = \Psi_\lambda(x_1, x_2, \dots, x_N),$$

with $x_i = (\mathbf{r}_i, \sigma_i)$ and the projection of σ_i takes the values $\{-1/2, +1/2\}$ for particles with spin $1/2$. We will hereafter always refer to Ψ_λ as the exact wave function, and if the ground state is not degenerate we label it as

$$\Psi_0 = \Psi_0(x_1, x_2, \dots, x_N).$$

Definitions

Since the solution Ψ_λ seldomly can be found in closed form, approximations are sought. Here we define an approximative wave function or an ansatz to the exact wave function as

$$\Phi_\lambda = \Phi_\lambda(x_1, x_2, \dots, x_N),$$

with

$$\Phi_0 = \Phi_0(x_1, x_2, \dots, x_N),$$

being the ansatz to the ground state.

Definitions

The wave function Ψ_λ is sought in the Hilbert space of either symmetric or anti-symmetric N -body functions, namely

$$\Psi_\lambda \in \mathcal{H}_N := \mathcal{H}_1 \oplus \mathcal{H}_1 \oplus \dots \oplus \mathcal{H}_1,$$

where the single-particle Hilbert space $\hat{\mathcal{H}}_1$ is the space of square integrable functions over $\in \mathbb{R}^d \oplus (\sigma)$ resulting in

$$\mathcal{H}_1 := L^2(\mathbb{R}^d \oplus (\sigma)).$$

Definitions

Our Hamiltonian is invariant under the permutation (interchange) of two particles. Since we deal with fermions however, the total wave function is antisymmetric. Let \hat{P} be an operator which interchanges two particles. Due to the symmetries we have ascribed to our Hamiltonian, this operator commutes with the total Hamiltonian,

$$[\hat{H}, \hat{P}] = 0,$$

meaning that $\Psi_\lambda(x_1, x_2, \dots, x_N)$ is an eigenfunction of \hat{P} as well, that is

$$\hat{P}_{ij}\Psi_\lambda(x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_N) = \beta\Psi_\lambda(x_1, x_2, \dots, x_j, \dots, x_i, \dots, x_N),$$

where β is the eigenvalue of \hat{P} . We have introduced the suffix ij in order to indicate that we permute particles i and j . The Pauli principle tells us that the total wave function for a system of fermions has to be antisymmetric, resulting in the eigenvalue $\beta = -1$.

Definitions and notations

The Schrodinger equation reads

$$\hat{H}(x_1, x_2, \dots, x_N) \Psi_\lambda(x_1, x_2, \dots, x_N) = E_\lambda \Psi_\lambda(x_1, x_2, \dots, x_N), \quad (1)$$

where the vector x_i represents the coordinates (spatial and spin) of particle i , λ stands for all the quantum numbers needed to classify a given N -particle state and Ψ_λ is the pertaining eigenfunction. Throughout this course, Ψ refers to the exact eigenfunction, unless otherwise stated.

Definitions and notations

We write the Hamilton operator, or Hamiltonian, in a generic way

$$\hat{H} = \hat{T} + \hat{V}$$

where \hat{T} represents the kinetic energy of the system

$$\hat{T} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} = \sum_{i=1}^N \left(-\frac{\hbar^2}{2m_i} \nabla_{\mathbf{i}}^2 \right) = \sum_{i=1}^N t(x_i)$$

while the operator \hat{V} for the potential energy is given by

$$\hat{V} = \sum_{i=1}^N \hat{u}_{\text{ext}}(x_i) + \sum_{j < i=1}^N v(x_i, x_j) + \sum_{i < j < k=1}^N v(x_i, x_j, x_k) + \dots \quad (2)$$

Hereafter we use natural units, viz. $\hbar = c = e = 1$, with e the elementary charge and c the speed of light. This means that momenta and masses have dimension energy.

Definitions and notations

If one does quantum chemistry, after having introduced the Born-Oppenheimer approximation which effectively freezes out the nucleonic degrees of freedom, the Hamiltonian for $N = n_e$ electrons takes the following form

$$\hat{H} = \sum_{i=1}^{n_e} t(x_i) - \sum_{i=1}^{n_e} k \frac{Z}{r_i} + \sum_{i < j}^{n_e} \frac{k}{r_{ij}},$$

with $k = 1.44 \text{ eVnm}$

Definitions and notations

We can rewrite this as

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \sum_{i=1}^{n_e} \hat{h}_0(x_i) + \sum_{i<j}^{n_e} \frac{1}{r_{ij}}, \quad (3)$$

where we have defined

$$r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|,$$

and

$$\hat{h}_0(x_i) = \hat{t}(x_i) - \frac{Z}{x_i}. \quad (4)$$

The first term of Eq. (3), H_0 , is the sum of the N *one-body* Hamiltonians \hat{h}_0 . Each individual Hamiltonian \hat{h}_0 contains the kinetic energy operator of an electron and its potential energy due to the attraction of the nucleus. The second term, H_I , is the sum of the $n_e(n_e - 1)/2$ two-body interactions between each pair of electrons. Note that the double sum carries a restriction $i < j$.

Definitions and notations

The potential energy term due to the attraction of the nucleus defines the onebody field $u_i = u_{\text{ext}}(x_i)$ of Eq. (2). We have moved this term into the \hat{H}_0 part of the Hamiltonian, instead of keeping it in \hat{V} as in Eq. (2). The reason is that we will hereafter treat \hat{H}_0 as our non-interacting Hamiltonian. For a many-body wavefunction Φ_λ defined by an appropriate single-particle basis, we may solve exactly the non-interacting eigenvalue problem

$$\hat{H}_0 \Phi_\lambda = w_\lambda \Phi_\lambda,$$

with w_λ being the non-interacting energy. This energy is defined by the sum over single-particle energies to be defined below. For atoms the single-particle energies could be the hydrogen-like single-particle energies corrected for the charge Z . For nuclei and quantum dots, these energies could be given by the harmonic oscillator in three and two dimensions, respectively.

Definitions and notations

We will assume that the interacting part of the Hamiltonian can be approximated by a two-body interaction. This means that our Hamiltonian is written as

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \sum_{i=1}^N \hat{h}_0(x_i) + \sum_{i<j}^N V(r_{ij}), \quad (5)$$

with

$$H_0 = \sum_{i=1}^N \hat{h}_0(x_i) = \sum_{i=1}^N (\hat{t}(x_i) + \hat{u}_{\text{ext}}(x_i)). \quad (6)$$

The onebody part $u_{\text{ext}}(x_i)$ is normally approximated by a harmonic oscillator potential or the Coulomb interaction an electron feels from the nucleus. However, other potentials are fully possible, such as one derived from the self-consistent solution of the Hartree-Fock equations.

Definitions and notations

Our Hamiltonian is invariant under the permutation (interchange) of two particles. Since we deal with fermions however, the total wave function is antisymmetric. Let \hat{P} be an operator which interchanges two particles. Due to the symmetries we have ascribed to our Hamiltonian, this operator commutes with the total Hamiltonian,

$$[\hat{H}, \hat{P}] = 0,$$

meaning that $\Psi_\lambda(x_1, x_2, \dots, x_N)$ is an eigenfunction of \hat{P} as well, that is

$$\hat{P}_{ij}\Psi_\lambda(x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_N) = \beta\Psi_\lambda(x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_N),$$

where β is the eigenvalue of \hat{P} . We have introduced the suffix ij in order to indicate that we permute particles i and j . The Pauli principle tells us that the total wave function for a system of fermions has to be antisymmetric, resulting in the eigenvalue $\beta = -1$.

Definitions and notations

In our case we assume that we can approximate the exact eigenfunction with a Slater determinant

$$\Phi(x_1, x_2, \dots, x_N, \alpha, \beta, \dots, \sigma) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_\alpha(x_1) & \psi_\alpha(x_2) & \dots & \dots & \psi_\alpha(x_N) \\ \psi_\beta(x_1) & \psi_\beta(x_2) & \dots & \dots & \psi_\beta(x_N) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \psi_\sigma(x_1) & \psi_\sigma(x_2) & \dots & \dots & \psi_\sigma(x_N) \end{vmatrix}, \quad (7)$$

where x_i stand for the coordinates and spin values of a particle i and $\alpha, \beta, \dots, \gamma$ are quantum numbers needed to describe remaining quantum numbers.

Definitions and notations

The single-particle function $\psi_\alpha(x_i)$ are eigenfunctions of the onebody Hamiltonian h_i , that is

$$\hat{h}_0(x_i) = \hat{t}(x_i) + \hat{u}_{\text{ext}}(x_i),$$

with eigenvalues

$$\hat{h}_0(x_i)\psi_\alpha(x_i) = (\hat{t}(x_i) + \hat{u}_{\text{ext}}(x_i))\psi_\alpha(x_i) = \varepsilon_\alpha\psi_\alpha(x_i).$$

The energies ε_α are the so-called non-interacting single-particle energies, or unperturbed energies. The total energy is in this case the sum over all single-particle energies, if no two-body or more complicated many-body interactions are present.

Definitions and notations

Let us denote the ground state energy by E_0 . According to the variational principle we have

$$E_0 \leq E[\Phi] = \int \Phi^* \hat{H} \Phi d\tau$$

where Φ is a trial function which we assume to be normalized

$$\int \Phi^* \Phi d\tau = 1,$$

where we have used the shorthand $d\tau = d\mathbf{r}_1 d\mathbf{r}_2 \dots d\mathbf{r}_N$.

Brief reminder on some linear algebra properties

Before we proceed with a more compact representation of a Slater determinant, we would like to repeat some linear algebra properties which will be useful for our derivations of the energy as function of a Slater determinant, Hartree-Fock theory and later the nuclear shell model.

The inverse of a matrix is defined by

$$\mathbf{A}^{-1} \cdot \mathbf{A} = I$$

A unitary matrix \mathbf{A} is one whose inverse is its adjoint

$$\mathbf{A}^{-1} = \mathbf{A}^\dagger$$

A real unitary matrix is called orthogonal and its inverse is equal to its transpose. A hermitian matrix is its own self-adjoint, that is

$$\mathbf{A} = \mathbf{A}^\dagger.$$

Basic Matrix Features

Matrix Properties Reminder

Relations	Name	matrix elements
$A = A^T$	symmetric	$a_{ij} = a_{ji}$
$A = (A^T)^{-1}$	real orthogonal	$\sum_k a_{ik} a_{jk} = \sum_k a_{ki} a_{kj} = \delta_{ij}$
$A = A^*$	real matrix	$a_{ij} = a_{ij}^*$
$A = A^\dagger$	hermitian	$a_{ij} = a_{ji}^*$
$A = (A^\dagger)^{-1}$	unitary	$\sum_k a_{ik} a_{jk}^* = \sum_k a_{ki}^* a_{kj} = \delta_{ij}$

Basic Matrix Features

Since we will deal with Fermions (identical and indistinguishable particles) we will form an ansatz for a given state in terms of so-called Slater determinants determined by a chosen basis of single-particle functions.

For a given $n \times n$ matrix \mathbf{A} we can write its determinant

$$\det(\mathbf{A}) = |\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nn} \end{vmatrix},$$

in a more compact form as

$$|\mathbf{A}| = \sum_{i=1}^{n!} (-1)^{p_i} \hat{P}_i a_{11} a_{22} \dots a_{nn},$$

where \hat{P}_i is a permutation operator which permutes the column indices $1, 2, 3, \dots, n$ and the sum runs over all $n!$ permutations. The quantity p_i represents the number of transpositions of column indices that are needed in order to bring a given permutation back to its initial ordering, in our case given by $a_{11} a_{22} \dots a_{nn}$ here.

Basic Matrix Features, simple 2×2 determinant

A simple 2×2 determinant illustrates this. We have

$$\det(\mathbf{A}) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = (-1)^0 a_{11} a_{22} + (-1)^1 a_{12} a_{21},$$

where in the last term we have interchanged the column indices 1 and 2. The natural ordering we have chosen is $a_{11} a_{22}$.

Definitions and notations

With the above we can rewrite our Slater determinant in a more compact form. In the Hartree-Fock method the trial function is the Slater determinant of Eq. (7) which can be rewritten as

$$\Phi(x_1, x_2, \dots, x_N, \alpha, \beta, \dots, \nu) = \frac{1}{\sqrt{N!}} \sum_P (-1)^P \hat{P} \psi_\alpha(x_1) \psi_\beta(x_2) \dots \psi_\nu(x_N) = \sqrt{N!} \hat{A} \Phi_H,$$

where we have introduced the antisymmetrization operator \hat{A} defined by the summation over all possible permutations of two particles.

Definitions and notations

It is defined as

$$\hat{A} = \frac{1}{N!} \sum_p (-)^p \hat{P}, \quad (8)$$

with p standing for the number of permutations. We have introduced for later use the so-called Hartree-function, defined by the simple product of all possible single-particle functions

$$\Phi_H(x_1, x_2, \dots, x_N, \alpha, \beta, \dots, \nu) = \psi_\alpha(x_1) \psi_\beta(x_2) \dots \psi_\nu(x_N).$$

Definitions and notations

Both \hat{H}_0 and \hat{H}_I are invariant under all possible permutations of any two particles and hence commute with \hat{A}

$$[H_0, \hat{A}] = [H_I, \hat{A}] = 0. \quad (9)$$

Furthermore, \hat{A} satisfies

$$\hat{A}^2 = \hat{A}, \quad (10)$$

since every permutation of the Slater determinant reproduces it.

Definitions and notations

The expectation value of \hat{H}_0

$$\int \Phi^* \hat{H}_0 \Phi d\tau = N! \int \Phi_H^* \hat{A} \hat{H}_0 \hat{A} \Phi_H d\tau$$

is readily reduced to

$$\int \Phi^* \hat{H}_0 \Phi d\tau = N! \int \Phi_H^* \hat{H}_0 \hat{A} \Phi_H d\tau,$$

where we have used Eqs. (9) and (10). The next step is to replace the anti-symmetrization operator by its definition and to replace \hat{H}_0 with the sum of one-body operators

$$\int \Phi^* \hat{H}_0 \Phi d\tau = \sum_{i=1}^N \sum_p (-)^p \int \Phi_H^* \hat{h}_0 \hat{P} \Phi_H d\tau.$$

Definitions and notations

The integral vanishes if two or more particles are permuted in only one of the Hartree-functions Φ_H because the individual single-particle wave functions are orthogonal. We obtain then

$$\int \Phi^* \hat{H}_0 \Phi d\tau = \sum_{i=1}^N \int \Phi_H^* \hat{h}_0 \Phi_H d\tau.$$

Orthogonality of the single-particle functions allows us to further simplify the integral, and we arrive at the following expression for the expectation values of the sum of one-body Hamiltonians

$$\int \Phi^* \hat{H}_0 \Phi d\tau = \sum_{\mu=1}^N \int \psi_{\mu}^*(\mathbf{r}) \hat{h}_0 \psi_{\mu}(\mathbf{r}) d\mathbf{r}. \quad (11)$$

Definitions and notations

We introduce the following shorthand for the above integral

$$\langle \mu | \hat{h}_0 | \mu \rangle = \int \psi_{\mu}^*(\mathbf{r}) \hat{h}_0 \psi_{\mu}(\mathbf{r}) d\mathbf{r},$$

and rewrite Eq. (11) as

$$\int \Phi^* \hat{H}_0 \Phi d\tau = \sum_{\mu=1}^N \langle \mu | \hat{h}_0 | \mu \rangle. \quad (12)$$

Definitions and notations

The expectation value of the two-body part of the Hamiltonian is obtained in a similar manner. We have

$$\int \Phi^* \hat{H}_I \Phi d\tau = N! \int \Phi_H^* \hat{A} \hat{H}_I \hat{A} \Phi_H d\tau,$$

which reduces to

$$\int \Phi^* \hat{H}_I \Phi d\tau = \sum_{i \leq j=1}^N \sum_p (-)^p \int \Phi_H^* V(r_{ij}) \hat{P} \Phi_H d\tau,$$

by following the same arguments as for the one-body Hamiltonian.

Definitions and notations

Because of the dependence on the inter-particle distance r_{ij} , permutations of any two particles no longer vanish, and we get

$$\int \Phi^* \hat{H}_I \Phi d\tau = \sum_{i < j=1}^N \int \Phi_H^* V(r_{ij}) (1 - P_{ij}) \Phi_H d\tau.$$

where P_{ij} is the permutation operator that interchanges particle i and particle j . Again we use the assumption that the single-particle wave functions are orthogonal.

Definitions and notations

We obtain

$$\int \Phi^* \hat{H}_I \Phi d\tau = \frac{1}{2} \sum_{\mu=1}^N \sum_{\nu=1}^N \left[\int \psi_{\mu}^*(x_i) \psi_{\nu}^*(x_j) V(r_{ij}) \psi_{\mu}(x_i) \psi_{\nu}(x_j) dx_i dx_j \right. \quad (13)$$

$$\left. - \int \psi_{\mu}^*(x_i) \psi_{\nu}^*(x_j) V(r_{ij}) \psi_{\nu}(x_i) \psi_{\mu}(x_j) dx_i dx_j \right]. \quad (14)$$

The first term is the so-called direct term. It is frequently also called the Hartree term, while the second is due to the Pauli principle and is called the exchange term or just the Fock term. The factor 1/2 is introduced because we now run over all pairs twice.

Definitions and notations

The last equation allows us to introduce some further definitions. The single-particle wave functions $\psi_{\mu}(x)$, defined by the quantum numbers μ and x are defined as the overlap

$$\psi_{\alpha}(x) = \langle x | \alpha \rangle.$$

Definitions and notations

We introduce the following shorthands for the above two integrals

$$\langle \mu\nu | \hat{v} | \mu\nu \rangle = \int \psi_{\mu}^*(x_i) \psi_{\nu}^*(x_j) V(r_{ij}) \psi_{\mu}(x_i) \psi_{\nu}(x_j) dx_i dx_j,$$

and

$$\langle \mu\nu | \hat{v} | \nu\mu \rangle = \int \psi_{\mu}^*(x_i) \psi_{\nu}^*(x_j) V(r_{ij}) \psi_{\nu}(x_i) \psi_{\mu}(x_j) dx_i dx_j.$$

Definitions and notations

The direct and exchange matrix elements can be brought together if we define the antisymmetrized matrix element

$$\langle \mu\nu | \hat{v} | \mu\nu \rangle_{\text{AS}} = \langle \mu\nu | \hat{v} | \mu\nu \rangle - \langle \mu\nu | \hat{v} | \nu\mu \rangle,$$

or for a general matrix element

$$\langle \mu\nu | \hat{v} | \sigma\tau \rangle_{\text{AS}} = \langle \mu\nu | \hat{v} | \sigma\tau \rangle - \langle \mu\nu | \hat{v} | \tau\sigma \rangle.$$

It has the symmetry property

$$\langle \mu\nu | \hat{v} | \sigma\tau \rangle_{\text{AS}} = -\langle \mu\nu | \hat{v} | \tau\sigma \rangle_{\text{AS}} = -\langle \nu\mu | \hat{v} | \sigma\tau \rangle_{\text{AS}}.$$

Definitions and notations

The antisymmetric matrix element is also hermitian, implying

$$\langle \mu\nu | \hat{v} | \sigma\tau \rangle_{\text{AS}} = \langle \sigma\tau | \hat{v} | \mu\nu \rangle_{\text{AS}}.$$

With these notations we rewrite Eq. (14) as

$$\int \Phi^* \hat{H}_I \Phi d\tau = \frac{1}{2} \sum_{\mu=1}^N \sum_{\nu=1}^N \langle \mu\nu | \hat{v} | \mu\nu \rangle_{\text{AS}}. \quad (15)$$

Definitions and notations

Combining Eqs. (12) and (15) we obtain the energy functional

$$E[\Phi] = \sum_{\mu=1}^N \langle \mu | \hat{h}_0 | \mu \rangle + \frac{1}{2} \sum_{\mu=1}^N \sum_{\nu=1}^N \langle \mu\nu | \hat{v} | \mu\nu \rangle_{\text{AS}}. \quad (16)$$

which we will use as our starting point for the Hartree-Fock calculations.

Our system, quantum dots

We will deal only with systems where all possible single-particle states below a certain level are filled up. Such systems are called closed shell systems, a naming inspired from atomic and nuclear physics. These closed shell systems define what is frequently named **magic numbers**. Quantum dots exhibit also magic numbers, meaning that the addition or removal of one electron requires more energy than systems where the lowest-lying shells are not filled. Using the harmonic oscillator in two dimensions as basis functions (with degenerate single-particle energies) the magic numbers are $N = 2$, $N = 6$, $N = 12$, $N = 20$ etc, where N is the number of electrons. See the table below for more details.

We write our Hamiltonian as a one-body part

$$\hat{H}_0 = \sum_{i=1}^{N_e} \left(-\frac{1}{2} \nabla_i^2 + \frac{\omega^2}{2} r_i^2 \right),$$

and an interacting part

$$\hat{V} = \sum_{i < j}^{N_e} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}.$$

The unperturbed part of the Hamiltonian yields the single-particle energies

$$\epsilon_i = \omega (2n + |m| + 1),$$

where $n = 0, 1, 2, 3, \dots$ and $m = 0, \pm 1, \pm 2, \dots$. The index i runs from $0, 1, 2, \dots$.

First tasks

The first project deals with setting up a Hartree-Fock program that is flexible enough to handle different magic numbers. Hartree-Fock is the simplest possible many-body method. We will use the harmonic oscillator functions as basis functions and via the Hartree-Fock method modify the potential which confines the motion of the electrons by accounting for the Coulomb repulsion between electrons. The Hartree-Fock method leads to a set of coupled single-particle equations. If properly written, the system can easily be extended to three-dimensional electronic systems or other fermionic many-body systems.

First tasks, single-particle basis

If only the harmonic oscillator part of the Hamiltonian, the so-called unperturbed part is given by

$$\hat{H}_0 = \sum_{i=1}^N \left(-\frac{1}{2} \nabla_i^2 + \frac{1}{2} \omega^2 r_i^2 \right).$$

The wave function for one electron in an oscillator potential in two dimensions is

$$\phi_{n_x, n_y}(x, y) = A H_{n_x}(\sqrt{\omega}x) H_{n_y}(\sqrt{\omega}y) \exp(-\omega(x^2 + y^2)/2).$$

The functions $H_{n_x}(\sqrt{\omega}x)$ are so-called Hermite polynomials, discussed in the appendix while A is a normalization constant. For the lowest-lying state we have $n_x = n_y = 0$ and an energy $\epsilon_{n_x, n_y} = \omega(n_x + n_y + 1) = \omega$.

Project 1 a): getting started

Make a program that sets up all the quantum numbers for the single-particle basis using a harmonic oscillator in two dimensions. The list should contain the quantum numbers n_x and n_y , spin and its projection and the single particle energies in atomic units. Define a cutoff in the list according to the harmonic oscillator excitation energy. The table here lists the first four harmonic oscillator shells, with its pertinent degeneracies, and the total number of electrons which can be accomodated up to a given energy. These numbers are the so-called *magic numbers* for a two-dimensional quantum dot. The degeneracies take into account the two spin values an electron can take. Convince yourself about the correctness of this table and use it to check that your code is running correctly.

Shell number	(n_x, n_y)	Energy	Degeneracy	N
1	(0, 0)	$\hbar\omega$	2	2
2	(1, 0), (0, 1)	$2\hbar\omega$	4	6
3	(2, 0), (0, 2), (1, 1)	$3\hbar\omega$	6	12
3	(3, 0), (0, 3), (2, 1), (1, 2)	$4\hbar\omega$	8	20

Project 1 b): Setting up the single-particle basis

With the single-particle basis from the previous step, you should now write a class which encodes the information about the harmonic oscillator single-particle basis. This should include the single-particle energies, spin and its projections as well as the harmonic oscillator functions that depend on n_x and n_y . This class can then be used for both the Monte Carlo and the Hartree-Fock calculations.

Project 1 c): Computing the two-body matrix elements

In the Hartree-Fock project we need to be able to compute the two-electron matrix elements of the two-body Coulomb interaction. In order to set up the two-body matrix elements, we need to define the so-called direct and exchange matrix elements. This leads to what we call an anti-symmetrized matrix element.

We introduce the following shorthands for the integrals

$$\langle pq|\hat{v}|rs\rangle = \int \psi_p^*(x_i)\psi_q^*(x_j)V(r_{ij})\psi_r(x_i)\psi_s(x_j)dx_idx_j,$$

which defines the so-called direct matrix element and

$$\langle pq|\hat{v}|sr\rangle = \int \psi_p^*(x_i)\psi_q^*(x_j)V(r_{ij})\psi_s(x_i)\psi_r(x_j)dx_idx_j,$$

which defines the exchange element. The variables $pqrs$ define all the single-particle quantum numbers. In our case these are n_x , n_y and spin and its projections.

Two-body matrix elements

These matrix elements are defined in terms of two-body quantum numbers. Which quantum numbers are conserved? Which are the possible values the total spin of a two-body state can have?

The direct and exchange matrix elements can be brought together if we define the anti-symmetrized matrix element

$$\langle pq|\hat{v}|rs\rangle_{\text{AS}} = \langle pq|\hat{v}|rs\rangle - \langle pq|\hat{v}|sr\rangle.$$

It has the symmetry property

$$\langle pq|\hat{v}|rs\rangle_{\text{AS}} = -\langle pq|\hat{v}|sr\rangle_{\text{AS}} = -\langle qp|\hat{v}|rs\rangle_{\text{AS}},$$

and

$$\langle pq|\hat{v}|rs\rangle_{\text{AS}} = \langle qp|\hat{v}|sr\rangle_{\text{AS}}.$$

Your task is to write a function which calculates the integral

$$\langle pq|\hat{v}|rs\rangle = \int \psi_p^*(x_i)\psi_q^*(x_j)V(r_{ij})\psi_r(x_i)\psi_s(x_j)dx_idx_j,$$

and then assemble the direct and exchange terms in order to construct the anti-symmetrized matrix elements to be used in the Hartree-Fock calculations. The integral is four-dimensional but can eventually be simplified. Your task is to figure out how. You should also consider whether it is appropriate to store the matrix elements for the Hartree-Fock calculations. [Chapter five](#) of the lecture notes discusses Gaussian quadrature. You may find it convenient to use Hermite polynomials in order to set up the integration points and weights as well as parallelizing the calculations of the integrals.

Program example for numerical integration

The following program shows the integration of a first-order Hermite polynomial for the four-dimensional case. The Hermite polynomials have the following norm

$$\int_{-\infty}^{\infty} e^{-x^2} H_n(x)^2 dx = 2^n n! \sqrt{\pi}.$$

```
// OpenMP code for numerical integration with Gauss-Hermite quadrature
#include <cstdlib>
#include <iostream>
#include <cmath>
#include <iomanip>
#include <omp.h>
using namespace std;

// The integrand in Cartesian coordinates
double Integrand(double, double, double, double);
// The Gauss Hermite integration function
double GaussHermiteIntegration(int);
// Getting the Gaussian quadrature weights and integration points
void GaussHermiteQuadrature(double *, double *, int);
// Start main program
int main (int argc, char* argv[])
{
    // read from terminal the number of integration points
    int n = atoi(argv[1]);
    int thread_num;

    double wtime;
    cout << " Compute 4d-integral using Gaussian quadrature with Hermite quadrature." << endl;
    omp_set_num_threads(4);
    thread_num = omp_get_max_threads ();
    cout << " The number of processors available = " << omp_get_num_procs () << endl ;
    cout << " The number of threads available = " << thread_num << endl;
    cout << " The number of integration points = " << n << endl;
    wtime = omp_get_wtime ( );
    double Integral = GaussHermiteIntegration(n);
    wtime = omp_get_wtime ( ) - wtime;
    cout << setiosflags(ios::showpoint | ios::uppercase);
    cout << setprecision(15) << setw(20) << "Time used integration=" << wtime << endl;
    cout << "Final Integral = " << Integral << endl;
    return 0;
} // end main function

// The integrand, here a first order Hermite polynomial for x1 only
// The other variables are given by zeroth-order Hermite polynomials.
```



```

double Integrand(double x1, double y1, double x2, double y2)
{
    return (2*x1)*(2*x1);
}

//Plain Gauss-Hermite integration with cartesian variables, brute force
double GaussHermiteIntegration(int n)
{
    double *x = new double [n+1];
    double *w = new double [n];
    GaussHermiteQuadrature(x, w, n);
    double Integral = 0.0;
    int i, j, k, l;
    #pragma omp parallel default(shared) private (i, j, k, l) reduction(+:Integral)
    {
        #pragma omp for
        for (i = 0; i < n; i++){
            for (j = 0; j < n; j++){
                for (k = 0; k < n; k++){
                    for (l = 0; l < n; l++){
                        Integral += w[i]*w[j]*w[k]*w[l]*Integrand(x[i],x[j],x[k],x[l]);
                    }
                }
            }
        }
    } // end parallel region
    delete [] x;
    delete [] w;
    return Integral;
}

// Setting Gaussian quadrature weights and integration points
void GaussHermiteQuadrature(double *x, double *w, int n)
{
    int i,its,j,m;
    double p1,p2,p3,pp,z,z1;
    double Epsilon = 3.0e-14, PIM4 = 0.7511255444649425;
    int MaxIterations = 10;
    m=(n+1)/2;
    for (i=1;i<=m;i++) {
        if (i == 1) {
            z=sqrt((double)(2*n+1))-1.85575*pow((double)(2*n+1),-0.16667);
        } else if (i == 2) {
            z -= 1.14*pow((double)n,0.426)/z;
        } else if (i == 3) {
            z=1.86*z-0.86*x[0];
        } else if (i == 4) {
            z=1.91*z-0.91*x[1];
        } else {
            z=2.0*z-x[i-3];
        }
        for (its=1;its<=MaxIterations;its++) {
            p1=PIM4;
            p2=0.0;
            for (j=1;j<=n;j++) {
                p3=p2;
                p2=p1;
                p1=z*sqrt(2.0/j)*p2-sqrt(((double)(j-1))/j)*p3;
            }
            pp=sqrt((double)2*n)*p2;

```

```

        z1=z;
        z=z1-p1/pp;
        if (fabs(z-z1) <= Epsilon) break;
    }
    if (its > MaxIterations) cout << "too many iterations in Hermite quadrature" << endl;
    x[i-1]=z;
    x[n-i] = -z;
    w[i-1]=2.0/(pp*pp);
    w[n-i]=w[i-1];
}
} // Gaussian quadrature weights and integration points

```

Gaussian Quadrature, main idea

In general, an integration formula which is based on a Taylor series using N points, will integrate exactly a polynomial P of degree $N - 1$. That is, the N weights ω_n can be chosen to satisfy N linear equations, see chapter 3 of Ref. [3]. A greater precision for a given amount of numerical work can be achieved if we are willing to give up the requirement of equally spaced integration points. In Gaussian quadrature (hereafter GQ), both the mesh points and the weights are to be determined. The points will not be equally spaced.

The theory behind GQ is to obtain an arbitrary weight ω through the use of so-called orthogonal polynomials. These polynomials are orthogonal in some interval say e.g., $[-1,1]$. Our points x_i are chosen in some optimal sense subject only to the constraint that they should lie in this interval. Together with the weights we have then $2N$ (N the number of points) parameters at our disposal.

Gaussian Quadrature

Even though the integrand is not smooth, we could render it smooth by extracting from it the weight function of an orthogonal polynomial, i.e., we are rewriting

$$I = \int_a^b f(x)dx = \int_a^b W(x)g(x)dx \approx \sum_{i=1}^N \omega_i g(x_i), \quad (17)$$

where g is smooth and W is the weight function, which is to be associated with a given orthogonal polynomial. Note that with a given weight function we end up evaluating the integrand for the function $g(x_i)$.

Gaussian Quadrature, weight function

The weight function W is non-negative in the integration interval $x \in [a, b]$ such that for any $n \geq 0$, the integral $\int_a^b |x|^n W(x)dx$ is integrable. The naming weight function arises from the fact that it may be used to give more emphasis to one part of the interval than another. A quadrature formula

$$\int_a^b W(x)f(x)dx \approx \sum_{i=1}^N \omega_i f(x_i), \quad (18)$$

with N distinct quadrature points (mesh points) is called a Gaussian quadrature formula if it integrates all polynomials $p \in P_{2N-1}$ exactly, that is

$$\int_a^b W(x)p(x)dx = \sum_{i=1}^N \omega_i p(x_i), \quad (19)$$

It is assumed that $W(x)$ is continuous and positive and that the integral

$$\int_a^b W(x)dx$$

exists. Note that the replacement of $f \rightarrow Wg$ is normally a better approximation due to the fact that we may isolate possible singularities of W and its derivatives at the endpoints of the interval.

Gaussian Quadrature weights and integration points

The quadrature weights or just weights (not to be confused with the weight function) are positive and the sequence of Gaussian quadrature formulae is convergent if the sequence Q_N of quadrature formulae

$$Q_N(f) \rightarrow Q(f) = \int_a^b f(x)dx,$$

in the limit $N \rightarrow \infty$.

Gaussian Quadrature

Then we say that the sequence

$$Q_N(f) = \sum_{i=1}^N \omega_i^{(N)} f(x_i^{(N)}),$$

is convergent for all polynomials p , that is

$$Q_N(p) = Q(p)$$

if there exists a constant C such that

$$\sum_{i=1}^N |\omega_i^{(N)}| \leq C,$$

for all N which are natural numbers.

Error in Gaussian Quadrature

The error for the Gaussian quadrature formulae of order N is given by

$$\int_a^b W(x)f(x)dx - \sum_{k=1}^N w_k f(x_k) = \frac{f^{2N}(\xi)}{(2N)!} \int_a^b W(x)[q_N(x)]^2 dx$$

where q_N is the chosen orthogonal polynomial and ξ is a number in the interval $[a, b]$. We have assumed that $f \in C^{2N}[a, b]$, viz. the space of all real or complex $2N$ times continuously differentiable functions.

Important polynomials in Gaussian Quadrature

In science there are several important orthogonal polynomials which arise from the solution of differential equations. Well-known examples are the Legendre, Hermite, Laguerre and Chebyshev polynomials. They have the following weight functions

Weight function	Interval	Polynomial
$W(x) = 1$	$x \in [-1, 1]$	Legendre
$W(x) = e^{-x^2}$	$-\infty \leq x \leq \infty$	Hermite
$W(x) = x^\alpha e^{-x}$	$0 \leq x \leq \infty$	Laguerre
$W(x) = 1/(\sqrt{1-x^2})$	$-1 \leq x \leq 1$	Chebyshev

The importance of the use of orthogonal polynomials in the evaluation of integrals can be summarized as follows.

Gaussian Quadrature, win-win situation

Methods based on Taylor series using N points will integrate exactly a polynomial P of degree $N - 1$. If a function $f(x)$ can be approximated with a polynomial of degree $N - 1$

$$f(x) \approx P_{N-1}(x),$$

with N mesh points we should be able to integrate exactly the polynomial P_{N-1} .

Gaussian quadrature methods promise more than this. We can get a better polynomial approximation with order greater than N to $f(x)$ and still get away with only N mesh points. More precisely, we approximate

$$f(x) \approx P_{2N-1}(x),$$

and with only N mesh points these methods promise that

$$\int f(x)dx \approx \int P_{2N-1}(x)dx = \sum_{i=0}^{N-1} P_{2N-1}(x_i)\omega_i,$$

Gaussian Quadrature, determining mesh points and weights

The reason why we can represent a function $f(x)$ with a polynomial of degree $2N - 1$ is due to the fact that we have $2N$ equations, N for the mesh points and N for the weights.

The mesh points are the zeros of the chosen orthogonal polynomial of order N , and the weights are determined from the inverse of a matrix. An orthogonal polynomial of degree N defined in an interval $[a, b]$ has precisely N distinct zeros on the open interval (a, b) .

Before we detail how to obtain mesh points and weights with orthogonal polynomials, let us revisit some features of orthogonal polynomials by specializing to Legendre polynomials. In the text below, we reserve hereafter the labelling L_N for a Legendre polynomial of order N , while P_N is an arbitrary polynomial of order N . These polynomials form then the basis for the Gauss-Legendre method.

Orthogonal polynomials, Legendre

The Legendre polynomials are the solutions of an important differential equation in Science, namely

$$C(1 - x^2)P - m_l^2 P + (1 - x^2) \frac{d}{dx} \left((1 - x^2) \frac{dP}{dx} \right) = 0.$$

Here C is a constant. For $m_l = 0$ we obtain the Legendre polynomials as solutions, whereas $m_l \neq 0$ yields the so-called associated Legendre polynomials. This differential equation arises in for example the solution of the angular dependence of Schroedinger's equation with spherically symmetric potentials such as the Coulomb potential.

Orthogonal polynomials, Legendre

The corresponding polynomials P are

$$L_k(x) = \frac{1}{2^k k!} \frac{d^k}{dx^k} (x^2 - 1)^k \quad k = 0, 1, 2, \dots,$$

which, up to a factor, are the Legendre polynomials L_k . The latter fulfil the orthogonality relation

$$\int_{-1}^1 L_i(x) L_j(x) dx = \frac{2}{2i + 1} \delta_{ij}, \quad (20)$$

and the recursion relation

$$(j + 1)L_{j+1}(x) + jL_{j-1}(x) - (2j + 1)xL_j(x) = 0. \quad (21)$$

Orthogonal polynomials, Legendre

It is common to choose the normalization condition

$$L_N(1) = 1.$$

With these equations we can determine a Legendre polynomial of arbitrary order with input polynomials of order $N - 1$ and $N - 2$.

As an example, consider the determination of L_0 , L_1 and L_2 . We have that

$$L_0(x) = c,$$

with c a constant. Using the normalization equation $L_0(1) = 1$ we get that

$$L_0(x) = 1.$$

Orthogonal polynomials, Legendre

For $L_1(x)$ we have the general expression

$$L_1(x) = a + bx,$$

and using the orthogonality relation

$$\int_{-1}^1 L_0(x)L_1(x)dx = 0,$$

we obtain $a = 0$ and with the condition $L_1(1) = 1$, we obtain $b = 1$, yielding

$$L_1(x) = x.$$

Orthogonal polynomials, Legendre

We can proceed in a similar fashion in order to determine the coefficients of L_2

$$L_2(x) = a + bx + cx^2,$$

using the orthogonality relations

$$\int_{-1}^1 L_0(x)L_2(x)dx = 0,$$

and

$$\int_{-1}^1 L_1(x)L_2(x)dx = 0,$$

and the condition $L_2(1) = 1$ we would get

$$L_2(x) = \frac{1}{2} (3x^2 - 1). \quad (22)$$

Orthogonal polynomials, Legendre

We note that we have three equations to determine the three coefficients a , b and c .

Alternatively, we could have employed the recursion relation of Eq. (21), resulting in

$$2L_2(x) = 3xL_1(x) - L_0,$$

which leads to Eq. (22).

Orthogonal polynomials, Legendre

The orthogonality relation above is important in our discussion on how to obtain the weights and mesh points. Suppose we have an arbitrary polynomial Q_{N-1} of order $N-1$ and a Legendre polynomial $L_N(x)$ of order N . We could represent Q_{N-1} by the Legendre polynomials through

$$Q_{N-1}(x) = \sum_{k=0}^{N-1} \alpha_k L_k(x), \quad (23)$$

where α_k 's are constants.

Using the orthogonality relation of Eq. (20) we see that

$$\int_{-1}^1 L_N(x) Q_{N-1}(x) dx = \sum_{k=0}^{N-1} \int_{-1}^1 L_N(x) \alpha_k L_k(x) dx = 0. \quad (24)$$

We will use this result in our construction of mesh points and weights in the next subsection.

Orthogonal polynomials, Legendre

In summary, the first few Legendre polynomials are

$$L_0(x) = 1,$$

$$L_1(x) = x,$$

$$L_2(x) = (3x^2 - 1)/2,$$

$$L_3(x) = (5x^3 - 3x)/2,$$

and

$$L_4(x) = (35x^4 - 30x^2 + 3)/8.$$

Orthogonal polynomials, simple code for Legendre polynomials

The following simple function implements the above recursion relation of Eq. (21). for computing Legendre polynomials of order N .

```
// This function computes the Legendre polynomial of degree N
double Legendre( int n, double x)
{
    double r, s, t;
    int m;
    r = 0; s = 1.;
    // Use recursion relation to generate p1 and p2
    for (m=0; m < n; m++)
    {
        t = r; r = s;
        s = (2*m+1)*x*r - m*t;
        s /= (m+1);
    } // end of do loop
    return s;
} // end of function Legendre
```

The variable s represents $L_{j+1}(x)$, while r holds $L_j(x)$ and t the value $L_{j-1}(x)$.

Other orthogonal polynomials, Laguerre polynomials

If we are able to rewrite our integral of Eq. (17) with a weight function $W(x) = x^\alpha e^{-x}$ with integration limits $[0, \infty)$, we could then use the Laguerre polynomials. The polynomials form then the basis for the Gauss-Laguerre method which can be applied to integrals of the form

$$I = \int_0^\infty f(x)dx = \int_0^\infty x^\alpha e^{-x} g(x)dx.$$

Other orthogonal polynomials, Laguerre polynomials

These polynomials arise from the solution of the differential equation

$$\left(\frac{d^2}{dx^2} - \frac{d}{dx} + \frac{\lambda}{x} - \frac{l(l+1)}{x^2} \right) \mathcal{L}(x) = 0,$$

where l is an integer $l \geq 0$ and λ a constant. This equation arises for example from the solution of the radial Schrödinger equation with a centrally symmetric potential such as the Coulomb potential.

Other orthogonal polynomials, Laguerre polynomials

The first few polynomials are

$$\begin{aligned}\mathcal{L}_0(x) &= 1, \\ \mathcal{L}_1(x) &= 1 - x,\end{aligned}$$

$$\begin{aligned}\mathcal{L}_2(x) &= 2 - 4x + x^2, \\ \mathcal{L}_3(x) &= 6 - 18x + 9x^2 - x^3,\end{aligned}$$

and

$$\mathcal{L}_4(x) = x^4 - 16x^3 + 72x^2 - 96x + 24.$$

Other orthogonal polynomials, Laguerre polynomials

They fulfil the orthogonality relation

$$\int_0^\infty e^{-x} \mathcal{L}_n(x)^2 dx = 1,$$

and the recursion relation

$$(n+1)\mathcal{L}_{n+1}(x) = (2n+1-x)\mathcal{L}_n(x) - n\mathcal{L}_{n-1}(x).$$

Other orthogonal polynomials, Hermite polynomials

In a similar way, for an integral which goes like

$$I = \int_{-\infty}^\infty f(x) dx = \int_{-\infty}^\infty e^{-x^2} g(x) dx.$$

we could use the Hermite polynomials in order to extract weights and mesh points. The Hermite polynomials are the solutions of the following differential equation

$$\frac{d^2 H(x)}{dx^2} - 2x \frac{dH(x)}{dx} + (\lambda - 1)H(x) = 0. \quad (25)$$

Other orthogonal polynomials, Hermite polynomials

A typical example is again the solution of Schrodinger's equation, but this time with a harmonic oscillator potential. The first few polynomials are

$$\begin{aligned}H_0(x) &= 1, \\ H_1(x) &= 2x, \\ H_2(x) &= 4x^2 - 2, \\ H_3(x) &= 8x^3 - 12,\end{aligned}$$

and

$$H_4(x) = 16x^4 - 48x^2 + 12.$$

They fulfil the orthogonality relation

$$\int_{-\infty}^\infty e^{-x^2} H_n(x)^2 dx = 2^n n! \sqrt{\pi},$$

and the recursion relation

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x).$$

Integration points and weights with orthogonal polynomials

To understand how the weights and the mesh points are generated, we define first a polynomial of degree $2N - 1$ (since we have $2N$ variables at hand, the mesh points and weights for N points). This polynomial can be represented through polynomial division by

$$P_{2N-1}(x) = L_N(x)P_{N-1}(x) + Q_{N-1}(x),$$

where $P_{N-1}(x)$ and $Q_{N-1}(x)$ are some polynomials of degree $N - 1$ or less. The function $L_N(x)$ is a Legendre polynomial of order N .

Recall that we wanted to approximate an arbitrary function $f(x)$ with a polynomial P_{2N-1} in order to evaluate

$$\int_{-1}^1 f(x)dx \approx \int_{-1}^1 P_{2N-1}(x)dx.$$

Integration points and weights with orthogonal polynomials

We can use Eq. (24) to rewrite the above integral as

$$\int_{-1}^1 P_{2N-1}(x)dx = \int_{-1}^1 (L_N(x)P_{N-1}(x) + Q_{N-1}(x))dx = \int_{-1}^1 Q_{N-1}(x)dx,$$

due to the orthogonality properties of the Legendre polynomials. We see that it suffices to evaluate the integral over $\int_{-1}^1 Q_{N-1}(x)dx$ in order to evaluate $\int_{-1}^1 P_{2N-1}(x)dx$. In addition, at the points x_k where L_N is zero, we have

$$P_{2N-1}(x_k) = Q_{N-1}(x_k) \quad k = 0, 1, \dots, N-1,$$

and we see that through these N points we can fully define $Q_{N-1}(x)$ and thereby the integral. Note that we have chosen to let the numbering of the points run from 0 to $N - 1$. The reason for this choice is that we wish to have the same numbering as the order of a polynomial of degree $N - 1$. This numbering will be useful below when we introduce the matrix elements which define the integration weights w_i .

Integration points and weights with orthogonal polynomials

We develop then $Q_{N-1}(x)$ in terms of Legendre polynomials, as done in Eq. (23),

$$Q_{N-1}(x) = \sum_{i=0}^{N-1} \alpha_i L_i(x). \quad (26)$$

Using the orthogonality property of the Legendre polynomials we have

$$\int_{-1}^1 Q_{N-1}(x)dx = \sum_{i=0}^{N-1} \alpha_i \int_{-1}^1 L_0(x)L_i(x)dx = 2\alpha_0,$$

where we have just inserted $L_0(x) = 1$!

Integration points and weights with orthogonal polynomials

Instead of an integration problem we need now to define the coefficient α_0 . Since we know the values of Q_{N-1} at the zeros of L_N , we may rewrite Eq. (26) as

$$Q_{N-1}(x_k) = \sum_{i=0}^{N-1} \alpha_i L_i(x_k) = \sum_{i=0}^{N-1} \alpha_i L_{ik} \quad k = 0, 1, \dots, N-1. \quad (27)$$

Since the Legendre polynomials are linearly independent of each other, none of the columns in the matrix L_{ik} are linear combinations of the others.

Integration points and weights with orthogonal polynomials

This means that the matrix L_{ik} has an inverse with the properties

$$\hat{L}^{-1} \hat{L} = \hat{I}.$$

Multiplying both sides of Eq. (27) with $\sum_{j=0}^{N-1} L_{ji}^{-1}$ results in

$$\sum_{i=0}^{N-1} (L^{-1})_{ki} Q_{N-1}(x_i) = \alpha_k. \quad (28)$$

Integration points and weights with orthogonal polynomials

We can derive this result in an alternative way by defining the vectors

$$\hat{x}_k = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} \quad \hat{\alpha} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{pmatrix},$$

and the matrix

$$\hat{L} = \begin{pmatrix} L_0(x_0) & L_1(x_0) & \dots & L_{N-1}(x_0) \\ L_0(x_1) & L_1(x_1) & \dots & L_{N-1}(x_1) \\ \dots & \dots & \dots & \dots \\ L_0(x_{N-1}) & L_1(x_{N-1}) & \dots & L_{N-1}(x_{N-1}) \end{pmatrix}.$$

Integration points and weights with orthogonal polynomials

We have then

$$Q_{N-1}(\hat{x}_k) = \hat{L} \hat{\alpha},$$

yielding (if \hat{L} has an inverse)

$$\hat{L}^{-1} Q_{N-1}(\hat{x}_k) = \hat{\alpha},$$

which is Eq. (28).

Integration points and weights with orthogonal polynomials

Using the above results and the fact that

$$\int_{-1}^1 P_{2N-1}(x)dx = \int_{-1}^1 Q_{N-1}(x)dx,$$

we get

$$\int_{-1}^1 P_{2N-1}(x)dx = \int_{-1}^1 Q_{N-1}(x)dx = 2\alpha_0 = 2 \sum_{i=0}^{N-1} (L^{-1})_{0i} P_{2N-1}(x_i).$$

Integration points and weights with orthogonal polynomials

If we identify the weights with $2(L^{-1})_{0i}$, where the points x_i are the zeros of L_N , we have an integration formula of the type

$$\int_{-1}^1 P_{2N-1}(x)dx = \sum_{i=0}^{N-1} \omega_i P_{2N-1}(x_i)$$

and if our function $f(x)$ can be approximated by a polynomial P of degree $2N-1$, we have finally that

$$\int_{-1}^1 f(x)dx \approx \int_{-1}^1 P_{2N-1}(x)dx = \sum_{i=0}^{N-1} \omega_i P_{2N-1}(x_i).$$

In summary, the mesh points x_i are defined by the zeros of an orthogonal polynomial of degree N , that is L_N , while the weights are given by $2(L^{-1})_{0i}$.

Application to the case $N = 2$

Let us apply the above formal results to the case $N = 2$. This means that we can approximate a function $f(x)$ with a polynomial $P_3(x)$ of order $2N-1 = 3$.

The mesh points are the zeros of $L_2(x) = 1/2(3x^2 - 1)$. These points are $x_0 = -1/\sqrt{3}$ and $x_1 = 1/\sqrt{3}$.

Specializing Eq. (27)

$$Q_{N-1}(x_k) = \sum_{i=0}^{N-1} \alpha_i L_i(x_k) \quad k = 0, 1, \dots, N-1.$$

to $N = 2$ yields

$$Q_1(x_0) = \alpha_0 - \alpha_1 \frac{1}{\sqrt{3}},$$

and

$$Q_1(x_1) = \alpha_0 + \alpha_1 \frac{1}{\sqrt{3}},$$

since $L_0(x = \pm 1/\sqrt{3}) = 1$ and $L_1(x = \pm 1/\sqrt{3}) = \pm 1/\sqrt{3}$.

Application to the case $N = 2$

The matrix L_{ik} defined in Eq. (27) is then

$$\hat{L} = \begin{pmatrix} 1 & -\frac{1}{\sqrt{3}} \\ 1 & \frac{1}{\sqrt{3}} \end{pmatrix},$$

with an inverse given by

$$\hat{L}^{-1} = \frac{\sqrt{3}}{2} \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -1 & 1 \end{pmatrix}.$$

The weights are given by the matrix elements $2(L_{0k})^{-1}$. We have thence $\omega_0 = 1$ and $\omega_1 = 1$.

Application to the case $N = 2$

Obviously, there is no problem in changing the numbering of the matrix elements $i, k = 0, 1, 2, \dots, N-1$ to $i, k = 1, 2, \dots, N$. We have chosen to start from zero, since we deal with polynomials of degree $N-1$.

Summarizing, for Legendre polynomials with $N = 2$ we have weights

$$\omega : \{1, 1\},$$

and mesh points

$$x : \left\{ -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right\}.$$

Application to the case $N = 2$

If we wish to integrate

$$\int_{-1}^1 f(x) dx,$$

with $f(x) = x^2$, we approximate

$$I = \int_{-1}^1 x^2 dx \approx \sum_{i=0}^{N-1} \omega_i x_i^2.$$

Application to the case $N = 2$

The exact answer is $2/3$. Using $N = 2$ with the above two weights and mesh points we get

$$I = \int_{-1}^1 x^2 dx = \sum_{i=0}^1 \omega_i x_i^2 = \frac{1}{3} + \frac{1}{3} = \frac{2}{3},$$

the exact answer!

If we were to emply the trapezoidal rule we would get

$$I = \int_{-1}^1 x^2 dx = \frac{b-a}{2} ((a)^2 + (b)^2) / 2 = \frac{1 - (-1)}{2} ((-1)^2 + (1)^2) / 2 = 1!$$

With just two points we can calculate exactly the integral for a second-order polynomial since our methods approximates the exact function with higher order polynomial. How many points do you need with the trapezoidal rule in order to achieve a similar accuracy?

Back to Hermite polynomials and our type of integrals

Our type of integrals will involve expressions on the form (in one dimension as an example)

$$I = \int_{-\infty}^{\infty} \exp(-x^2/2) f(x) dx.$$

When we use Gauss-Hermite quadrature, the exponential function is absorbed into the weight points. We need however to keep in mind that the basic form is given by

$$I = \int_{-\infty}^{\infty} \exp(-x^2) g(x) dx \approx \sum_{i=1}^n w_i g(x_i).$$

This means that we need to make a change of variables $u^2 = x^2/2$ in the first equation so that we end up with

$$I = \int_{-\infty}^{\infty} \exp(-x^2/2) f(x) dx = \sqrt{2} \int_{-\infty}^{\infty} \exp(-u^2) f(\sqrt{2}u) du \approx \sqrt{2} \sum_{i=1}^n w_i f(\sqrt{2}u_i).$$

For Gauss-Hermite quadrature, the weights have a simple analytic expression in terms of Hermite polynomials

$$w_i = \frac{2^{n1} n! \sqrt{\pi}}{n^2 [H_{n-1}(x_i)]^2}.$$

Our integrals

We will need to deal with following type of four dimensional integrals

$$\langle pq|\hat{v}|rs\rangle = A \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp[-(x_1^2 + x_2^2 + y_1^2 + y_2^2)/2] f(x_1, y_1, x_2, y_2) dx_1 dy_1 dx_2 dy_2,$$

where A is a normalization constant and

$$f(x_1, y_1, x_2, y_2) = H_{n_{x_p}}(x_1) H_{n_{y_p}}(y_1) H_{n_{x_q}}(x_2) H_{n_{y_q}}(y_2) H_{n_{x_r}}(x_1) H_{n_{y_r}}(y_1) H_{n_{x_s}}(x_2) H_{n_{y_s}}(y_2) \frac{1}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}$$

where H_n are the various Hermite polynomials. For the case where all $n_{x,y} = 0$ the above integral simplifies to

$$\langle pq|\hat{v}|rs\rangle = A \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\exp [-(x_1^2 + x_2^2 + y_1^2 + y_2^2)/2]}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} dx_1 dy_1 dx_2 dy_2.$$

Changing from the laboratory frame to the relative and center of mass frame, with

$$\mathbf{R} = \frac{\mathbf{r}_1 + \mathbf{r}_2}{2} \quad \mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$$

and introducing polar coordinates the above integral can be written as

$$I = 4\pi^2 A \int_0^{\infty} \exp(-R^2) R dR \int_0^{\infty} \exp(-r^2) r dr \frac{1}{r} = 2A\pi^{5/2}.$$

This provides a useful check of our program.

Our integrals

The integral

$$\langle pq|\hat{v}|rs\rangle = A \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp [-(x_1^2 + x_2^2 + y_1^2 + y_2^2)/2] f(x_1, y_1, x_2, y_2) dx_1 dy_1 dx_2 dy_2,$$

can be calculate in analytical form if we switch to polar coordinates.

Single-particle functions in polar coordinates

Instead of Hermite polynomials which we will need for the Variational Monte Carlo, it is more convenient for the Hartree-Fock calculations to use polar coordinates.

Using polar coordinates

$$x = r \cos \theta \tag{29}$$

$$y = r \sin \theta \tag{30}$$

$$r = \sqrt{x^2 + y^2}, \tag{31}$$

the normalized solution for the angular part in two dimensions is

$$Y(\theta) = \frac{1}{\sqrt{2\pi}} e^{im\theta}$$

The total wavefunction must satisfy the physical condition that $\psi(r, \theta) = \psi(r, \theta + 2\pi)$ This makes a restriction on the quantum number m which can take integral values

$$m = 0, \pm 1, \pm 2, \dots$$

Eigenfunctions in two dimensions

The time-independent part of the wave function is separated in an angular and a radial part

$$\psi(r, \theta) = R(r) \frac{1}{\sqrt{2\pi}} e^{im\theta}, \quad m = 0, \pm 1, \pm 2, \dots$$

The solution of the radial equation is

$$R_{nm}(r) = \sqrt{\frac{2n!}{(n+|m|)!}} \beta^{\frac{1}{2}(|m|+1)} r^{|m|} e^{-\frac{1}{2}\beta r^2} L_n^{|m|}(\beta r^2)$$

Here the subscript n denote the principal quantum number, and m is the angular momentum number

$$n = 0, 1, 2, 3, \dots \quad (32)$$

$$m = 0, \pm 1, \pm 2, \pm 3, \dots \quad (33)$$

$L_n^{|m|}$ is the associated Laguerre polynomials discussed above, and β is defined as

$$\beta = \frac{m\omega}{\hbar}$$

Final expression for the eigenfunction

The final eigenfunction for an electron moving in a two-dimensional harmonic oscillator is then

$$\psi(r, \theta) = \sqrt{\frac{n!}{\pi(n+|m|)!}} \beta^{\frac{1}{2}(|m|+1)} r^{|m|} e^{-\frac{1}{2}\beta r^2} L_n^{|m|}(\beta r^2) e^{im\theta},$$

with the eigenvalue

$$E = \hbar\omega(2n + |m| + 1)$$

which is the same as the energy with cartesian coordinates but now in terms of n_x and n_y , that is

$$E = \hbar\omega(n_x + n_y + 1).$$

Program for computing the Coulomb interaction in polar coordinates

The program is based on the analytical expression given by [Anisimova and Matulis](#). It returns the value of the integral (without checking for spin values, you need to add this test) using as input the quantum numbers n_i and m_i , that is

$$\langle pq|\hat{v}|rs\rangle = \langle (n_p m_p)(n_q m_q)|\hat{v}|(n_r m_r)(n_s m_s)\rangle,$$

and the main code is (click on the above link for the full code)


```

#include "Coulomb_Functions.hpp"

int main(int argc, char * argv[])
{
    if(argc != 10){ std::cerr << "Wrong Input: should be ./QD_Coulomb hw n1 m1 n2 m2 n3 m3 n4 m4\n";
    double hw = std::atof(argv[1]);
    int n1 = std::atoi(argv[2]);
    int m1 = std::atoi(argv[3]);
    int n2 = std::atoi(argv[4]);
    int m2 = std::atoi(argv[5]);
    int n3 = std::atoi(argv[6]);
    int m3 = std::atoi(argv[7]);
    int n4 = std::atoi(argv[8]);
    int m4 = std::atoi(argv[9]);

    double TBME = Coulomb_H0(hw, n1, m1, n2, m2, n3, m3, n4, m4);
    std::cout << std::setprecision(12);
    std::cout << "< " << n1 << ", " << m1 << " ; " << n2 << ", " << m2 << " || V || " << n3 << ", " << m3 << " ; " << n4 << ", " << m4 << " >> \n";

    return 0;
}

```

Conserved quantum numbers

When setting up the matrix elements

$$\langle pq|\hat{v}|rs\rangle = \langle (n_p m_p)(n_q m_q)|\hat{v}|(n_r m_r)(n_s m_s)\rangle,$$

you need to take into account that there are conserved two-electron quantum numbers since the Hamiltonian is invariant under rotations, namely

$$m_p + m_q = M = m_r + m_s,$$

and the total spin projection which is not included in the above code. You need to add this as a test, that is check that

$$\sigma_p + \sigma_q = S_z = \sigma_r + \sigma_s.$$

Finally, you need to ensure that the single-particle spinors satisfy $\sigma_p = \sigma_r$ and $\sigma_q = \sigma_s$. Pay in particular attention to this when you compute the exchange matrix elements.

Reminder on Variational Calculus and Lagrangian Multipliers

The calculus of variations involves problems where the quantity to be minimized or maximized is an integral.

In the general case we have an integral of the type

$$E[\Phi] = \int_a^b f(\Phi(x), \frac{\partial \Phi}{\partial x}, x) dx,$$

where E is the quantity which is sought minimized or maximized. The problem is that although f is a function of the variables Φ , $\partial \Phi / \partial x$ and x , the exact

dependence of Φ on x is not known. This means again that even though the integral has fixed limits a and b , the path of integration is not known. In our case the unknown quantities are the single-particle wave functions and we wish to choose an integration path which makes the functional $E[\Phi]$ stationary. This means that we want to find minima, or maxima or saddle points. In physics we search normally for minima. Our task is therefore to find the minimum of $E[\Phi]$ so that its variation δE is zero subject to specific constraints. In our case the constraints appear as the integral which expresses the orthogonality of the single-particle wave functions. The constraints can be treated via the technique of Lagrangian multipliers

Variational Calculus and Lagrangian Multipliers, simple example

Let us specialize to the expectation value of the energy for one particle in three-dimensions. This expectation value reads

$$E = \int dx dy dz \psi^*(x, y, z) \hat{H} \psi(x, y, z),$$

with the constraint

$$\int dx dy dz \psi^*(x, y, z) \psi(x, y, z) = 1,$$

and a Hamiltonian

$$\hat{H} = -\frac{1}{2}\nabla^2 + V(x, y, z).$$

We will, for the sake of notational convenience, skip the variables x, y, z below, and write for example $V(x, y, z) = V$.

Manipulating terms

The integral involving the kinetic energy can be written as, with the function ψ vanishing strongly for large values of x, y, z (given here by the limits a and b),

$$\int_a^b dx dy dz \psi^* \left(-\frac{1}{2}\nabla^2 \right) \psi dx dy dz = \psi^* \nabla \psi \Big|_a^b + \int_a^b dx dy dz \frac{1}{2} \nabla \psi^* \nabla \psi.$$

We will drop the limits a and b in the remaining discussion. Inserting this expression into the expectation value for the energy and taking the variational minimum we obtain

$$\delta E = \delta \left\{ \int dx dy dz \left(\frac{1}{2} \nabla \psi^* \nabla \psi + V \psi^* \psi \right) \right\} = 0.$$

Adding the Lagrangian multiplier

The constraint appears in integral form as

$$\int dx dy dz \psi^* \psi = \text{constant},$$

and multiplying with a Lagrangian multiplier λ and taking the variational minimum we obtain the final variational equation

$$\delta \left\{ \int dx dy dz \left(\frac{1}{2} \nabla \psi^* \nabla \psi + V \psi^* \psi - \lambda \psi^* \psi \right) \right\} = 0.$$

We introduce the function f

$$f = \frac{1}{2} \nabla \psi^* \nabla \psi + V \psi^* \psi - \lambda \psi^* \psi = \frac{1}{2} (\psi_x^* \psi_x + \psi_y^* \psi_y + \psi_z^* \psi_z) + V \psi^* \psi - \lambda \psi^* \psi,$$

where we have skipped the dependence on x, y, z and introduced the shorthand ψ_x , ψ_y and ψ_z for the various derivatives.

And with the Euler-Lagrange equations we get

For ψ^* the Euler-Lagrange equations yield

$$\frac{\partial f}{\partial \psi^*} - \frac{\partial}{\partial x} \frac{\partial f}{\partial \psi_x^*} - \frac{\partial}{\partial y} \frac{\partial f}{\partial \psi_y^*} - \frac{\partial}{\partial z} \frac{\partial f}{\partial \psi_z^*} = 0,$$

which results in

$$-\frac{1}{2} (\psi_{xx} + \psi_{yy} + \psi_{zz}) + V \psi = \lambda \psi.$$

We can then identify the Lagrangian multiplier as the energy of the system. The last equation is nothing but the standard Schroedinger equation and the variational approach discussed here provides a powerful method for obtaining approximate solutions of the wave function.

Hartree-Fock by varying the coefficients of a wave function expansion

In deriving the Hartree-Fock equations, we will expand the single-particle functions in a known basis and vary the coefficients, that is, the new single-particle wave function is written as a linear expansion in terms of a fixed chosen orthogonal basis (for example the well-known harmonic oscillator functions or the hydrogen-like functions etc). We define our new Hartree-Fock single-particle basis by performing a unitary transformation on our previous basis (labelled with greek indices) as

$$\psi_p^{HF} = \sum_{\lambda} C_{p\lambda} \phi_{\lambda}. \quad (34)$$

In this case we vary the coefficients $C_{p\lambda}$. If the basis has infinitely many solutions, we need to truncate the above sum. We assume that the basis ϕ_{λ} is orthogonal. A unitary transformation keeps the orthogonality, as discussed in exercise 1 below.

More on linear algebra

In the previous slide we stated that a unitary transformation keeps the orthogonality, as discussed in exercise 1 below. To see this consider first a basis of vectors \mathbf{v}_i ,

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ \vdots \\ v_{in} \end{bmatrix}$$

We assume that the basis is orthogonal, that is

$$\mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}.$$

An orthogonal or unitary transformation

$$\mathbf{w}_i = \mathbf{U} \mathbf{v}_i,$$

preserves the dot product and orthogonality since

$$\mathbf{w}_j^T \mathbf{w}_i = (\mathbf{U} \mathbf{v}_j)^T \mathbf{U} \mathbf{v}_i = \mathbf{v}_j^T \mathbf{U}^T \mathbf{U} \mathbf{v}_i = \mathbf{v}_j^T \mathbf{v}_i = \delta_{ij}.$$

Coefficients of a wave function expansion

This means that if the coefficients $C_{p\lambda}$ belong to a unitary or orthogonal transformation (using the Dirac bra-ket notation)

$$|p\rangle = \sum_{\lambda} C_{p\lambda} |\lambda\rangle,$$

orthogonality is preserved, that is $\langle \alpha | \beta \rangle = \delta_{\alpha\beta}$ and $\langle p | q \rangle = \delta_{pq}$.

This property is extremely useful when we build up a basis of many-body Slater determinant based states.

Note also that although a basis $|\alpha\rangle$ contains an infinity of states, for practical calculations we have always to make some truncations.

More Basic Matrix Features, simple 2×2 determinant, useful property of determinants

Before we develop the Hartree-Fock equations, there is another very useful property of determinants that we will use both in connection with Hartree-Fock calculations and later shell-model calculations.

Consider the following determinant

$$\begin{vmatrix} \alpha_1 b_{11} + \alpha_2 b_{12} & a_{12} \\ \alpha_1 b_{21} + \alpha_2 b_{22} & a_{22} \end{vmatrix} = \alpha_1 \begin{vmatrix} b_{11} & a_{12} \\ b_{21} & a_{22} \end{vmatrix} + \alpha_2 \begin{vmatrix} b_{12} & a_{12} \\ b_{22} & a_{22} \end{vmatrix}$$

More Basic Matrix Features, $n \times n$ determinant

We can generalize this to an $n \times n$ matrix and have

$$\begin{vmatrix} a_{11} & a_{12} & \dots & \sum_{k=1}^n c_k b_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \sum_{k=1}^n c_k b_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \sum_{k=1}^n c_k b_{nk} & \dots & a_{nn} \end{vmatrix} = \sum_{k=1}^n c_k \begin{vmatrix} a_{11} & a_{12} & \dots & b_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & b_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_{nk} & \dots & a_{nn} \end{vmatrix}.$$

This is a property we will use in our Hartree-Fock discussions.

More Basic Matrix Features, a general $n \times n$ determinant

We can generalize the previous results, now with all elements a_{ij} being given as functions of linear combinations of various coefficients c and elements b_{ij} ,

$$\begin{vmatrix} \sum_{k=1}^n b_{1k} c_{k1} & \sum_{k=1}^n b_{1k} c_{k2} & \dots & \sum_{k=1}^n b_{1k} c_{kj} & \dots & \sum_{k=1}^n b_{1k} c_{kn} \\ \sum_{k=1}^n b_{2k} c_{k1} & \sum_{k=1}^n b_{2k} c_{k2} & \dots & \sum_{k=1}^n b_{2k} c_{kj} & \dots & \sum_{k=1}^n b_{2k} c_{kn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sum_{k=1}^n b_{nk} c_{k1} & \sum_{k=1}^n b_{nk} c_{k2} & \dots & \sum_{k=1}^n b_{nk} c_{kj} & \dots & \sum_{k=1}^n b_{nk} c_{kn} \end{vmatrix} = \det(\mathbf{C})\det(\mathbf{B}),$$

where $\det(\mathbf{C})$ and $\det(\mathbf{B})$ are the determinants of $n \times n$ matrices with elements c_{ij} and b_{ij} respectively. This is a property we will use in our Hartree-Fock discussions. Convince yourself about the correctness of the above expression by setting $n = 2$.

A general Slater determinant

With our definition of the new basis in terms of an orthogonal basis we have

$$\psi_p(x) = \sum_{\lambda} C_{p\lambda} \phi_{\lambda}(x).$$

If the coefficients $C_{p\lambda}$ belong to an orthogonal or unitary matrix, the new basis is also orthogonal. Our Slater determinant in the new basis $\psi_p(x)$ is written as

$$\frac{1}{\sqrt{A!}} \begin{vmatrix} \psi_p(x_1) & \psi_p(x_2) & \dots & \dots & \psi_p(x_A) \\ \psi_q(x_1) & \psi_q(x_2) & \dots & \dots & \psi_q(x_A) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \psi_t(x_1) & \psi_t(x_2) & \dots & \dots & \psi_t(x_A) \end{vmatrix} = \frac{1}{\sqrt{A!}} \begin{vmatrix} \sum_{\lambda} C_{p\lambda} \phi_{\lambda}(x_1) & \sum_{\lambda} C_{p\lambda} \phi_{\lambda}(x_2) & \dots & \dots & \sum_{\lambda} C_{p\lambda} \phi_{\lambda}(x_A) \\ \sum_{\lambda} C_{q\lambda} \phi_{\lambda}(x_1) & \sum_{\lambda} C_{q\lambda} \phi_{\lambda}(x_2) & \dots & \dots & \sum_{\lambda} C_{q\lambda} \phi_{\lambda}(x_A) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{\lambda} C_{t\lambda} \phi_{\lambda}(x_1) & \sum_{\lambda} C_{t\lambda} \phi_{\lambda}(x_2) & \dots & \dots & \sum_{\lambda} C_{t\lambda} \phi_{\lambda}(x_A) \end{vmatrix}$$

which is nothing but $\det(\mathbf{C})\det(\Phi)$, with $\det(\Phi)$ being the determinant given by the basis functions $\phi_{\lambda}(x)$.

Hartree-Fock by varying the coefficients of a wave function expansion

It is normal to choose a single-particle basis defined as the eigenfunctions of parts of the full Hamiltonian. The typical situation consists of the solutions of the one-body part of the Hamiltonian, that is we have

$$\hat{h}_0\phi_\lambda = \epsilon_\lambda\phi_\lambda.$$

The single-particle wave functions $\phi_\lambda(\mathbf{r})$, defined by the quantum numbers λ and \mathbf{r} are defined as the overlap

$$\phi_\lambda(\mathbf{r}) = \langle \mathbf{r} | \lambda \rangle.$$

Hartree-Fock by varying the coefficients of a wave function expansion

In our discussions hereafter we will use our definitions of single-particle states above and below the Fermi (F) level given by the labels $ijkl \dots \leq F$ for so-called single-hole states and $abcd \dots > F$ for so-called particle states. For general single-particle states we employ the labels $pqrs \dots$.

Hartree-Fock by varying the coefficients of a wave function expansion

In Eq. (16), restated here

$$E[\Phi] = \sum_{\mu=1}^A \langle \mu | h | \mu \rangle + \frac{1}{2} \sum_{\mu=1}^A \sum_{\nu=1}^A \langle \mu\nu | \hat{v} | \mu\nu \rangle_{AS},$$

we found the expression for the energy functional in terms of the basis function $\phi_\lambda(\mathbf{r})$. We then varied the above energy functional with respect to the basis functions $|\mu\rangle$. Now we are interested in defining a new basis defined in terms of a chosen basis as defined in Eq. (34). We can then rewrite the energy functional as

$$E[\Phi^{HF}] = \sum_{i=1}^A \langle i | h | i \rangle + \frac{1}{2} \sum_{ij=1}^A \langle ij | \hat{v} | ij \rangle_{AS}, \quad (35)$$

where Φ^{HF} is the new Slater determinant defined by the new basis of Eq. (34).

Hartree-Fock by varying the coefficients of a wave function expansion

Using Eq. (34) we can rewrite Eq. (35) as

$$E[\Psi] = \sum_{i=1}^A \sum_{\alpha\beta} C_{i\alpha}^* C_{i\beta} \langle \alpha | h | \beta \rangle + \frac{1}{2} \sum_{ij=1}^A \sum_{\alpha\beta\gamma\delta} C_{i\alpha}^* C_{j\beta}^* C_{i\gamma} C_{j\delta} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle_{AS}. \quad (36)$$

Hartree-Fock by varying the coefficients of a wave function expansion

We wish now to minimize the above functional. We introduce again a set of Lagrange multipliers, noting that since $\langle i|j\rangle = \delta_{i,j}$ and $\langle \alpha|\beta\rangle = \delta_{\alpha,\beta}$, the coefficients $C_{i\gamma}$ obey the relation

$$\langle i|j\rangle = \delta_{i,j} = \sum_{\alpha\beta} C_{i\alpha}^* C_{i\beta} \langle \alpha|\beta\rangle = \sum_{\alpha} C_{i\alpha}^* C_{i\alpha},$$

which allows us to define a functional to be minimized that reads

$$F[\Phi^{HF}] = E[\Phi^{HF}] - \sum_{i=1}^A \epsilon_i \sum_{\alpha} C_{i\alpha}^* C_{i\alpha}. \quad (37)$$

Hartree-Fock by varying the coefficients of a wave function expansion

Minimizing with respect to $C_{i\alpha}^*$, remembering that the equations for $C_{i\alpha}^*$ and $C_{i\alpha}$ can be written as two independent equations, we obtain

$$\frac{d}{dC_{i\alpha}^*} \left[E[\Phi^{HF}] - \sum_j \epsilon_j \sum_{\alpha} C_{j\alpha}^* C_{j\alpha} \right] = 0,$$

which yields for every single-particle state i and index α (recalling that the coefficients $C_{i\alpha}$ are matrix elements of a unitary (or orthogonal for a real symmetric matrix) matrix) the following Hartree-Fock equations

$$\sum_{\beta} C_{i\beta} \langle \alpha|h|\beta\rangle + \sum_{j=1}^A \sum_{\beta\gamma\delta} C_{j\beta}^* C_{j\delta} C_{i\gamma} \langle \alpha\beta|\hat{v}|\gamma\delta\rangle_{AS} = \epsilon_i^{HF} C_{i\alpha}.$$

Hartree-Fock by varying the coefficients of a wave function expansion

We can rewrite this equation as (changing dummy variables)

$$\sum_{\beta} \left\{ \langle \alpha|h|\beta\rangle + \sum_j \sum_{\gamma\delta} C_{j\gamma}^* C_{j\delta} \langle \alpha\gamma|\hat{v}|\beta\delta\rangle_{AS} \right\} C_{i\beta} = \epsilon_i^{HF} C_{i\alpha}.$$

Note that the sums over greek indices run over the number of basis set functions (in principle an infinite number).

Hartree-Fock by varying the coefficients of a wave function expansion

Defining

$$h_{\alpha\beta}^{HF} = \langle \alpha | h | \beta \rangle + \sum_{j=1}^A \sum_{\gamma\delta} C_{j\gamma}^* C_{j\delta} \langle \alpha\gamma | \hat{v} | \beta\delta \rangle_{AS},$$

we can rewrite the new equations as

$$\sum_{\gamma} h_{\alpha\beta}^{HF} C_{i\beta} = \epsilon_i^{HF} C_{i\alpha}. \quad (38)$$

The latter is nothing but a standard eigenvalue problem.

It suffices to tabulate the matrix elements $\langle \alpha | h | \beta \rangle$ and $\langle \alpha\gamma | \hat{v} | \beta\delta \rangle_{AS}$ once and for all. Successive iterations require thus only a look-up in tables over one-body and two-body matrix elements. These details will be discussed below when we solve the Hartree-Fock equations numerically.

Hartree-Fock algorithm

Our Hartree-Fock matrix is thus

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

The Hartree-Fock equations are solved in an iterative waym starting with a guess for the coefficients $C_{j\gamma} = \delta_{j,\gamma}$ and solving the equations by diagonalization till the new single-particle energies ϵ_i^{HF} do not change anymore by a prefixed quantity.

Hartree-Fock algorithm

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

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

The Hartree-Fock eigenvalue problem

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

can be written out in a more compact form as

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

Hartree-Fock algorithm

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 subscript n stands for iteration n . To set up the algorithm we can proceed as follows:

- 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.
- The Hartree-Fock matrix simplifies then to (assuming that the coefficients $C_{i\alpha}$ are real)

$$\hat{h}_{\alpha\beta}^{HF} = \epsilon_{\alpha}\delta_{\alpha,\beta} + \sum_{j=1}^A \sum_{\gamma\delta} C_{j\gamma}^{(0)} C_{j\delta}^{(0)} \langle \alpha\gamma | \hat{v} | \beta\delta \rangle_{AS}.$$

Hartree-Fock algorithm

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

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

$$\sum_{j=1}^A \sum_{\gamma\delta} C_{j\gamma}^{(1)} C_{j\delta}^{(1)} \langle \alpha\gamma | \hat{v} | \beta\delta \rangle_{AS}.$$

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

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

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.

Analysis of Hartree-Fock equations and Koopman's theorem

We can rewrite the ground state energy by adding and subtracting $\hat{u}^{HF}(x_i)$

$$E_0^{HF} = \langle \Phi_0 | \hat{H} | \Phi_0 \rangle = \sum_{i \leq F}^A \langle i | \hat{h}_0 + \hat{u}^{HF} | j \rangle + \frac{1}{2} \sum_{i \leq F}^A \sum_{j \leq F}^A [\langle ij | \hat{v} | ij \rangle - \langle ij | \hat{v} | ji \rangle] - \sum_{i \leq F}^A \langle i | \hat{u}^{HF} | i \rangle,$$

which results in

$$E_0^{HF} = \sum_{i \leq F}^A \epsilon_i^{HF} + \frac{1}{2} \sum_{i \leq F}^A \sum_{j \leq F}^A [\langle ij | \hat{v} | ij \rangle - \langle ij | \hat{v} | ji \rangle] - \sum_{i \leq F}^A \langle i | \hat{u}^{HF} | i \rangle.$$

Our single-particle states $ijk \dots$ are now single-particle states obtained from the solution of the Hartree-Fock equations.

Analysis of Hartree-Fock equations and Koopman's theorem

Using our definition of the Hartree-Fock single-particle energies we obtain then the following expression for the total ground-state energy

$$E_0^{HF} = \sum_{i \leq F}^A \varepsilon_i - \frac{1}{2} \sum_{i \leq F}^A \sum_{j \leq F}^A [\langle ij | \hat{v} | ij \rangle - \langle ij | \hat{v} | ji \rangle].$$

This form will be used in our discussion of Koopman's theorem.

Analysis of Hartree-Fock equations and Koopman's theorem

Atomic physics case We have

$$E[\Phi^{HF}(N)] = \sum_{i=1}^H \langle i | \hat{h}_0 | i \rangle + \frac{1}{2} \sum_{ij=1}^N \langle ij | \hat{v} | ij \rangle_{AS},$$

where $\Phi^{HF}(N)$ is the new Slater determinant defined by the new basis of Eq. (34) for N electrons (same Z). If we assume that the single-particle wave functions in the new basis do not change when we remove one electron or add one electron, we can then define the corresponding energy for the $N - 1$ systems as

$$E[\Phi^{HF}(N-1)] = \sum_{i=1; i \neq k}^N \langle i | \hat{h}_0 | i \rangle + \frac{1}{2} \sum_{ij=1; i, j \neq k}^N \langle ij | \hat{v} | ij \rangle_{AS},$$

where we have removed a single-particle state $k \leq F$, that is a state below the Fermi level.

Analysis of Hartree-Fock equations and Koopman's theorem

Calculating the difference

$$E[\Phi^{HF}(N)] - E[\Phi^{HF}(N-1)] = \langle k | \hat{h}_0 | k \rangle + \frac{1}{2} \sum_{i=1; i \neq k}^N \langle ik | \hat{v} | ik \rangle_{AS} - \frac{1}{2} \sum_{j=1; j \neq k}^N \langle kj | \hat{v} | kj \rangle_{AS},$$

we obtain

$$E[\Phi^{HF}(N)] - E[\Phi^{HF}(N-1)] = \langle k | \hat{h}_0 | k \rangle + \frac{1}{2} \sum_{j=1}^N \langle kj | \hat{v} | kj \rangle_{AS}$$

which is just our definition of the Hartree-Fock single-particle energy

$$E[\Phi^{HF}(N)] - E[\Phi^{HF}(N-1)] = \epsilon_k^{HF}$$

Analysis of Hartree-Fock equations and Koopman's theorem

Similarly, we can now compute the difference (we label the single-particle states above the Fermi level as $abcd > F$)

$$E[\Phi^{\text{HF}}(N+1)] - E[\Phi^{\text{HF}}(N)] = \epsilon_a^{\text{HF}}.$$

These two equations can thus be used to the electron affinity or ionization energies, respectively. Koopman's theorem states that for example the ionization energy of a closed-shell system is given by the energy of the highest occupied single-particle state. If we assume that changing the number of electrons from N to $N+1$ does not change the Hartree-Fock single-particle energies and eigenfunctions, then Koopman's theorem simply states that the ionization energy of an atom is given by the single-particle energy of the last bound state. In a similar way, we can also define the electron affinities.

Analysis of Hartree-Fock equations and Koopman's theorem

As an example, consider a simple model for atomic sodium, Na. Neutral sodium has eleven electrons, with the weakest bound one being confined the $3s$ single-particle quantum numbers. The energy needed to remove an electron from neutral sodium is rather small, 5.1391 eV, a feature which pertains to all alkali metals. Having performed a Hartree-Fock calculation for neutral sodium would then allows us to compute the ionization energy by using the single-particle energy for the $3s$ states, namely $\epsilon_{3s}^{\text{HF}}$.

From these considerations, we see that Hartree-Fock theory allows us to make a connection between experimental observables (here ionization and affinity energies) and the underlying interactions between particles. In this sense, we are now linking the dynamics and structure of a many-body system with the laws of motion which govern the system. Our approach is a reductionistic one, meaning that we want to understand the laws of motion in terms of the particles or degrees of freedom which we believe are the fundamental ones. Our Slater determinant, being constructed as the product of various single-particle functions, follows this philosophy.

*

Exercise 1: Hartree-Fock Slater determinant

Consider a Slater determinant built up of orthogonal single-particle orbitals ψ_λ , with $\lambda = 1, 2, \dots, A$.

The unitary transformation

$$\psi_a = \sum_{\lambda} C_{a\lambda} \phi_\lambda,$$

brings us into the new basis. The new basis has quantum numbers $a = 1, 2, \dots, A$.

aragraph!paragraph>paragraph>-0.5em

a) Show that the new basis is orthogonal.
aragraph!paragraph>paragraph>-0.5em

b) Show that the new Slater determinant constructed from the new single-particle wave functions can be written as the determinant based on the previous basis and the determinant of the matrix C .

aragraph!paragraph>paragraph>-0.5em

c) Show that the old and the new Slater determinants are equal up to a complex constant with absolute value unity.

Hint. Hint: C is a unitary matrix.

*

Exercise 2: Matrix elements for the Hartree-Fock method and the nuclear shell model

We will assume that we can build various Slater determinants using an orthogonal single-particle basis ψ_λ , with $\lambda = 1, 2, \dots, A$.

The aim of this exercise is to set up specific matrix elements that will turn useful when we start our discussions of the nuclear shell model. In particular you will notice, depending on the character of the operator, that many matrix elements will actually be zero.

Consider three A -particle Slater determinants $|\Phi_0\rangle$, $|\Phi_i^a\rangle$ and $|\Phi_{ij}^{ab}\rangle$, where the notation means that Slater determinant $|\Phi_i^a\rangle$ differs from $|\Phi_0\rangle$ by one single-particle state, that is a single-particle state ψ_i is replaced by a single-particle state ψ_a . It will later be interpreted as a so-called one-particle-one-hole excitation. Similarly, the Slater determinant $|\Phi_{ij}^{ab}\rangle$ differs by two single-particle states from $|\Phi_0\rangle$ and is normally thought of as a two-particle-two-hole excitation.

Define a general onebody operator $\hat{F} = \sum_i^A \hat{f}(x_i)$ and a general twobody operator $\hat{G} = \sum_{i>j}^A \hat{g}(x_i, x_j)$ with g being invariant under the interchange of the coordinates of particles i and j . You can use here the results from the second exercise set, exercise 3.

aragraph!paragraph>paragraph>-0.5em

a) Calculate

$$\langle \Phi_0 | \hat{F} | \Phi_0 \rangle,$$

and

$$\langle \Phi_0 | \hat{G} | \Phi_0 \rangle.$$

aragraph!paragraph>paragraph>-0.5em

b) Find thereafter

$$\langle \Phi_0 | \hat{F} | \Phi_i^a \rangle,$$

and

$$\langle \Phi_0 | \hat{G} | \Phi_i^a \rangle,$$

aragraph!paragraph>paragraph>-0.5em

c) Finally, find

$$\langle \Phi_0 | \hat{F} | \Phi_{ij}^{ab} \rangle,$$

and

$$\langle \Phi_0 | \hat{G} | \Phi_{ij}^{ab} \rangle.$$

What happens with the two-body operator if we have a transition probability of the type

$$\langle \Phi_0 | \hat{G} | \Phi_{ijk}^{abc} \rangle,$$

where the Slater determinant to the right of the operator differs by more than two single-particle states?

aragraph!paragraph>paragraph>-0.5em

d) With an orthogonal basis of Slater determinants Φ_λ , we can now construct an exact many-body state as a linear expansion of Slater determinants, that is, a given exact state

$$\Psi_i = \sum_{\lambda=0}^{\infty} C_{i\lambda} \Phi_\lambda.$$

In all practical calculations the infinity is replaced by a given truncation in the sum.

If you are to compute the expectation value of (at most) a two-body Hamiltonian for the above exact state

$$\langle \Psi_i | \hat{H} | \Psi_i \rangle,$$

based on the calculations above, which are the only elements which will contribute? (there is no need to perform any calculation here, use your results from exercises a), b), and c)).

These results simplify to a large extent shell-model calculations.

Developing a Hartree-Fock program

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

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

with \hat{V}_{ij} is the two-body potential whose matrix elements are calculated on fly in your program. See expression below.

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

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

can be written out in a more compact form as

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

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

$$\rho_{\gamma\delta} = \sum_{i=1}^N \langle \gamma | i \rangle \langle i | \delta \rangle = \sum_{i=1}^N C_{i\gamma} C_{i\delta}^*. \quad (39)$$

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 .

Program for computing the Coulomb interaction in polar coordinates

Here we show a simple code in python for a Hartree-Fock calculation using precalculated matrix elements.

```
import numpy as np
from decimal import Decimal
# expectation value for the one body part, Harmonic oscillator in three dimensions
def onebody(i, n, l):
    homega = 10.0
    return homega*(2*n[i] + l[i] + 1.5)

if __name__ == '__main__':
    Nparticles = 16
    """ Read quantum numbers from file """
    index = []
    n = []
    l = []
    j = []
    mj = []
    tz = []
```

```

spOrbitals = 0
with open("nucleispnumbers.dat", "r") as qnumfile:
    for line in qnumfile:
        nums = line.split()
        if len(nums) != 0:
            index.append(int(nums[0]))
            n.append(int(nums[1]))
            l.append(int(nums[2]))
            j.append(int(nums[3]))
            mj.append(int(nums[4]))
            tz.append(int(nums[5]))
            spOrbitals += 1

""" Read two-nucleon interaction elements (integrals) from file, brute force 4-dim array """
nninteraction = np.zeros([spOrbitals, spOrbitals, spOrbitals, spOrbitals])
with open("nucleitwobody.dat", "r") as infile:
    for line in infile:
        number = line.split()
        a = int(number[0]) - 1
        b = int(number[1]) - 1
        c = int(number[2]) - 1
        d = int(number[3]) - 1
        #print a, b, c, d, float(l[4])
        nninteraction[a][b][c][d] = Decimal(number[4])

""" Set up single-particle integral """
singleparticleH = np.zeros(spOrbitals)
for i in range(spOrbitals):
    singleparticleH[i] = Decimal(onebody(i, n, l))

""" Star HF-iterations, preparing variables and density matrix """

""" Coefficients for setting up density matrix, assuming only one along the diagonals """
C = np.eye(spOrbitals) # HF coefficients
DensityMatrix = np.zeros([spOrbitals, spOrbitals])
for gamma in range(spOrbitals):
    for delta in range(spOrbitals):
        sum = 0.0
        for i in range(Nparticles):
            sum += C[gamma][i]*C[delta][i]
        DensityMatrix[gamma][delta] = Decimal(sum)
maxHFiter = 100
epsilon = 1.0e-5
difference = 1.0
hf_count = 0
oldenergies = np.zeros(spOrbitals)
newenergies = np.zeros(spOrbitals)
while hf_count < maxHFiter and difference > epsilon:
    print "##### Iteration %i #####" % hf_count
    HFmatrix = np.zeros([spOrbitals, spOrbitals])
    for alpha in range(spOrbitals):
        for beta in range(spOrbitals):
            """ If tests for three-dimensional systems, including isospin conservation """
            if l[alpha] != l[beta] and j[alpha] != j[beta] and mj[alpha] != mj[beta]:
                """ Setting up the Fock matrix using the density matrix and antisymmetrization """
                sumFockTerm = 0.0
                for gamma in range(spOrbitals):
                    for delta in range(spOrbitals):
                        if (mj[alpha]+mj[gamma]) != (mj[beta]+mj[delta]) and (tz[alpha] != tz[beta]):
                            sumFockTerm += DensityMatrix[gamma][delta]*nninteraction[alpha][beta][gamma][delta]
            HFmatrix[alpha][beta] = Decimal(sumFockTerm)

```

```

        """ Adding the one-body term, here plain harmonic oscillator """
        if beta == alpha: HFmatrix[alpha][alpha] += singleparticleH[alpha]
    spenergies, C = np.linalg.eigh(HFmatrix)
    """ Setting up new density matrix in m-scheme """
    DensityMatrix = np.zeros([spOrbitals, spOrbitals])
    for gamma in range(spOrbitals):
        for delta in range(spOrbitals):
            sum = 0.0
            for i in range(Nparticles):
                sum += C[gamma][i]*C[delta][i]
            DensityMatrix[gamma][delta] = Decimal(sum)
    newenergies = spenergies
    """ Brute force computation of difference between previous and new sp HF energies """
    sum = 0.0
    for i in range(spOrbitals):
        sum += (abs(newenergies[i]-oldenergies[i]))/spOrbitals
    difference = sum
    oldenergies = newenergies
    print "Single-particle energies, ordering may have changed "
    for i in range(spOrbitals):
        print('{0:4d} {1:.4f}'.format(i, Decimal(oldenergies[i])))
    hf_count += 1

```

Practicalities with the Hartree-Fock code development, basis construction

When setting up the Hartree-Fock algorithm you will find it convenient to number the basis states by filling the lowest subshells. For the first two shells we could then have the following mapping.

$$\begin{aligned}
 |0\rangle &= \{n = 0, m = 0, m_s = -0.5\} \\
 |1\rangle &= \{n = 0, m = 0, m_s = 0.5\} \\
 |2\rangle &= \{n = 0, m = -1, m_s = -0.5\} \\
 |3\rangle &= \{n = 0, m = -1, m_s = 0.5\} \\
 |4\rangle &= \{n = 0, m = +1, m_s = -0.5\} \\
 |5\rangle &= \{n = 0, m = +1, m_s = 0.5\}
 \end{aligned}$$

Practicalities with the Hartree-Fock code development, two-body basis construction, brute force

In the setup of the two-body matrix elements we can typically opt between two alternatives. We can read the matrix elements from file or calculate the matrix elements on the fly. The latter requires simply that we call the abovementioned function for setting up the two-body matrix elements when we run our Hartree-Fock code.

If we however wish to read from file, we can store the matrix elements in two ways:

- Brute force
- Organize according to conserved two-body quantum numbers

The brute force way is easy to implement.

Two-body interaction, brute force part I

We can read in from file the two-body interaction matrix elements or compute once and for all and store in memory. The elements are

$$\langle pr|\hat{v}|qs\rangle.$$

The time-consuming part in the Hartree-Fock calculations involves the calculation of the two-body matrix. Furthermore, the storage of these matrix elements plays also an important role, in particular we wish to access the table of matrix elements as fast as possible.

Two-body interaction, brute force part II

In a brute force algorithm for storing the matrix elements, if we have d basis functions, we end up with the need of storing d^4 matrix elements. We can reduce this considerably by the following considerations. In the calculation of the two-body matrix elements $\langle pr|\hat{v}|qs\rangle$ we have the following symmetries

1. Invariance under permutations, that is

$$\langle pq|\hat{v}|rs\rangle = \langle qp|\hat{v}|sr\rangle.$$

1. The functions entering the evaluation of the integrals are all real, meaning that if we interchange $p \leftrightarrow q$ or $r \leftrightarrow s$, we end up with the same matrix element.

This reduces by a factor of eight the total number of matrix elements to be stored if we also use that we can store only for $p < q$ and $r < s$.

Two-body interaction, brute force part III

Furthermore, in setting up a table for the two-body matrix elements we can convert the need of using four indices $pqrs$ of

$$\langle pr|\hat{v}|qs\rangle,$$

which in a brute forces way could be coded as a four-dimensional array, to a two-dimensional array V_{lm} , where l and m stand for all possible two-body configurations pq .

Each number l and m in V_{lm} should then point to a set of single-particle states (p, q) and (r, s) .

In our case, since we have symmetries which allow us to set $p \leq q$, we have, with d single particle states a total of $d(d+1)/2$ two-body configurations.

Two-body interaction, brute force part IV

How do we store such a matrix? The simplest thing to do is to convert it into a one-dimensional array. How do we achieve that?

We now have a matrix V of dimension $n \times n$ and we want to store the elements V_{lm} as a one-dimensional array A using $0 \leq l \leq m \leq n-1$. For

1. $l = 0$ we have n elements
2. $l = 1$ we have $n-1$ elements
3. ...
4. $l = \nu$ we have $n-\nu$ elements
5. ...
6. $l = n-1$ we have 1 element,

and the total number is

$$\sum_{\nu=0}^{n-1} (n-\nu) = \frac{n(n+1)}{2}.$$

Two-body interaction, brute force part V

To find the number ($\text{number}(l, m)$) in a one-dimensional array A which corresponds to a matrix element V_{lm} , we note that

$$\text{number}(l, m) = \sum_{\nu=0}^{l-1} (n-\nu) + m - l = \frac{l(2n-l-1)}{2} + m.$$

The first matrix element $V(0, 0)$ is obviously given by the element $A(0)$.

We have thus reduced a four dimensional array to a one-dimensional array, where the given pairs (p, q) and (r, s) point to the matrix indices l and m , respectively. The latter are used to find the explicit number $\text{number}(l, m)$ which points to the desired matrix element stored in a one-dimensional array.

Practicalities with the Hartree-Fock code development, two-body basis construction

Another way to store the matrix elements is to organize the matrix elements according to conserved quantum numbers. This means setting up a block structure and to look up matrix elements using two-body conserved quantum numbers. For quantum dots, the two-body quantum numbers that are conserved are The total orbital momentum projection

$$M = m_1 + m_2,$$

and the total spin projection

$$M_s = m_{s_1} + m_{s_2}$$

Two-body basis construction

For 2 shells we have

M	M_s	α	State
-2	0	1	$ 3, 2\rangle$
-1	-1	3	$ 2, 0\rangle$
-1	0	4	$ 3, 0\rangle$
-1	0	4	$ 2, 1\rangle$
-1	1	5	$ 3, 1\rangle$
0	-1	6	$ 4, 2\rangle$
0	0	7	$ 1, 0\rangle$
0	0	7	$ 5, 2\rangle$
0	0	7	$ 4, 3\rangle$
0	1	8	$ 5, 3\rangle$
1	-1	9	$ 4, 0\rangle$
1	0	10	$ 5, 0\rangle$
1	0	10	$ 4, 1\rangle$
1	1	11	$ 5, 1\rangle$
2	0	13	$ 5, 4\rangle$

In this table we have paired orbitals with same M and M_s . Where $\alpha \in \{0, \dots, N\}$ and N is the number of pairs $\{M, M_s\}$. Notice that for $\alpha = 1$ we only have one pair, and for $\alpha = 4$, we have two pairs.

Hartree-Fock code

In developing our Hartree-Fock code, we can define a configuration class with the following mapping for a single-particle state

$$|\alpha\rangle \rightarrow |nmm_s\rangle,$$

and a similar mapping for a two-body state

$$|\alpha\beta\rangle \rightarrow |MM_s\rangle,$$

where M is the total angular momentum

$$M = m_\alpha + m_\beta,$$

and the M_s is the total spin

$$M_s = m_{s_1} + m_{s_2}.$$

Hartree-Fock code, setting up tables of matrix elements

The single-particle part is diagonal for the harmonic oscillator basis, that is

$$\langle\alpha|\hat{h}_0|\beta\rangle = \delta_{\alpha\beta}\epsilon_\alpha$$

with

$$\epsilon_\alpha = \hbar\omega(2n_\alpha + |m_\alpha| + 1).$$

The two-body matrix elements are diagonal in the M and M_s , that is

$$\langle MM_s|\hat{v}|M'M'_s\rangle = \delta_{M,M'}\delta_{M_s,M'_s}\langle MM_s|\hat{v}|MM_s\rangle.$$

Example calculations, $N = 6$ and $\omega = 1.0$ a.u.

We list here some selected Hartree-Fock results for $N = 6$ electrons as function of the number of major shells R included. Here we have $\omega = 1$ a.u.

R	E_0^{HF}
3	21.59320
4	20.76692
5	20.7484
6	20.72026
7	20.72013
8	20.71925
9	20.71925
10	20.71922
11	20.71922
12	20.71922
13	20.71922

The results are practically converged for approximately $R = 10 - 13$.

Example calculations, $N = 6$ and $\omega = 0.1$ a.u.

We list here some selected Hartree-Fock results for $N = 6$ electrons as function of the number of major shells R included. Here we have $\omega = 0.1$ a.u.

R	E_0^{HF}
4	4.01979
5	3.96315
6	3.87062
7	3.86314
8	3.85288
9	3.85259
10	3.85239
11	3.85239
12	3.85238
13	3.85238

Again, the results are practically converged for approximately $R = 10 - 13$.