Morten Hjorth-Jensen

Quantum mechanics for many-particle systems

From standard methods to quantum computing and machine learning

December 25, 2023

Springer

Contents

Part I Linear algebra and second quantization

1 Many-body Hamiltonians, basic linear algebra and Second Quantization 3

2 Second quantization 11

3 Full configuration interaction theory 27

Part I
Linear algebra and second quantization

Chapter 1
Many-body Hamiltonians, basic linear algebra and Second Quantization

Definitions and notations

Before we proceed we need some definitions. 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 the sum of some onebody part and a twobody part

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \sum_{i=1}^{A} \hat{h}_0(x_i) + \sum_{i<j}^{A} \hat{v}(r_{ij}), \tag{1.1}$$

with

$$H_0 = \sum_{i=1}^{A} \hat{h}_0(x_i). \tag{1.2}$$

The onebody part $u_{\mathrm{ext}}(x_i)$ 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 to be discussed here.

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

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

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 $\beta = -1$.

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

$$\Phi(x_1, x_2, \dots, x_A, \alpha, \beta, \dots, \sigma) = \frac{1}{\sqrt{A!}} \begin{vmatrix} \psi_\alpha(x_1) & \psi_\alpha(x_2) & \dots & \dots & \psi_\alpha(x_A) \\ \psi_\beta(x_1) & \psi_\beta(x_2) & \dots & \dots & \psi_\beta(x_A) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \psi_\sigma(x_1) & \psi_\sigma(x_2) & \dots & \dots & \psi_\sigma(x_A) \end{vmatrix}, \tag{1.3}$$

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

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

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

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

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

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

in a more compact form as

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

where $\hat{P}_i$ is a permutation operator which permutes the column indices $1,2,3,\ldots,n$ 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 $a_{11}a_{22}\ldots a_{nn}$ here.

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 $a_{11}a_{22}$.

Back to the derivation of the energy.

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

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

with eigenvalues

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

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.

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 $d\tau = dx_1 dr_2 \dots dr_A$.

In the Hartree-Fock method the trial function is the Slater determinant of Eq. (1.3) which can be rewritten as

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

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

It is defined as

$$\hat{A} = \frac{1}{A!}\sum_p (-)^p \hat{P}, \tag{1.4}$$

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

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

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

$$[H_0, \hat{A}] = [H_I, \hat{A}] = 0. \tag{1.5}$$

Furthermore, $\hat{A}$ satisfies

$$\hat{A}^2 = \hat{A}, \tag{1.6}$$

since every permutation of the Slater determinant reproduces it.

The expectation value of $\hat{H}_0$

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

is readily reduced to

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

where we have used Eqs. (1.5) and (1.6). The next step is to replace the antisymmetrization operator by its definition and to replace $\hat{H}_0$ with the sum of one-body operators

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

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

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

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

$$\int \Phi^* \hat{H}_0 \Phi d\tau = \sum_{\mu=1}^{A} \int \psi_\mu^*(x) \hat{h}_0 \psi_\mu(x) dx d\mathbf{r}. \tag{1.7}$$

We introduce the following shorthand for the above integral

$$\langle \mu | \hat{h}_0 | \mu \rangle = \int \psi_\mu^*(x) \hat{h}_0 \psi_\mu(x) dx,$$

and rewrite Eq. (1.7) as

$$\int \Phi^* \hat{H}_0 \Phi d\tau = \sum_{\mu=1}^{A} \langle \mu | \hat{h}_0 | \mu \rangle. \tag{1.8}$$

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

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

which reduces to

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

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

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

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

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.

We obtain

$$\int \Phi^* \hat{H}_I \Phi d\tau = \frac{1}{2} \sum_{\mu=1}^{A} \sum_{\nu=1}^{A} \Big[\int \psi_\mu^*(x_i) \psi_\nu^*(x_j) \hat{v}(r_{ij}) \psi_\mu(x_i) \psi_\nu(x_j) dx_i dx_j \tag{1.9}$$

$$- \int \psi_\mu^*(x_i) \psi_\nu^*(x_j) \hat{v}(r_{ij}) \psi_\nu(x_i) \psi_\mu(x_j) dx_i dx_j \Big]. \tag{1.10}$$

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.

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

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

We introduce the following shorthands for the above two integrals

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

and

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

Preparing for later studies: varying the coefficients of a wave function expansion and orthogonal transformations

It is common to 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 single-particle basis (this is a normal approach for Hartree-Fock theory) by performing a unitary transformation on our previous basis (labelled with greek indices) as

$$\psi^{new}_p = \sum_\lambda C_{p\lambda}\phi_\lambda. \tag{1.11}$$

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.

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

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

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

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

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 stated that a unitary transformation keeps the orthogonality. To see this consider first a basis of vectors $\mathbf{v}_i$,

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

We assume that the basis is orthogonal, that is

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

An orthogonal or unitary transformation

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

preserves the dot product and orthogonality since

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

This means that if the coefficients $C_{p\lambda}$ belong to a unitary or orthogonal trasformation (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 propertry is extremely useful when we build up a basis of many-body Stater 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.

Before we develop for example 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

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

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.

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 $n = 2$.

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 $\phi_\lambda(x)$.

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

The energy functional is

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

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. (1.11). We can then rewrite the energy functional as

$$E[\Phi^{New}] = \sum_{i=1}^{A} \langle i|h|i\rangle + \frac{1}{2}\sum_{ij=1}^{A} \langle ij|\hat{v}|ij\rangle_{AS}, \tag{1.12}$$

where Φ^{New} is the new Slater determinant defined by the new basis of Eq. (1.11).

Using Eq. (1.11) we can rewrite Eq. (1.12) as

$$E[\Psi] = \sum_{i=1}^{A}\sum_{\alpha\beta} C^*_{i\alpha}C_{i\beta}\langle \alpha|h|\beta\rangle + \frac{1}{2}\sum_{ij=1}^{A}\sum_{\alpha\beta\gamma\delta} C^*_{i\alpha}C^*_{j\beta}C_{i\gamma}C_{j\delta}\langle \alpha\beta|\hat{v}|\gamma\delta\rangle_{AS}. \tag{1.13}$$

Chapter 2
Second quantization

We introduce the time-independent operators $a^\dagger_\alpha$ and a_α which create and annihilate, respectively, a particle in the single-particle state φ_α. We define the fermion creation operator $a^\dagger_\alpha$

$$a^\dagger_\alpha|0\rangle \equiv |\alpha\rangle, \tag{2.1}$$

and

$$a^\dagger_\alpha|\alpha_1\dots\alpha_n\rangle_{\rm AS} \equiv |\alpha\alpha_1\dots\alpha_n\rangle_{\rm AS} \tag{2.2}$$

In Eq. (2.1) the operator $a^\dagger_\alpha$ acts on the vacuum state $|0\rangle$, which does not contain any particles. Alternatively, we could define a closed-shell nucleus or atom as our new vacuum, but then we need to introduce the particle-hole formalism, see the discussion to come.

In Eq. (2.2) $a^\dagger_\alpha$ acts on an antisymmetric n-particle state and creates an antisymmetric $(n+1)$-particle state, where the one-body state φ_α is occupied, under the condition that $\alpha \neq \alpha_1,\alpha_2,\dots,\alpha_n$. It follows that we can express an antisymmetric state as the product of the creation operators acting on the vacuum state.

$$|\alpha_1\dots\alpha_n\rangle_{\rm AS} = a^\dagger_{\alpha_1}a^\dagger_{\alpha_2}\dots a^\dagger_{\alpha_n}|0\rangle \tag{2.3}$$

It is easy to derive the commutation and anticommutation rules for the fermionic creation operators $a^\dagger_\alpha$. Using the antisymmetry of the states (2.3)

$$|\alpha_1\dots\alpha_i\dots\alpha_k\dots\alpha_n\rangle_{\rm AS} = -|\alpha_1\dots\alpha_k\dots\alpha_i\dots\alpha_n\rangle_{\rm AS} \tag{2.4}$$

we obtain

$$a^\dagger_{\alpha_i}a^\dagger_{\alpha_k} = -a^\dagger_{\alpha_k}a^\dagger_{\alpha_i} \tag{2.5}$$

Using the Pauli principle

$$|\alpha_1\dots\alpha_i\dots\alpha_i\dots\alpha_n\rangle_{\rm AS} = 0 \tag{2.6}$$

it follows that

$$a^\dagger_{\alpha_i}a^\dagger_{\alpha_i} = 0. \tag{2.7}$$

If we combine Eqs. (2.5) and (2.7), we obtain the well-known anti-commutation rule

$$a^\dagger_\alpha a^\dagger_\beta + a^\dagger_\beta a^\dagger_\alpha \equiv \{a^\dagger_\alpha, a^\dagger_\beta\} = 0 \tag{2.8}$$

The hermitian conjugate of $a^\dagger_\alpha$ is

$$a_\alpha = (a^\dagger_\alpha)^\dagger \tag{2.9}$$

If we take the hermitian conjugate of Eq. (2.8), we arrive at

$$\{a_\alpha, a_\beta\} = 0 \tag{2.10}$$

What is the physical interpretation of the operator a_α and what is the effect of a_α on a given state $|\alpha_1\alpha_2\dots\alpha_n\rangle_{\rm AS}$? Consider the following matrix element

$$\langle\alpha_1\alpha_2\dots\alpha_n|a_\alpha|\alpha'_1\alpha'_2\dots\alpha'_m\rangle \tag{2.11}$$

where both sides are antisymmetric. We distinguish between two cases. The first (1) is when $\alpha\in\{\alpha_i\}$. Using the Pauli principle of Eq. (2.6) it follows

$$\langle\alpha_1\alpha_2\dots\alpha_n|a_\alpha = 0 \tag{2.12}$$

The second (2) case is when $\alpha\notin\{\alpha_i\}$. It follows that an hermitian conjugation

$$\langle\alpha_1\alpha_2\dots\alpha_n|a_\alpha = \langle\alpha\alpha_1\alpha_2\dots\alpha_n| \tag{2.13}$$

Eq. (2.13) holds for case (1) since the lefthand side is zero due to the Pauli principle. We write Eq. (2.11) as

$$\langle\alpha_1\alpha_2\dots\alpha_n|a_\alpha|\alpha'_1\alpha'_2\dots\alpha'_m\rangle = \langle\alpha_1\alpha_2\dots\alpha_n|\alpha\alpha'_1\alpha'_2\dots\alpha'_m\rangle \tag{2.14}$$

Here we must have $m=n+1$ if Eq. (2.14) has to be trivially different from zero.

For the last case, the minus and plus signs apply when the sequence $\alpha,\alpha_1,\alpha_2,\dots,\alpha_n$ and $\alpha'_1,\alpha'_2,\dots,\alpha'_{n+1}$ are related to each other via even and odd permutations. If we assume that $\alpha\notin\{\alpha_i\}$ we obtain

$$\langle\alpha_1\alpha_2\dots\alpha_n|a_\alpha|\alpha'_1\alpha'_2\dots\alpha'_{n+1}\rangle = 0 \tag{2.15}$$

when $\alpha\in\{\alpha'_i\}$. If $\alpha\notin\{\alpha'_i\}$, we obtain

$$a_\alpha\underbrace{|\alpha'_1\alpha'_2\dots\alpha'_{n+1}\rangle}_{\neq\alpha} = 0 \tag{2.16}$$

and in particular

$$a_\alpha|0\rangle = 0 \tag{2.17}$$

If $\{\alpha\alpha_i\}=\{\alpha'_i\}$, performing the right permutations, the sequence $\alpha,\alpha_1,\alpha_2,\dots,\alpha_n$ is identical with the sequence $\alpha'_1,\alpha'_2,\dots,\alpha'_{n+1}$. This results in

$$\langle\alpha_1\alpha_2\dots\alpha_n|a_\alpha|\alpha\alpha_1\alpha_2\dots\alpha_n\rangle = 1 \tag{2.18}$$

and thus

$$a_\alpha|\alpha\alpha_1\alpha_2\dots\alpha_n\rangle = |\alpha_1\alpha_2\dots\alpha_n\rangle \tag{2.19}$$

The action of the operator a_α from the left on a state vector is to to remove one particle in the state α. If the state vector does not contain the single-particle state α, the outcome of the operation is zero. The operator a_α is normally called for a destruction or annihilation operator.

The next step is to establish the commutator algebra of $a^\dagger_\alpha$ and a_β.

The action of the anti-commutator $\{a^\dagger_\alpha,a_\alpha\}$ on a given n-particle state is

$$\begin{aligned} a^\dagger_\alpha a_\alpha\underbrace{|\alpha_1\alpha_2\dots\alpha_n\rangle}_{\neq\alpha} &= 0 \\ a_\alpha a^\dagger_\alpha\underbrace{|\alpha_1\alpha_2\dots\alpha_n\rangle}_{\neq\alpha} = a_\alpha\underbrace{|\alpha\alpha_1\alpha_2\dots\alpha_n\rangle}_{\neq\alpha} &= \underbrace{|\alpha_1\alpha_2\dots\alpha_n\rangle}_{\neq\alpha} \end{aligned} \tag{2.20}$$

if the single-particle state α is not contained in the state.

If it is present we arrive at

$$\begin{aligned} a_\alpha^\dagger a_\alpha|\alpha_1\alpha_2\ldots\alpha_k\alpha\alpha_{k+1}\ldots\alpha_{n-1}\rangle &= a_\alpha^\dagger a_\alpha(-1)^k|\alpha\alpha_1\alpha_2\ldots\alpha_{n-1}\rangle \\ &= (-1)^k|\alpha\alpha_1\alpha_2\ldots\alpha_{n-1}\rangle = |\alpha_1\alpha_2\ldots\alpha_k\alpha\alpha_{k+1}\ldots\alpha_{n-1}\rangle \\ a_\alpha a_\alpha^\dagger|\alpha_1\alpha_2\ldots\alpha_k\alpha\alpha_{k+1}\ldots\alpha_{n-1}\rangle &= 0 \end{aligned} \tag{2.21}$$

From Eqs. (2.20) and (2.21) we arrive at

$$\{a_\alpha^\dagger, a_\alpha\} = a_\alpha^\dagger a_\alpha + a_\alpha a_\alpha^\dagger = 1 \tag{2.22}$$

The action of $\left\{a_\alpha^\dagger, a_\beta\right\}$, with $\alpha \neq \beta$ on a given state yields three possibilities. The first case is a state vector which contains both α and β, then either α or β and finally none of them.

The first case results in

$$\begin{aligned} a_\alpha^\dagger a_\beta|\alpha\beta\alpha_1\alpha_2\ldots\alpha_{n-2}\rangle &= 0 \\ a_\beta a_\alpha^\dagger|\alpha\beta\alpha_1\alpha_2\ldots\alpha_{n-2}\rangle &= 0 \end{aligned} \tag{2.23}$$

while the second case gives

$$\begin{aligned} a_\alpha^\dagger a_\beta|\beta\underbrace{\alpha_1\alpha_2\ldots\alpha_{n-1}}_{\neq\alpha}\rangle &= |\alpha\underbrace{\alpha_1\alpha_2\ldots\alpha_{n-1}}_{\neq\alpha}\rangle \\ a_\beta a_\alpha^\dagger|\beta\underbrace{\alpha_1\alpha_2\ldots\alpha_{n-1}}_{\neq\alpha}\rangle &= a_\beta|\alpha\beta\underbrace{\beta\alpha_1\alpha_2\ldots\alpha_{n-1}}_{\neq\alpha}\rangle \\ &= -|\alpha\underbrace{\alpha_1\alpha_2\ldots\alpha_{n-1}}_{\neq\alpha}\rangle \end{aligned} \tag{2.24}$$

Finally if the state vector does not contain α and β

$$\begin{aligned} a_\alpha^\dagger a_\beta|\underbrace{\alpha_1\alpha_2\ldots\alpha_n}_{\neq\alpha,\beta}\rangle &= 0 \\ a_\beta a_\alpha^\dagger|\underbrace{\alpha_1\alpha_2\ldots\alpha_n}_{\neq\alpha,\beta}\rangle &= a_\beta|\alpha\underbrace{\alpha_1\alpha_2\ldots\alpha_n}_{\neq\alpha,\beta}\rangle = 0 \end{aligned} \tag{2.25}$$

For all three cases we have

$$\{a_\alpha^\dagger, a_\beta\} = a_\alpha^\dagger a_\beta + a_\beta a_\alpha^\dagger = 0, \quad \alpha \neq \beta \tag{2.26}$$

We can summarize our findings in Eqs. (2.22) and (2.26) as

$$\{a_\alpha^\dagger, a_\beta\} = \delta_{\alpha\beta} \tag{2.27}$$

with $\delta_{\alpha\beta}$ is the Kroenecker δ-symbol.

The properties of the creation and annihilation operators can be summarized as (for fermions)

$$a_\alpha^\dagger|0\rangle \equiv |\alpha\rangle,$$

and

$$a_\alpha^\dagger|\alpha_1\ldots\alpha_n\rangle_{\mathrm{AS}} \equiv |\alpha\alpha_1\ldots\alpha_n\rangle_{\mathrm{AS}}.$$

from which follows

$$|\alpha_1\ldots\alpha_n\rangle_{\mathrm{AS}} = a_{\alpha_1}^\dagger a_{\alpha_2}^\dagger\ldots a_{\alpha_n}^\dagger|0\rangle.$$

The hermitian conjugate has the folowing properties

$$a_\alpha = (a_\alpha^\dagger)^\dagger.$$

Finally we found

$$a_\alpha \underbrace{|\alpha_1'\alpha_2'\dots\alpha_{n+1}'\rangle}_{\neq\alpha} = 0, \quad \text{in particular } a_\alpha|0\rangle = 0,$$

and

$$a_\alpha|\alpha\alpha_1\alpha_2\dots\alpha_n\rangle = |\alpha_1\alpha_2\dots\alpha_n\rangle,$$

and the corresponding commutator algebra

$$\{a_\alpha^\dagger, a_\beta^\dagger\} = \{a_\alpha, a_\beta\} = 0 \qquad \{a_\alpha^\dagger, a_\beta\} = \delta_{\alpha\beta}.$$

One-body operators in second quantization

A very useful operator is the so-called number-operator. Most physics cases we will study in this text conserve the total number of particles. The number operator is therefore a useful quantity which allows us to test that our many-body formalism conserves the number of particles. In for example (d,p) or (p,d) reactions it is important to be able to describe quantum mechanical states where particles get added or removed. A creation operator $a_\alpha^\dagger$ adds one particle to the single-particle state α of a give many-body state vector, while an annihilation operator a_α removes a particle from a single-particle state α.

Let us consider an operator proportional with $a_\alpha^\dagger a_\beta$ and $\alpha = \beta$. It acts on an n-particle state resulting in

$$a_\alpha^\dagger a_\alpha|\alpha_1\alpha_2\dots\alpha_n\rangle = \begin{cases} 0 & \alpha \notin \{\alpha_i\} \\ \\ |\alpha_1\alpha_2\dots\alpha_n\rangle & \alpha \in \{\alpha_i\} \end{cases} \tag{2.28}$$

Summing over all possible one-particle states we arrive at

$$\left(\sum_\alpha a_\alpha^\dagger a_\alpha\right)|\alpha_1\alpha_2\dots\alpha_n\rangle = n|\alpha_1\alpha_2\dots\alpha_n\rangle \tag{2.29}$$

The operator

$$\hat{N} = \sum_\alpha a_\alpha^\dagger a_\alpha \tag{2.30}$$

is called the number operator since it counts the number of particles in a give state vector when it acts on the different single-particle states. It acts on one single-particle state at the time and falls therefore under category one-body operators. Next we look at another important one-body operator, namely $\hat{H}_0$ and study its operator form in the occupation number representation.

We want to obtain an expression for a one-body operator which conserves the number of particles. Here we study the one-body operator for the kinetic energy plus an eventual external one-body potential. The action of this operator on a particular n-body state with its pertinent expectation value has already been studied in coordinate space. In coordinate space the operator reads

$$\hat{H}_0 = \sum_i \hat{h}_0(x_i) \tag{2.31}$$

and the anti-symmetric n-particle Slater determinant is defined as

$$\Phi(x_1,x_2,\dots,x_n,\alpha_1,\alpha_2,\dots,\alpha_n) = \frac{1}{\sqrt{n!}}\sum_p (-1)^p \hat{P}\psi_{\alpha_1}(x_1)\psi_{\alpha_2}(x_2)\dots\psi_{\alpha_n}(x_n).$$

Defining

$$\hat{h}_0(x_i)\psi_{\alpha_i}(x_i) = \sum_{\alpha_k'} \psi_{\alpha_k'}(x_i)\langle\alpha_k'|\hat{h}_0|\alpha_k\rangle \tag{2.32}$$

we can easily evaluate the action of $\hat{H}_0$ on each product of one-particle functions in Slater determinant. From Eq. (2.32) we obtain the following result without permuting any particle pair

$$\begin{aligned}
& \left(\sum_i \hat{h}_0(x_i)\right)\psi_{\alpha_1}(x_1)\psi_{\alpha_2}(x_2)\dots\psi_{\alpha_n}(x_n) \\
= & \sum_{\alpha_1'}\langle\alpha_1'|\hat{h}_0|\alpha_1\rangle\psi_{\alpha_1'}(x_1)\psi_{\alpha_2}(x_2)\dots\psi_{\alpha_n}(x_n) \\
+ & \sum_{\alpha_2'}\langle\alpha_2'|\hat{h}_0|\alpha_2\rangle\psi_{\alpha_1}(x_1)\psi_{\alpha_2'}(x_2)\dots\psi_{\alpha_n}(x_n) \\
+ & \dots \\
+ & \sum_{\alpha_n'}\langle\alpha_n'|\hat{h}_0|\alpha_n\rangle\psi_{\alpha_1}(x_1)\psi_{\alpha_2}(x_2)\dots\psi_{\alpha_n'}(x_n)
\end{aligned} \tag{2.33}$$

If we interchange particles 1 and 2 we obtain

$$\begin{aligned}
& \left(\sum_i \hat{h}_0(x_i)\right)\psi_{\alpha_1}(x_2)\psi_{\alpha_1}(x_2)\dots\psi_{\alpha_n}(x_n) \\
= & \sum_{\alpha_2'}\langle\alpha_2'|\hat{h}_0|\alpha_2\rangle\psi_{\alpha_1}(x_2)\psi_{\alpha_2'}(x_1)\dots\psi_{\alpha_n}(x_n) \\
+ & \sum_{\alpha_1'}\langle\alpha_1'|\hat{h}_0|\alpha_1\rangle\psi_{\alpha_1'}(x_2)\psi_{\alpha_2}(x_1)\dots\psi_{\alpha_n}(x_n) \\
+ & \dots \\
+ & \sum_{\alpha_n'}\langle\alpha_n'|\hat{h}_0|\alpha_n\rangle\psi_{\alpha_1}(x_2)\psi_{\alpha_1}(x_2)\dots\psi_{\alpha_n'}(x_n)
\end{aligned} \tag{2.34}$$

We can continue by computing all possible permutations. We rewrite also our Slater determinant in its second quantized form and skip the dependence on the quantum numbers x_i. Summing up all contributions and taking care of all phases $(-1)^p$ we arrive at

$$\begin{aligned}
\hat{H}_0|\alpha_1,\alpha_2,\dots,\alpha_n\rangle = & \sum_{\alpha_1'}\langle\alpha_1'|\hat{h}_0|\alpha_1\rangle|\alpha_1'\alpha_2\dots\alpha_n\rangle \\
+ & \sum_{\alpha_2'}\langle\alpha_2'|\hat{h}_0|\alpha_2\rangle|\alpha_1\alpha_2'\dots\alpha_n\rangle \\
+ & \dots \\
+ & \sum_{\alpha_n'}\langle\alpha_n'|\hat{h}_0|\alpha_n\rangle|\alpha_1\alpha_2\dots\alpha_n'\rangle
\end{aligned} \tag{2.35}$$

In Eq. (2.35) we have expressed the action of the one-body operator of Eq. (2.31) on the n-body state in its second quantized form. This equation can be further manipulated if we use the properties of the creation and annihilation operator on each primed quantum number, that is

$$|\alpha_1\alpha_2\dots\alpha_k'\dots\alpha_n\rangle = a^\dagger_{\alpha_k'}a_{\alpha_k}|\alpha_1\alpha_2\dots\alpha_k\dots\alpha_n\rangle \tag{2.36}$$

Inserting this in the right-hand side of Eq. (2.35) results in

$$\begin{aligned}
\hat{H}_0|\alpha_1\alpha_2\dots\alpha_n\rangle = & \sum_{\alpha_1'}\langle\alpha_1'|\hat{h}_0|\alpha_1\rangle a^\dagger_{\alpha_1'}a_{\alpha_1}|\alpha_1\alpha_2\dots\alpha_n\rangle \\
+ & \sum_{\alpha_2'}\langle\alpha_2'|\hat{h}_0|\alpha_2\rangle a^\dagger_{\alpha_2'}a_{\alpha_2}|\alpha_1\alpha_2\dots\alpha_n\rangle \\
+ & \dots \\
+ & \sum_{\alpha_n'}\langle\alpha_n'|\hat{h}_0|\alpha_n\rangle a^\dagger_{\alpha_n'}a_{\alpha_n}|\alpha_1\alpha_2\dots\alpha_n\rangle \\
= & \sum_{\alpha,\beta}\langle\alpha|\hat{h}_0|\beta\rangle a^\dagger_\alpha a_\beta|\alpha_1\alpha_2\dots\alpha_n\rangle
\end{aligned} \tag{2.37}$$

In the number occupation representation or second quantization we get the following expression for a one-body operator which conserves the number of particles

$$\hat{H}_0 = \sum_{\alpha\beta}\langle\alpha|\hat{h}_0|\beta\rangle a^\dagger_\alpha a_\beta \tag{2.38}$$

Obviously, $\hat{H}_0$ can be replaced by any other one-body operator which preserved the number of particles. The stucture of the operator is therefore not limited to say the kinetic or single-particle energy only.

The opearator $\hat{H}_0$ takes a particle from the single-particle state β to the single-particle state α with a probability for the transition given by the expectation value $\langle\alpha|\hat{h}_0|\beta\rangle$.

It is instructive to verify Eq. (2.38) by computing the expectation value of $\hat{H}_0$ between two single-particle states

$$\langle\alpha_1|\hat{h}_0|\alpha_2\rangle = \sum_{\alpha\beta}\langle\alpha|\hat{h}_0|\beta\rangle\langle 0|a_{\alpha_1}a^\dagger_\alpha a_\beta a^\dagger_{\alpha_2}|0\rangle \tag{2.39}$$

Using the commutation relations for the creation and annihilation operators we have

$$a_{\alpha_1}a^\dagger_\alpha a_\beta a^\dagger_{\alpha_2} = (\delta_{\alpha\alpha_1} - a^\dagger_\alpha a_{\alpha_1})(\delta_{\beta\alpha_2} - a^\dagger_{\alpha_2}a_\beta), \tag{2.40}$$

which results in

$$\langle 0|a_{\alpha_1}a^\dagger_\alpha a_\beta a^\dagger_{\alpha_2}|0\rangle = \delta_{\alpha\alpha_1}\delta_{\beta\alpha_2} \tag{2.41}$$

and

$$\langle\alpha_1|\hat{h}_0|\alpha_2\rangle = \sum_{\alpha\beta}\langle\alpha|\hat{h}_0|\beta\rangle\delta_{\alpha\alpha_1}\delta_{\beta\alpha_2} = \langle\alpha_1|\hat{h}_0|\alpha_2\rangle \tag{2.42}$$

Two-body operators in second quantization

Let us now derive the expression for our two-body interaction part, which also conserves the number of particles. We can proceed in exactly the same way as for the one-body operator. In the coordinate representation our two-body interaction part takes the following expression

$$\hat{H}_I = \sum_{i<j}V(x_i,x_j) \tag{2.43}$$

where the summation runs over distinct pairs. The term V can be an interaction model for the nucleon-nucleon interaction or the interaction between two electrons. It can also include additional two-body interaction terms.

The action of this operator on a product of two single-particle functions is defined as

$$V(x_i,x_j)\psi_{\alpha_k}(x_i)\psi_{\alpha_l}(x_j) = \sum_{\alpha_k'\alpha_l'}\psi'_{\alpha_k}(x_i)\psi'_{\alpha_l}(x_j)\langle\alpha_k'\alpha_l'|\hat{v}|\alpha_k\alpha_l\rangle \tag{2.44}$$

We can now let $\hat{H}_I$ act on all terms in the linear combination for $|\alpha_1\alpha_2\ldots\alpha_n\rangle$. Without any permutations we have

$$\begin{aligned}
&\left(\sum_{i<j} V(x_i,x_j)\right)\psi_{\alpha_1}(x_1)\psi_{\alpha_2}(x_2)\ldots\psi_{\alpha_n}(x_n)\\
=\;&\sum_{\alpha_1'\alpha_2'}\langle\alpha_1'\alpha_2'|\hat{v}|\alpha_1\alpha_2\rangle\psi'_{\alpha_1}(x_1)\psi'_{\alpha_2}(x_2)\ldots\psi_{\alpha_n}(x_n)\\
+\;&\ldots\\
+\;&\sum_{\alpha_1'\alpha_n'}\langle\alpha_1'\alpha_n'|\hat{v}|\alpha_1\alpha_n\rangle\psi'_{\alpha_1}(x_1)\psi_{\alpha_2}(x_2)\ldots\psi'_{\alpha_n}(x_n)\\
+\;&\ldots\\
+\;&\sum_{\alpha_2'\alpha_n'}\langle\alpha_2'\alpha_n'|\hat{v}|\alpha_2\alpha_n\rangle\psi_{\alpha_1}(x_1)\psi'_{\alpha_2}(x_2)\ldots\psi'_{\alpha_n}(x_n)\\
+\;&\ldots
\end{aligned}\tag{2.45}$$

where on the rhs we have a term for each distinct pairs.

For the other terms on the rhs we obtain similar expressions and summing over all terms we obtain

$$\begin{aligned}
H_I|\alpha_1\alpha_2\ldots\alpha_n\rangle =\;&\sum_{\alpha_1',\alpha_2'}\langle\alpha_1'\alpha_2'|\hat{v}|\alpha_1\alpha_2\rangle|\alpha_1'\alpha_2'\ldots\alpha_n\rangle\\
+\;&\ldots\\
+\;&\sum_{\alpha_1',\alpha_n'}\langle\alpha_1'\alpha_n'|\hat{v}|\alpha_1\alpha_n\rangle|\alpha_1'\alpha_2\ldots\alpha_n'\rangle\\
+\;&\ldots\\
+\;&\sum_{\alpha_2',\alpha_n'}\langle\alpha_2'\alpha_n'|\hat{v}|\alpha_2\alpha_n\rangle|\alpha_1\alpha_2'\ldots\alpha_n'\rangle\\
+\;&\ldots
\end{aligned}\tag{2.46}$$

We introduce second quantization via the relation

$$\begin{aligned}
&a^\dagger_{\alpha_k'}a^\dagger_{\alpha_l'}a_{\alpha_l}a_{\alpha_k}|\alpha_1\alpha_2\ldots\alpha_k\ldots\alpha_l\ldots\alpha_n\rangle\\
=\;&(-1)^{k-1}(-1)^{l-2}a^\dagger_{\alpha_k'}a^\dagger_{\alpha_l'}a_{\alpha_l}a_{\alpha_k}|\alpha_k\alpha_l\underbrace{\alpha_1\alpha_2\ldots\alpha_n}_{\neq\alpha_k,\alpha_l}\rangle\\
=\;&(-1)^{k-1}(-1)^{l-2}|\alpha_k'\alpha_l'\underbrace{\alpha_1\alpha_2\ldots\alpha_n}_{\neq\alpha_k',\alpha_l'}\rangle\\
=\;&|\alpha_1\alpha_2\ldots\alpha_k'\ldots\alpha_l'\ldots\alpha_n\rangle
\end{aligned}\tag{2.47}$$

Inserting this in (2.46) gives

$$
\begin{aligned}
H_I|\alpha_1\alpha_2\ldots\alpha_n\rangle = & \sum_{\alpha_1',\alpha_2'}\langle\alpha_1'\alpha_2'|\hat{v}|\alpha_1\alpha_2\rangle a^\dagger_{\alpha_1'}a^\dagger_{\alpha_2'}a_{\alpha_2}a_{\alpha_1}|\alpha_1\alpha_2\ldots\alpha_n\rangle \\
+ & \ldots \\
= & \sum_{\alpha_1',\alpha_n'}\langle\alpha_1'\alpha_n'|\hat{v}|\alpha_1\alpha_n\rangle a^\dagger_{\alpha_1'}a^\dagger_{\alpha_n'}a_{\alpha_n}a_{\alpha_1}|\alpha_1\alpha_2\ldots\alpha_n\rangle \\
+ & \ldots \\
= & \sum_{\alpha_2',\alpha_n'}\langle\alpha_2'\alpha_n'|\hat{v}|\alpha_2\alpha_n\rangle a^\dagger_{\alpha_2'}a^\dagger_{\alpha_n'}a_{\alpha_n}a_{\alpha_2}|\alpha_1\alpha_2\ldots\alpha_n\rangle \\
+ & \ldots \\
= & \sum_{\alpha,\beta,\gamma,\delta}{}'\langle\alpha\beta|\hat{v}|\gamma\delta\rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma|\alpha_1\alpha_2\ldots\alpha_n\rangle \qquad (2.48)
\end{aligned}
$$

Here we let $\sum'$ indicate that the sums running over α and β run over all single-particle states, while the summations γ and δ run over all pairs of single-particle states. We wish to remove this restriction and since

$$\langle\alpha\beta|\hat{v}|\gamma\delta\rangle = \langle\beta\alpha|\hat{v}|\delta\gamma\rangle \qquad (2.49)$$

we get

$$
\begin{aligned}
\sum_{\alpha\beta}\langle\alpha\beta|\hat{v}|\gamma\delta\rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma = & \sum_{\alpha\beta}\langle\beta\alpha|\hat{v}|\delta\gamma\rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma \qquad (2.50)\\
= & \sum_{\alpha\beta}\langle\beta\alpha|\hat{v}|\delta\gamma\rangle a^\dagger_\beta a^\dagger_\alpha a_\gamma a_\delta \qquad (2.51)
\end{aligned}
$$

where we have used the anti-commutation rules.

Changing the summation indices α and β in (2.51) we obtain

$$\sum_{\alpha\beta}\langle\alpha\beta|\hat{v}|\gamma\delta\rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma = \sum_{\alpha\beta}\langle\alpha\beta|\hat{v}|\delta\gamma\rangle a^\dagger_\alpha a^\dagger_\beta a_\gamma a_\delta \qquad (2.52)$$

From this it follows that the restriction on the summation over γ and δ can be removed if we multiply with a factor $\frac{1}{2}$, resulting in

$$\hat{H}_I = \frac{1}{2}\sum_{\alpha\beta\gamma\delta}\langle\alpha\beta|\hat{v}|\gamma\delta\rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma \qquad (2.53)$$

where we sum freely over all single-particle states α, β, γ og δ.

With this expression we can now verify that the second quantization form of $\hat{H}_I$ in Eq. (2.53) results in the same matrix between two anti-symmetrized two-particle states as its corresponding coordinate space representation. We have

$$\langle\alpha_1\alpha_2|\hat{H}_I|\beta_1\beta_2\rangle = \frac{1}{2}\sum_{\alpha\beta\gamma\delta}\langle\alpha\beta|\hat{v}|\gamma\delta\rangle\langle 0|a_{\alpha_2}a_{\alpha_1}a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma a^\dagger_{\beta_1}a^\dagger_{\beta_2}|0\rangle. \qquad (2.54)$$

Using the commutation relations we get

$$
\begin{aligned}
& a_{\alpha_2} a_{\alpha_1} a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma a^\dagger_{\beta_1} a^\dagger_{\beta_2} \\
= \; & a_{\alpha_2} a_{\alpha_1} a^\dagger_\alpha a^\dagger_\beta (a_\delta \delta_{\gamma\beta_1} a^\dagger_{\beta_2} - a_\delta a^\dagger_{\beta_1} a_\gamma a^\dagger_{\beta_2}) \\
= \; & a_{\alpha_2} a_{\alpha_1} a^\dagger_\alpha a^\dagger_\beta (\delta_{\gamma\beta_1}\delta_{\delta\beta_2} - \delta_{\gamma\beta_1} a^\dagger_{\beta_2} a_\delta - a_\delta a^\dagger_{\beta_1}\delta_{\gamma\beta_2} + a_\delta a^\dagger_{\beta_1} a^\dagger_{\beta_2} a_\gamma) \\
= \; & a_{\alpha_2} a_{\alpha_1} a^\dagger_\alpha a^\dagger_\beta (\delta_{\gamma\beta_1}\delta_{\delta\beta_2} - \delta_{\gamma\beta_1} a^\dagger_{\beta_2} a_\delta \\
& - \delta_{\delta\beta_1}\delta_{\gamma\beta_2} + \delta_{\gamma\beta_2} a^\dagger_{\beta_1} a_\delta + a_\delta a^\dagger_{\beta_1} a^\dagger_{\beta_2} a_\gamma)
\end{aligned} \tag{2.55}
$$

The vacuum expectation value of this product of operators becomes

$$
\begin{aligned}
& \langle 0 | a_{\alpha_2} a_{\alpha_1} a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma a^\dagger_{\beta_1} a^\dagger_{\beta_2} | 0 \rangle \\
= \; & (\delta_{\gamma\beta_1}\delta_{\delta\beta_2} - \delta_{\delta\beta_1}\delta_{\gamma\beta_2}) \langle 0 | a_{\alpha_2} a_{\alpha_1} a^\dagger_\alpha a^\dagger_\beta | 0 \rangle \\
= \; & (\delta_{\gamma\beta_1}\delta_{\delta\beta_2} - \delta_{\delta\beta_1}\delta_{\gamma\beta_2})(\delta_{\alpha\alpha_1}\delta_{\beta\alpha_2} - \delta_{\beta\alpha_1}\delta_{\alpha\alpha_2})
\end{aligned} \tag{2.56}
$$

Insertion of Eq. (2.56) in Eq. (2.54) results in

$$
\begin{aligned}
\langle \alpha_1\alpha_2 | \hat{H}_I | \beta_1\beta_2 \rangle = \; & \frac{1}{2}\big[\langle \alpha_1\alpha_2 | \hat{v} | \beta_1\beta_2 \rangle - \langle \alpha_1\alpha_2 | \hat{v} | \beta_2\beta_1 \rangle \\
& - \langle \alpha_2\alpha_1 | \hat{v} | \beta_1\beta_2 \rangle + \langle \alpha_2\alpha_1 | \hat{v} | \beta_2\beta_1 \rangle \big] \\
= \; & \langle \alpha_1\alpha_2 | \hat{v} | \beta_1\beta_2 \rangle - \langle \alpha_1\alpha_2 | \hat{v} | \beta_2\beta_1 \rangle \\
= \; & \langle \alpha_1\alpha_2 | \hat{v} | \beta_1\beta_2 \rangle_{\mathrm{AS}}.
\end{aligned} \tag{2.57}
$$

The two-body operator can also be expressed in terms of the anti-symmetrized matrix elements we discussed previously as

$$
\begin{aligned}
\hat{H}_I = \; & \frac{1}{2} \sum_{\alpha\beta\gamma\delta} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma \\
= \; & \frac{1}{4} \sum_{\alpha\beta\gamma\delta} \left[\langle \alpha\beta | \hat{v} | \gamma\delta \rangle - \langle \alpha\beta | \hat{v} | \delta\gamma \rangle \right] a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma \\
= \; & \frac{1}{4} \sum_{\alpha\beta\gamma\delta} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle_{\mathrm{AS}} a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma
\end{aligned} \tag{2.58}
$$

The factors in front of the operator, either $\frac{1}{4}$ or $\frac{1}{2}$ tells whether we use antisymmetrized matrix elements or not.

We can now express the Hamiltonian operator for a many-fermion system in the occupation basis representation as

$$
H = \sum_{\alpha,\beta} \langle \alpha | \hat{t} + \hat{u}_{\mathrm{ext}} | \beta \rangle a^\dagger_\alpha a_\beta + \frac{1}{4} \sum_{\alpha\beta\gamma\delta} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma. \tag{2.59}
$$

This is the form we will use in the rest of these lectures, assuming that we work with anti-symmetrized two-body matrix elements.

Particle-hole formalism

Second quantization is a useful and elegant formalism for constructing many-body states and quantum mechanical operators. One can express and translate many physical processes into simple pictures such as Feynman diagrams. Expecation values of many-body states are also easily calculated. However, although the equations are seemingly easy to set up, from a prac-

tical point of view, that is the solution of Schroedinger's equation, there is no particular gain. The many-body equation is equally hard to solve, irrespective of representation. The cliche that there is no free lunch brings us down to earth again. Note however that a transformation to a particular basis, for cases where the interaction obeys specific symmetries, can ease the solution of Schroedinger's equation.

But there is at least one important case where second quantization comes to our rescue. It is namely easy to introduce another reference state than the pure vacuum $|0\rangle$, where all single-particle states are active. With many particles present it is often useful to introduce another reference state than the vacuum state$|0\rangle$. We will label this state $|c\rangle$ (c for core) and as we will see it can reduce considerably the complexity and thereby the dimensionality of the many-body problem. It allows us to sum up to infinite order specific many-body correlations. The particle-hole representation is one of these handy representations.

In the original particle representation these states are products of the creation operators $a^\dagger_{\alpha_i}$ acting on the true vacuum $|0\rangle$. Following Eq. (2.3) we have

$$|\alpha_1\alpha_2\dots\alpha_{n-1}\alpha_n\rangle = a^\dagger_{\alpha_1}a^\dagger_{\alpha_2}\dots a^\dagger_{\alpha_{n-1}}a^\dagger_{\alpha_n}|0\rangle \tag{2.60}$$

$$|\alpha_1\alpha_2\dots\alpha_{n-1}\alpha_n\alpha_{n+1}\rangle = a^\dagger_{\alpha_1}a^\dagger_{\alpha_2}\dots a^\dagger_{\alpha_{n-1}}a^\dagger_{\alpha_n}a^\dagger_{\alpha_{n+1}}|0\rangle \tag{2.61}$$

$$|\alpha_1\alpha_2\dots\alpha_{n-1}\rangle = a^\dagger_{\alpha_1}a^\dagger_{\alpha_2}\dots a^\dagger_{\alpha_{n-1}}|0\rangle \tag{2.62}$$

If we use Eq. (2.60) as our new reference state, we can simplify considerably the representation of this state

$$|c\rangle \equiv |\alpha_1\alpha_2\dots\alpha_{n-1}\alpha_n\rangle = a^\dagger_{\alpha_1}a^\dagger_{\alpha_2}\dots a^\dagger_{\alpha_{n-1}}a^\dagger_{\alpha_n}|0\rangle \tag{2.63}$$

The new reference states for the $n+1$ and $n-1$ states can then be written as

$$|\alpha_1\alpha_2\dots\alpha_{n-1}\alpha_n\alpha_{n+1}\rangle = (-1)^n a^\dagger_{\alpha_{n+1}}|c\rangle \equiv (-1)^n|\alpha_{n+1}\rangle_c \tag{2.64}$$

$$|\alpha_1\alpha_2\dots\alpha_{n-1}\rangle = (-1)^{n-1}a_{\alpha_n}|c\rangle \equiv (-1)^{n-1}|\alpha_{n-1}\rangle_c \tag{2.65}$$

The first state has one additional particle with respect to the new vacuum state $|c\rangle$ and is normally referred to as a one-particle state or one particle added to the many-body reference state. The second state has one particle less than the reference vacuum state $|c\rangle$ and is referred to as a one-hole state. When dealing with a new reference state it is often convenient to introduce new creation and annihilation operators since we have from Eq. (2.65)

$$a_\alpha|c\rangle \neq 0 \tag{2.66}$$

since α is contained in $|c\rangle$, while for the true vacuum we have $a_\alpha|0\rangle = 0$ for all α.

The new reference state leads to the definition of new creation and annihilation operators which satisfy the following relations

$$b_\alpha|c\rangle = 0 \tag{2.67}$$

$$\{b^\dagger_\alpha, b^\dagger_\beta\} = \{b_\alpha, b_\beta\} = 0$$

$$\{b^\dagger_\alpha, b_\beta\} = \delta_{\alpha\beta} \tag{2.68}$$

We assume also that the new reference state is properly normalized

$$\langle c|c\rangle = 1 \tag{2.69}$$

The physical interpretation of these new operators is that of so-called quasiparticle states. This means that a state defined by the addition of one extra particle to a reference state $|c\rangle$ may not necesseraly be interpreted as one particle coupled to a core. We define now new creation operators that act on a state α creating a new quasiparticle state

$$b_\alpha^\dagger|c\rangle = \begin{cases} a_\alpha^\dagger|c\rangle = |\alpha\rangle, & \alpha > F \\ \\ a_\alpha|c\rangle = |\alpha^{-1}\rangle, & \alpha \leq F \end{cases} \tag{2.70}$$

where F is the Fermi level representing the last occupied single-particle orbit of the new reference state $|c\rangle$.

The annihilation is the hermitian conjugate of the creation operator

$$b_\alpha = (b_\alpha^\dagger)^\dagger,$$

resulting in

$$b_\alpha^\dagger = \begin{cases} a_\alpha^\dagger & \alpha > F \\ \\ a_\alpha & \alpha \leq F \end{cases} \qquad b_\alpha = \begin{cases} a_\alpha & \alpha > F \\ \\ a_\alpha^\dagger & \alpha \leq F \end{cases} \tag{2.71}$$

With the new creation and annihilation operator we can now construct many-body quasi-particle states, with one-particle-one-hole states, two-particle-two-hole states etc in the same fashion as we previously constructed many-particle states. We can write a general particle-hole state as

$$|\beta_1\beta_2\dots\beta_{n_p}\gamma_1^{-1}\gamma_2^{-1}\dots\gamma_{n_h}^{-1}\rangle \equiv \underbrace{b_{\beta_1}^\dagger b_{\beta_2}^\dagger\dots b_{\beta_{n_p}}^\dagger}_{>F}\underbrace{b_{\gamma_1}^\dagger b_{\gamma_2}^\dagger\dots b_{\gamma_{n_h}}^\dagger}_{\leq F}|c\rangle \tag{2.72}$$

We can now rewrite our one-body and two-body operators in terms of the new creation and annihilation operators. The number operator becomes

$$\hat{N} = \sum_\alpha a_\alpha^\dagger a_\alpha = \sum_{\alpha>F} b_\alpha^\dagger b_\alpha + n_c - \sum_{\alpha\leq F} b_\alpha^\dagger b_\alpha \tag{2.73}$$

where n_c is the number of particle in the new vacuum state $|c\rangle$. The action of $\hat{N}$ on a many-body state results in

$$N|\beta_1\beta_2\dots\beta_{n_p}\gamma_1^{-1}\gamma_2^{-1}\dots\gamma_{n_h}^{-1}\rangle = (n_p + n_c - n_h)|\beta_1\beta_2\dots\beta_{n_p}\gamma_1^{-1}\gamma_2^{-1}\dots\gamma_{n_h}^{-1}\rangle \tag{2.74}$$

Here $n = n_p + n_c - n_h$ is the total number of particles in the quasi-particle state of Eq. (2.72). Note that $\hat{N}$ counts the total number of particles present

$$N_{qp} = \sum_\alpha b_\alpha^\dagger b_\alpha, \tag{2.75}$$

gives us the number of quasi-particles as can be seen by computing

$$N_{qp} = |\beta_1\beta_2\dots\beta_{n_p}\gamma_1^{-1}\gamma_2^{-1}\dots\gamma_{n_h}^{-1}\rangle = (n_p + n_h)|\beta_1\beta_2\dots\beta_{n_p}\gamma_1^{-1}\gamma_2^{-1}\dots\gamma_{n_h}^{-1}\rangle \tag{2.76}$$

where $n_{qp} = n_p + n_h$ is the total number of quasi-particles.

We express the one-body operator $\hat{H}_0$ in terms of the quasi-particle creation and annihilation operators, resulting in

$$\begin{aligned} \hat{H}_0 = & \sum_{\alpha\beta>F}\langle\alpha|\hat{h}_0|\beta\rangle b_\alpha^\dagger b_\beta + \sum_{\alpha>F,\beta\leq F}\left[\langle\alpha|\hat{h}_0|\beta\rangle b_\alpha^\dagger b_\beta^\dagger + \langle\beta|\hat{h}_0|\alpha\rangle b_\beta b_\alpha\right] \\ + & \sum_{\alpha\leq F}\langle\alpha|\hat{h}_0|\alpha\rangle - \sum_{\alpha\beta\leq F}\langle\beta|\hat{h}_0|\alpha\rangle b_\alpha^\dagger b_\beta \end{aligned} \tag{2.77}$$

The first term gives contribution only for particle states, while the last one contributes only for holestates. The second term can create or destroy a set of quasi-particles and the third term is the contribution from the vacuum state $|c\rangle$.

Before we continue with the expressions for the two-body operator, we introduce a nomenclature we will use for the rest of this text. It is inspired by the notation used in quantum chemistry. We reserve the labels $i,j,k,\ldots$ for hole states and $a,b,c,\ldots$ for states above F, viz. particle states. This means also that we will skip the constraint $\leq F$ or $> F$ in the summation symbols. Our operator $\hat{H}_0$ reads now

$$\begin{aligned}\hat{H}_0 = & \sum_{ab}\langle a|\hat{h}|b\rangle b_a^\dagger b_b + \sum_{ai}\left[\langle a|\hat{h}|i\rangle b_a^\dagger b_i^\dagger + \langle i|\hat{h}|a\rangle b_i b_a\right] \\ + & \sum_i \langle i|\hat{h}|i\rangle - \sum_{ij}\langle j|\hat{h}|i\rangle b_i^\dagger b_j \end{aligned} \tag{2.78}$$

The two-particle operator in the particle-hole formalism is more complicated since we have to translate four indices $\alpha\beta\gamma\delta$ to the possible combinations of particle and hole states. When performing the commutator algebra we can regroup the operator in five different terms

$$\hat{H}_I = \hat{H}_I^{(a)} + \hat{H}_I^{(b)} + \hat{H}_I^{(c)} + \hat{H}_I^{(d)} + \hat{H}_I^{(e)} \tag{2.79}$$

Using anti-symmetrized matrix elements, bthe term $\hat{H}_I^{(a)}$ is

$$\hat{H}_I^{(a)} = \frac{1}{4}\sum_{abcd}\langle ab|\hat{V}|cd\rangle b_a^\dagger b_b^\dagger b_d b_c \tag{2.80}$$

The next term $\hat{H}_I^{(b)}$ reads

$$\hat{H}_I^{(b)} = \frac{1}{4}\sum_{abci}\left(\langle ab|\hat{V}|ci\rangle b_a^\dagger b_b^\dagger b_i^\dagger b_c + \langle ai|\hat{V}|cb\rangle b_a^\dagger b_i b_b b_c\right) \tag{2.81}$$

This term conserves the number of quasiparticles but creates or removes a three-particle-one-hole state. For $\hat{H}_I^{(c)}$ we have

$$\begin{aligned}\hat{H}_I^{(c)} = & \frac{1}{4}\sum_{abij}\left(\langle ab|\hat{V}|ij\rangle b_a^\dagger b_b^\dagger b_j^\dagger b_i^\dagger + \langle ij|\hat{V}|ab\rangle b_a b_b b_j b_i\right) + \\ & \frac{1}{2}\sum_{abij}\langle ai|\hat{V}|bj\rangle b_a^\dagger b_j^\dagger b_b b_i + \frac{1}{2}\sum_{abi}\langle ai|\hat{V}|bi\rangle b_a^\dagger b_b. \end{aligned} \tag{2.82}$$

The first line stands for the creation of a two-particle-two-hole state, while the second line represents the creation to two one-particle-one-hole pairs while the last term represents a contribution to the particle single-particle energy from the hole states, that is an interaction between the particle states and the hole states within the new vacuum state. The fourth term reads

$$\begin{aligned}\hat{H}_I^{(d)} = & \frac{1}{4}\sum_{aijk}\left(\langle ai|\hat{V}|jk\rangle b_a^\dagger b_k^\dagger b_j^\dagger b_i + \langle ji|\hat{V}|ak\rangle b_k^\dagger b_j b_i b_a\right) + \\ & \frac{1}{4}\sum_{aij}\left(\langle ai|\hat{V}|ji\rangle b_a^\dagger b_j^\dagger + \langle ji|\hat{V}|ai\rangle - \langle ji|\hat{V}|ia\rangle b_j b_a\right). \end{aligned} \tag{2.83}$$

The terms in the first line stand for the creation of a particle-hole state interacting with hole states, we will label this as a two-hole-one-particle contribution. The remaining terms are a particle-hole state interacting with the holes in the vacuum state. Finally we have

$$\hat{H}_I^{(e)} = \frac{1}{4}\sum_{ijkl}\langle kl|\hat{V}|ij\rangle b_i^\dagger b_j^\dagger b_l b_k + \frac{1}{2}\sum_{ijk}\langle ij|\hat{V}|kj\rangle b_k^\dagger b_i + \frac{1}{2}\sum_{ij}\langle ij|\hat{V}|ij\rangle \tag{2.84}$$

The first terms represents the interaction between two holes while the second stands for the interaction between a hole and the remaining holes in the vacuum state. It represents a contribution to single-hole energy to first order. The last term collects all contributions to the energy of the ground state of a closed-shell system arising from hole-hole correlations.

Summarizing and defining a normal-ordered Hamiltonian

$$\Phi_{AS}(\alpha_1,\dots,\alpha_A;x_1,\dots x_A) = \frac{1}{\sqrt{A}}\sum_{\hat{P}}(-1)^P\hat{P}\prod_{i=1}^{A}\psi_{\alpha_i}(x_i),$$

which is equivalent with $|\alpha_1\dots\alpha_A\rangle = a^\dagger_{\alpha_1}\dots a^\dagger_{\alpha_A}|0\rangle$. We have also

$$a^\dagger_p|0\rangle = |p\rangle, \quad a_p|q\rangle = \delta_{pq}|0\rangle$$

$$\delta_{pq} = \left\{a_p, a^\dagger_q\right\},$$

and

$$0 = \left\{a^\dagger_p, a_q\right\} = \left\{a_p, a_q\right\} = \left\{a^\dagger_p, a^\dagger_q\right\}$$
$$|\Phi_0\rangle = |\alpha_1\dots\alpha_A\rangle, \quad \alpha_1,\dots,\alpha_A \le \alpha_F$$

$$\left\{a^\dagger_p, a_q\right\} = \delta_{pq}, p,q \le \alpha_F$$
$$\left\{a_p, a^\dagger_q\right\} = \delta_{pq}, p,q > \alpha_F$$

with $i,j,\dots \le \alpha_F, \quad a,b,\dots > \alpha_F, \quad p,q,\dots -$ any

$$a_i|\Phi_0\rangle = |\Phi_i\rangle, \qquad a^\dagger_a|\Phi_0\rangle = |\Phi^a\rangle$$

and

$$a^\dagger_i|\Phi_0\rangle = 0 \qquad a_a|\Phi_0\rangle = 0$$

The one-body operator is defined as

$$\hat{F} = \sum_{pq}\langle p|\hat{f}|q\rangle a^\dagger_p a_q$$

while the two-body opreator is defined as

$$\hat{V} = \frac{1}{4}\sum_{pqrs}\langle pq|\hat{v}|rs\rangle_{AS} a^\dagger_p a^\dagger_q a_s a_r$$

where we have defined the antisymmetric matrix elements

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

We can also define a three-body operator

$$\hat{V}_3 = \frac{1}{36}\sum_{pqrstu}\langle pqr|\hat{v}_3|stu\rangle_{AS} a^\dagger_p a^\dagger_q a^\dagger_r a_u a_t a_s$$

with the antisymmetrized matrix element

$$\langle pqr|\hat{v}_3|stu\rangle_{AS} = \langle pqr|\hat{v}_3|stu\rangle + \langle pqr|\hat{v}_3|tus\rangle + \langle pqr|\hat{v}_3|ust\rangle - \langle pqr|\hat{v}_3|sut\rangle - \langle pqr|\hat{v}_3|tsu\rangle - \langle pqr|\hat{v}_3|uts\rangle. \tag{2.85}$$

Operators in second quantization

In the build-up of a shell-model or FCI code that is meant to tackle large dimensionalities is the action of the Hamiltonian $\hat{H}$ on a Slater determinant represented in second quantization as

$$|\alpha_1 \dots \alpha_n\rangle = a^\dagger_{\alpha_1} a^\dagger_{\alpha_2} \dots a^\dagger_{\alpha_n}|0\rangle.$$

The time consuming part stems from the action of the Hamiltonian on the above determinant,

$$\left(\sum_{\alpha\beta}\langle\alpha|t+u|\beta\rangle a^\dagger_\alpha a_\beta + \frac{1}{4}\sum_{\alpha\beta\gamma\delta}\langle\alpha\beta|\hat{v}|\gamma\delta\rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma\right) a^\dagger_{\alpha_1} a^\dagger_{\alpha_2} \dots a^\dagger_{\alpha_n}|0\rangle.$$

A practically useful way to implement this action is to encode a Slater determinant as a bit pattern.

Assume that we have at our disposal n different single-particle orbits $\alpha_0, \alpha_2, \dots, \alpha_{n-1}$ and that we can distribute among these orbits $N \leq n$ particles.

A Slater determinant can then be coded as an integer of n bits. As an example, if we have $n = 16$ single-particle states $\alpha_0, \alpha_1, \dots, \alpha_{15}$ and $N = 4$ fermions occupying the states α_3, α_6, α_{10} and α_{13} we could write this Slater determinant as

$$\Phi_\Lambda = a^\dagger_{\alpha_3} a^\dagger_{\alpha_6} a^\dagger_{\alpha_{10}} a^\dagger_{\alpha_{13}}|0\rangle.$$

The unoccupied single-particle states have bit value 0 while the occupied ones are represented by bit state 1. In the binary notation we would write this 16 bits long integer as

α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}
0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0

which translates into the decimal number

$$2^3 + 2^6 + 2^{10} + 2^{13} = 9288.$$

We can thus encode a Slater determinant as a bit pattern.

With N particles that can be distributed over n single-particle states, the total number of Slater determinats (and defining thereby the dimensionality of the system) is

$$\dim(\mathscr{H}) = \binom{n}{N}.$$

The total number of bit patterns is 2^n.

We assume again that we have at our disposal n different single-particle orbits $\alpha_0, \alpha_2, \dots, \alpha_{n-1}$ and that we can distribute among these orbits $N \leq n$ particles. The ordering among these states is important as it defines the order of the creation operators. We will write the determinant

$$\Phi_\Lambda = a^\dagger_{\alpha_3} a^\dagger_{\alpha_6} a^\dagger_{\alpha_{10}} a^\dagger_{\alpha_{13}}|0\rangle,$$

in a more compact way as

$$\Phi_{3,6,10,13} = |0001001000100100\rangle.$$

The action of a creation operator is thus

$$a^\dagger_{\alpha_4}\Phi_{3,6,10,13} = a^\dagger_{\alpha_4}|0001001000100100\rangle = a^\dagger_{\alpha_4}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_3}a^\dagger_{\alpha_4}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = -|0001101000100100\rangle.$$

Similarly

$$a^\dagger_{\alpha_6}\Phi_{3,6,10,13} = a^\dagger_{\alpha_6}|0001001000100100\rangle = a^\dagger_{\alpha_6}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_4}(a^\dagger_{\alpha_6})^2a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = 0!$$

This gives a simple recipe:

- If one of the bits b_j is 1 and we act with a creation operator on this bit, we return a null vector
- If $b_j = 0$, we set it to 1 and return a sign factor $(-1)^l$, where l is the number of bits set before bit j.

Consider the action of $a^\dagger_{\alpha_2}$ on various slater determinants:

$$\begin{aligned}
a^\dagger_{\alpha_2}\Phi_{00111} &= a^\dagger_{\alpha_2}|00111\rangle &&= 0\times|00111\rangle\\
a^\dagger_{\alpha_2}\Phi_{01011} &= a^\dagger_{\alpha_2}|01011\rangle &&= (-1)\times|01111\rangle\\
a^\dagger_{\alpha_2}\Phi_{01101} &= a^\dagger_{\alpha_2}|01101\rangle &&= 0\times|01101\rangle\\
a^\dagger_{\alpha_2}\Phi_{01110} &= a^\dagger_{\alpha_2}|01110\rangle &&= 0\times|01110\rangle\\
a^\dagger_{\alpha_2}\Phi_{10011} &= a^\dagger_{\alpha_2}|10011\rangle &&= (-1)\times|10111\rangle\\
a^\dagger_{\alpha_2}\Phi_{10101} &= a^\dagger_{\alpha_2}|10101\rangle &&= 0\times|10101\rangle\\
a^\dagger_{\alpha_2}\Phi_{10110} &= a^\dagger_{\alpha_2}|10110\rangle &&= 0\times|10110\rangle\\
a^\dagger_{\alpha_2}\Phi_{11001} &= a^\dagger_{\alpha_2}|11001\rangle &&= (+1)\times|11101\rangle\\
a^\dagger_{\alpha_2}\Phi_{11010} &= a^\dagger_{\alpha_2}|11010\rangle &&= (+1)\times|11110\rangle
\end{aligned}$$

What is the simplest way to obtain the phase when we act with one annihilation(creation) operator on the given Slater determinant representation?

We have an SD representation

$$\Phi_\Lambda = a^\dagger_{\alpha_0}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

in a more compact way as

$$\Phi_{0,3,6,10,13} = |1001001000100100\rangle.$$

The action of

$$a^\dagger_{\alpha_4}a_{\alpha_0}\Phi_{0,3,6,10,13} = a^\dagger_{\alpha_4}|0001001000100100\rangle = a^\dagger_{\alpha_4}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_3}a^\dagger_{\alpha_4}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = -|0001101000100100\rangle.$$

The action

$$a_{\alpha_0}\Phi_{0,3,6,10,13} = |0001001000100100\rangle,$$

can be obtained by subtracting the logical sum (AND operation) of $\Phi_{0,3,6,10,13}$ and a word which represents only α_0, that is

$$|1000000000000000\rangle,$$

from $\Phi_{0,3,6,10,13} = |1001001000100100\rangle$.

This operation gives $|0001001000100100\rangle$.

Similarly, we can form $a^{\dagger}_{\alpha_4}a_{\alpha_0}\Phi_{0,3,6,10,13}$, say, by adding $|0000100000000000\rangle$ to $a_{\alpha_0}\Phi_{0,3,6,10,13}$, first checking that their logical sum is zero in order to make sure that orbital α_4 is not already occupied.

It is trickier however to get the phase $(-1)^l$. One possibility is as follows

- Let S_1 be a word that represents the $1-$bit to be removed and all others set to zero.

In the previous example $S_1 = |1000000000000000\rangle$

- Define S_2 as the similar word that represents the bit to be added, that is in our case

$S_2 = |0000100000000000\rangle$.

- Compute then $S = S_1 - S_2$, which here becomes

$$S = |0111000000000000\rangle$$

- Perform then the logical AND operation of S with the word containing

$$\Phi_{0,3,6,10,13} = |1001001000100100\rangle,$$

which results in $|0001000000000000\rangle$. Counting the number of $1-$bits gives the phase. Here you need however an algorithm for bitcounting. Several efficient ones available.

Chapter 3
Full configuration interaction theory

Slater determinants as basis states, Repetition

The simplest possible choice for many-body wavefunctions are **product** wavefunctions. That is

$$\Psi(x_1,x_2,x_3,\ldots,x_A) \approx \phi_1(x_1)\phi_2(x_2)\phi_3(x_3)\ldots$$

because we are really only good at thinking about one particle at a time. Such product wavefunctions, without correlations, are easy to work with; for example, if the single-particle states $\phi_i(x)$ are orthonormal, then the product wavefunctions are easy to orthonormalize.

Similarly, computing matrix elements of operators are relatively easy, because the integrals factorize.

The price we pay is the lack of correlations, which we must build up by using many, many product wavefunctions. (Thus we have a trade-off: compact representation of correlations but difficult integrals versus easy integrals but many states required.)

Slater determinants as basis states, repetition

Because we have fermions, we are required to have antisymmetric wavefunctions, e.g.

$$\Psi(x_1,x_2,x_3,\ldots,x_A) = -\Psi(x_2,x_1,x_3,\ldots,x_A)$$

etc. This is accomplished formally by using the determinantal formalism

$$\Psi(x_1,x_2,\ldots,x_A) = \frac{1}{\sqrt{A!}}|\begin{vmatrix} \phi_1(x_1) & \phi_1(x_2) & \ldots & \phi_1(x_A) \\ \phi_2(x_1) & \phi_2(x_2) & \ldots & \phi_2(x_A) \\ \vdots & & & \\ \phi_A(x_1) & \phi_A(x_2) & \ldots & \phi_A(x_A) \end{vmatrix}|$$

Product wavefunction + antisymmetry = Slater determinant.

Slater determinants as basis states

$$\Psi(x_1,x_2,\ldots,x_A) = \frac{1}{\sqrt{A!}} \left| \begin{array}{cccc} \phi_1(x_1) & \phi_1(x_2) & \ldots & \phi_1(x_A) \\ \phi_2(x_1) & \phi_2(x_2) & \ldots & \phi_2(x_A) \\ & \vdots & & \\ \phi_A(x_1) & \phi_A(x_2) & \ldots & \phi_A(x_A) \end{array} \right|$$

Properties of the determinant (interchange of any two rows or any two columns yields a change in sign; thus no two rows and no two columns can be the same) lead to the Pauli principle:

- No two particles can be at the same place (two columns the same); and
- No two particles can be in the same state (two rows the same).

Slater determinants as basis states

As a practical matter, however, Slater determinants beyond $N = 4$ quickly become unwieldy. Thus we turn to the **occupation representation** or **second quantization** to simplify calculations.

The occupation representation or number representation, using fermion **creation** and **annihilation** operators, is compact and efficient. It is also abstract and, at first encounter, not easy to internalize. It is inspired by other operator formalism, such as the ladder operators for the harmonic oscillator or for angular momentum, but unlike those cases, the operators **do not have coordinate space representations**.

Instead, one can think of fermion creation/annihilation operators as a game of symbols that compactly reproduces what one would do, albeit clumsily, with full coordinate-space Slater determinants.

Quick repetition of the occupation representation

We start with a set of orthonormal single-particle states $\{\phi_i(x)\}$. (Note: this requirement, and others, can be relaxed, but leads to a more involved formalism.) **Any** orthonormal set will do.

To each single-particle state $\phi_i(x)$ we associate a creation operator $\hat{a}_i^\dagger$ and an annihilation operator $\hat{a}_i$.

When acting on the vacuum state $|0\rangle$, the creation operator $\hat{a}_i^\dagger$ causes a particle to occupy the single-particle state $\phi_i(x)$:

$$\phi_i(x) \rightarrow \hat{a}_i^\dagger |0\rangle$$

Quick repetition of the occupation representation

But with multiple creation operators we can occupy multiple states:

$$\phi_i(x)\phi_j(x')\phi_k(x'') \rightarrow \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_k^\dagger |0\rangle.$$

Now we impose antisymmetry, by having the fermion operators satisfy **anticommutation relations**:

$$\hat{a}_i^\dagger \hat{a}_j^\dagger + \hat{a}_j^\dagger \hat{a}_i^\dagger = [\hat{a}_i^\dagger, \hat{a}_j^\dagger]_+ = \{\hat{a}_i^\dagger, \hat{a}_j^\dagger\} = 0$$

so that

$$\hat{a}_i^\dagger \hat{a}_j^\dagger = -\hat{a}_j^\dagger \hat{a}_i^\dagger$$

Quick repetition of the occupation representation

Because of this property, automatically $\hat{a}_i^\dagger \hat{a}_i^\dagger = 0$, enforcing the Pauli exclusion principle. Thus when writing a Slater determinant using creation operators,

$$\hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_k^\dagger \ldots |0\rangle$$

each index $i, j, k, \ldots$ must be unique.

For some relevant exercises with solutions see chapter 8 of Lecture Notes in Physics, volume 936.

Full Configuration Interaction Theory

We have defined the ansatz for the ground state as

$$|\Phi_0\rangle = \left(\prod_{i \leq F} \hat{a}_i^\dagger \right) |0\rangle,$$

where the index i defines different single-particle states up to the Fermi level. We have assumed that we have N fermions. A given one-particle-one-hole ($1p1h$) state can be written as

$$|\Phi_i^a\rangle = \hat{a}_a^\dagger \hat{a}_i |\Phi_0\rangle,$$

while a $2p2h$ state can be written as

$$|\Phi_{ij}^{ab}\rangle = \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i |\Phi_0\rangle,$$

and a general $NpNh$ state as

$$|\Phi_{ijk\ldots}^{abc\ldots}\rangle = \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_c^\dagger \ldots \hat{a}_k \hat{a}_j \hat{a}_i |\Phi_0\rangle.$$

Full Configuration Interaction Theory

We can then expand our exact state function for the ground state as

$$|\Psi_0\rangle = C_0|\Phi_0\rangle + \sum_{ai} C_i^a|\Phi_i^a\rangle + \sum_{abij} C_{ij}^{ab}|\Phi_{ij}^{ab}\rangle + \cdots = (C_0 + \hat{C})|\Phi_0\rangle,$$

where we have introduced the so-called correlation operator

$$\hat{C} = \sum_{ai} C_i^a \hat{a}_a^\dagger \hat{a}_i + \sum_{abij} C_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i + \ldots$$

Since the normalization of Ψ_0 is at our disposal and since C_0 is by hypothesis non-zero, we may arbitrarily set $C_0 = 1$ with corresponding proportional changes in all other coefficients. Using this so-called intermediate normalization we have

$$\langle\Psi_0|\Phi_0\rangle = \langle\Phi_0|\Phi_0\rangle = 1,$$

resulting in

$$|\Psi_0\rangle = (1 + \hat{C})|\Phi_0\rangle.$$

Full Configuration Interaction Theory

We rewrite

$$|\Psi_0\rangle = C_0|\Phi_0\rangle + \sum_{ai} C_i^a|\Phi_i^a\rangle + \sum_{abij} C_{ij}^{ab}|\Phi_{ij}^{ab}\rangle + \ldots,$$

in a more compact form as

$$|\Psi_0\rangle = \sum_{PH} C_H^P \Phi_H^P = \left(\sum_{PH} C_H^P \hat{A}_H^P\right)|\Phi_0\rangle,$$

where H stands for $0, 1, \ldots, n$ hole states and P for $0, 1, \ldots, n$ particle states. Our requirement of unit normalization gives

$$\langle\Psi_0|\Phi_0\rangle = \sum_{PH} |C_H^P|^2 = 1,$$

and the energy can be written as

$$E = \langle\Psi_0|\hat{H}|\Phi_0\rangle = \sum_{PP'HH'} C_H^{*P} \langle\Phi_H^P|\hat{H}|\Phi_{H'}^{P'}\rangle C_{H'}^{P'}.$$

Full Configuration Interaction Theory

Normally

$$E = \langle \Psi_0 | \hat{H} | \Phi_0 \rangle = \sum_{PP'HH'} C_H^{*P} \langle \Phi_H^P | \hat{H} | \Phi_{H'}^{P'} \rangle C_{H'}^{P'},$$

is solved by diagonalization setting up the Hamiltonian matrix defined by the basis of all possible Slater determinants. A diagonalization is equivalent to finding the variational minimum of

$$\langle \Psi_0 | \hat{H} | \Phi_0 \rangle - \lambda \langle \Psi_0 | \Phi_0 \rangle,$$

where λ is a variational multiplier to be identified with the energy of the system. The minimization process results in

$$\delta \left[\langle \Psi_0 | \hat{H} | \Phi_0 \rangle - \lambda \langle \Psi_0 | \Phi_0 \rangle \right] =$$

$$\sum_{P'H'} \left\{ \delta[C_H^{*P}] \langle \Phi_H^P | \hat{H} | \Phi_{H'}^{P'} \rangle C_{H'}^{P'} + C_H^{*P} \langle \Phi_H^P | \hat{H} | \Phi_{H'}^{P'} \rangle \delta[C_{H'}^{P'}] - \lambda (\delta[C_H^{*P}] C_{H'}^{P'} + C_H^{*P} \delta[C_{H'}^{P'}] \right\} = 0.$$

Since the coefficients $\delta[C_H^{*P}]$ and $\delta[C_{H'}^{P'}]$ are complex conjugates it is necessary and sufficient to require the quantities that multiply with $\delta[C_H^{*P}]$ to vanish.

Full Configuration Interaction Theory

This leads to

$$\sum_{P'H'} \langle \Phi_H^P | \hat{H} | \Phi_{H'}^{P'} \rangle C_{H'}^{P'} - \lambda C_H^P = 0,$$

for all sets of P and H.

If we then multiply by the corresponding C_H^{*P} and sum over PH we obtain

$$\sum_{PP'HH'} C_H^{*P} \langle \Phi_H^P | \hat{H} | \Phi_{H'}^{P'} \rangle C_{H'}^{P'} - \lambda \sum_{PH} |C_H^P|^2 = 0,$$

leading to the identification $\lambda = E$. This means that we have for all PH sets

$$\sum_{P'H'} \langle \Phi_H^P | \hat{H} - E | \Phi_{H'}^{P'} \rangle = 0. \tag{3.1}$$

Full Configuration Interaction Theory

An alternative way to derive the last equation is to start from

$$(\hat{H} - E) | \Psi_0 \rangle = (\hat{H} - E) \sum_{P'H'} C_{H'}^{P'} | \Phi_{H'}^{P'} \rangle = 0,$$

and if this equation is successively projected against all Φ_H^P in the expansion of Ψ, then the last equation on the previous slide results. As stated previously, one solves this equation normally by diagonalization. If we are able to solve this equation exactly (that is numerically exactly) in a large Hilbert space (it will be truncated in terms of the number of single-particle states included in the definition of Slater determinants), it can then serve as a benchmark for other many-body methods which approximate the correlation operator $\hat{C}$.

Example of a Hamiltonian matrix

Suppose, as an example, that we have six fermions below the Fermi level. This means that we can make at most $6p-6h$ excitations. If we have an infinity of single particle states above the Fermi level, we will obviously have an infinity of say $2p-2h$ excitations. Each such way to configure the particles is called a **configuration**. We will always have to truncate in the basis of single-particle states. This gives us a finite number of possible Slater determinants. Our Hamiltonian matrix would then look like (where each block can have a large dimensionalities):

	$0p-0h$	$1p-1h$	$2p-2h$	$3p-3h$	$4p-4h$	$5p-5h$	$6p-6h$
$0p-0h$	x	x	x	0	0	0	0
$1p-1h$	x	x	x	x	0	0	0
$2p-2h$	x	x	x	x	x	0	0
$3p-3h$	0	x	x	x	x	x	0
$4p-4h$	0	0	x	x	x	x	x
$5p-5h$	0	0	0	x	x	x	x
$6p-6h$	0	0	0	0	x	x	x

with a two-body force. Why are there non-zero blocks of elements?

Example of a Hamiltonian matrix with a Hartree-Fock basis

If we use a Hartree-Fock basis, this corresponds to a particular unitary transformation where matrix elements of the type $\langle 0p-0h|\hat{H}|1p-1h\rangle = \langle \Phi_0|\hat{H}|\Phi_i^a\rangle = 0$ and our Hamiltonian matrix becomes

	$0p-0h$	$1p-1h$	$2p-2h$	$3p-3h$	$4p-4h$	$5p-5h$	$6p-6h$
$0p-0h$	$\tilde{x}$	0	$\tilde{x}$	0	0	0	0
$1p-1h$	0	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	0	0	0
$2p-2h$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	0	0
$3p-3h$	0	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	0
$4p-4h$	0	0	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$
$5p-5h$	0	0	0	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$
$6p-6h$	0	0	0	0	$\tilde{x}$	$\tilde{x}$	$\tilde{x}$

Shell-model jargon

If we do not make any truncations in the possible sets of Slater determinants (many-body states) we can make by distributing A nucleons among n single-particle states, we call such a calculation for **Full configuration interaction theory**

If we make truncations, we have different possibilities

- The standard nuclear shell-model. Here we define an effective Hilbert space with respect to a given core. The calculations are normally then performed for all many-body states that can be constructed from the effective Hilbert spaces. This approach requires a properly defined effective Hamiltonian

- We can truncate in the number of excitations. For example, we can limit the possible Slater determinants to only $1p-1h$ and $2p-2h$ excitations. This is called a configuration interaction calculation at the level of singles and doubles excitations, or just CISD.
- We can limit the number of excitations in terms of the excitation energies. If we do not define a core, this defines normally what is called the no-core shell-model approach.

What happens if we have a three-body interaction and a Hartree-Fock basis?

FCI and the exponential growth

Full configuration interaction theory calculations provide in principle, if we can diagonalize numerically, all states of interest. The dimensionality of the problem explodes however quickly.

The total number of Slater determinants which can be built with say N neutrons distributed among n single particle states is

$$\binom{n}{N} = \frac{n!}{(n-N)!N!}.$$

For a model space which comprises the first for major shells only $0s$, $0p$, $1s0d$ and $1p0f$ we have 40 single particle states for neutrons and protons. For the eight neutrons of oxygen-16 we would then have

$$\binom{40}{8} = \frac{40!}{(32)!8!} \sim 10^9,$$

and multiplying this with the number of proton Slater determinants we end up with approximately with a dimensionality d of $d \sim 10^{18}$.

Exponential wall

This number can be reduced if we look at specific symmetries only. However, the dimensionality explodes quickly!

- For Hamiltonian matrices of dimensionalities which are smaller than $d \sim 10^5$, we would use so-called direct methods for diagonalizing the Hamiltonian matrix
- For larger dimensionalities iterative eigenvalue solvers like Lanczos' method are used. The most efficient codes at present can handle matrices of $d \sim 10^{10}$.

A non-practical way of solving the eigenvalue problem

To see this, we look at the contributions arising from

$$\langle \Phi_H^P| = \langle \Phi_0|$$

in Eq. (3.1), that is we multiply with $\langle\Phi_0|$ from the left in

$$(\hat{H}-E)\sum_{P'H'}C_{H'}^{P'}|\Phi_{H'}^{P'}\rangle=0.$$

If we assume that we have a two-body operator at most, Slater's rule gives then an equation for the correlation energy in terms of C_i^a and C_{ij}^{ab} only. We get then

$$\langle\Phi_0|\hat{H}-E|\Phi_0\rangle+\sum_{ai}\langle\Phi_0|\hat{H}-E|\Phi_i^a\rangle C_i^a+\sum_{abij}\langle\Phi_0|\hat{H}-E|\Phi_{ij}^{ab}\rangle C_{ij}^{ab}=0,$$

or

$$E-E_0=\Delta E=\sum_{ai}\langle\Phi_0|\hat{H}|\Phi_i^a\rangle C_i^a+\sum_{abij}\langle\Phi_0|\hat{H}|\Phi_{ij}^{ab}\rangle C_{ij}^{ab},$$

where the energy E_0 is the reference energy and ΔE defines the so-called correlation energy. The single-particle basis functions could be the results of a Hartree-Fock calculation or just the eigenstates of the non-interacting part of the Hamiltonian.

A non-practical way of solving the eigenvalue problem

To see this, we look at the contributions arising from

$$\langle\Phi_H^P|=\langle\Phi_0|$$

in Eq. (3.1), that is we multiply with $\langle\Phi_0|$ from the left in

$$(\hat{H}-E)\sum_{P'H'}C_{H'}^{P'}|\Phi_{H'}^{P'}\rangle=0.$$

A non-practical way of solving the eigenvalue problem

If we assume that we have a two-body operator at most, Slater's rule gives then an equation for the correlation energy in terms of C_i^a and C_{ij}^{ab} only. We get then

$$\langle\Phi_0|\hat{H}-E|\Phi_0\rangle+\sum_{ai}\langle\Phi_0|\hat{H}-E|\Phi_i^a\rangle C_i^a+\sum_{abij}\langle\Phi_0|\hat{H}-E|\Phi_{ij}^{ab}\rangle C_{ij}^{ab}=0,$$

or

$$E-E_0=\Delta E=\sum_{ai}\langle\Phi_0|\hat{H}|\Phi_i^a\rangle C_i^a+\sum_{abij}\langle\Phi_0|\hat{H}|\Phi_{ij}^{ab}\rangle C_{ij}^{ab},$$

where the energy E_0 is the reference energy and ΔE defines the so-called correlation energy. The single-particle basis functions could be the results of a Hartree-Fock calculation or just the eigenstates of the non-interacting part of the Hamiltonian.

Rewriting the FCI equation

In our notes on Hartree-Fock calculations, we have already computed the matrix $\langle\Phi_0|\hat{H}|\Phi_i^a\rangle$ and $\langle\Phi_0|\hat{H}|\Phi_{ij}^{ab}\rangle$. If we are using a Hartree-Fock basis, then the matrix elements $\langle\Phi_0|\hat{H}|\Phi_i^a\rangle = 0$ and we are left with a *correlation energy* given by

$$E - E_0 = \Delta E^{HF} = \sum_{abij}\langle\Phi_0|\hat{H}|\Phi_{ij}^{ab}\rangle C_{ij}^{ab}.$$

Rewriting the FCI equation

Inserting the various matrix elements we can rewrite the previous equation as

$$\Delta E = \sum_{ai}\langle i|\hat{f}|a\rangle C_i^a + \sum_{abij}\langle ij|\hat{v}|ab\rangle C_{ij}^{ab}.$$

This equation determines the correlation energy but not the coefficients C.

Rewriting the FCI equation, does not stop here

We need more equations. Our next step is to set up

$$\langle\Phi_i^a|\hat{H}-E|\Phi_0\rangle + \sum_{bj}\langle\Phi_i^a|\hat{H}-E|\Phi_j^b\rangle C_j^b + \sum_{bcjk}\langle\Phi_i^a|\hat{H}-E|\Phi_{jk}^{bc}\rangle C_{jk}^{bc} + \sum_{bcdjkl}\langle\Phi_i^a|\hat{H}-E|\Phi_{jkl}^{bcd}\rangle C_{jkl}^{bcd} = 0,$$

as this equation will allow us to find an expression for the coefficents C_i^a since we can rewrite this equation as

$$\langle i|\hat{f}|a\rangle + \langle\Phi_i^a|\hat{H}|\Phi_i^a\rangle C_i^a + \sum_{bj\neq ai}\langle\Phi_i^a|\hat{H}|\Phi_j^b\rangle C_j^b + \sum_{bcjk}\langle\Phi_i^a|\hat{H}|\Phi_{jk}^{bc}\rangle C_{jk}^{bc} + \sum_{bcdjkl}\langle\Phi_i^a|\hat{H}|\Phi_{jkl}^{bcd}\rangle C_{jkl}^{bcd} = EC_i^a.$$

Rewriting the FCI equation, please stop here

We see that on the right-hand side we have the energy E. This leads to a non-linear equation in the unknown coefficients. These equations are normally solved iteratively (that is we can start with a guess for the coefficients C_i^a). A common choice is to use perturbation theory for the first guess, setting thereby

$$C_i^a = \frac{\langle i|\hat{f}|a\rangle}{\varepsilon_i - \varepsilon_a}.$$

Rewriting the FCI equation, more to add

The observant reader will however see that we need an equation for C_{jk}^{bc} and C_{jkl}^{bcd} as well. To find equations for these coefficients we need then to continue our multiplications from the left with the various Φ_H^P terms.

For C_{jk}^{bc} we need then

$$\langle \Phi_{ij}^{ab}|\hat{H}-E|\Phi_0\rangle + \sum_{kc}\langle \Phi_{ij}^{ab}|\hat{H}-E|\Phi_k^c\rangle C_k^c +$$

$$\sum_{cdkl}\langle \Phi_{ij}^{ab}|\hat{H}-E|\Phi_{kl}^{cd}\rangle C_{kl}^{cd} + \sum_{cdeklm}\langle \Phi_{ij}^{ab}|\hat{H}-E|\Phi_{klm}^{cde}\rangle C_{klm}^{cde} + \sum_{cdefklmn}\langle \Phi_{ij}^{ab}|\hat{H}-E|\Phi_{klmn}^{cdef}\rangle C_{klmn}^{cdef} = 0,$$

and we can isolate the coefficients C_{kl}^{cd} in a similar way as we did for the coefficients C_i^a.

Rewriting the FCI equation, more to add

A standard choice for the first iteration is to set

$$C_{ij}^{ab} = \frac{\langle ij|\hat{v}|ab\rangle}{\varepsilon_i+\varepsilon_j-\varepsilon_a-\varepsilon_b}.$$

At the end we can rewrite our solution of the Schroedinger equation in terms of n coupled equations for the coefficients C_H^P. This is a very cumbersome way of solving the equation. However, by using this iterative scheme we can illustrate how we can compute the various terms in the wave operator or correlation operator $\hat{C}$. We will later identify the calculation of the various terms C_H^P as parts of different many-body approximations to full CI. In particular, we can relate this non-linear scheme with Coupled Cluster theory and many-body perturbation theory.

Summarizing FCI and bringing in approximative methods

If we can diagonalize large matrices, FCI is the method of choice since:

- It gives all eigenvalues, ground state and excited states
- The eigenvectors are obtained directly from the coefficients C_H^P which result from the diagonalization
- We can compute easily expectation values of other operators, as well as transition probabilities
- Correlations are easy to understand in terms of contributions to a given operator beyond the Hartree-Fock contribution. This is the standard approach in many-body theory.

Definition of the correlation energy

The correlation energy is defined as, with a two-body Hamiltonian,

$$\Delta E = \sum_{ai} \langle i|\hat{f}|a\rangle C_i^a + \sum_{abij} \langle ij|\hat{v}|ab\rangle C_{ij}^{ab}.$$

The coefficients C result from the solution of the eigenvalue problem. The energy of say the ground state is then

$$E = E_{ref} + \Delta E,$$

where the so-called reference energy is the energy we obtain from a Hartree-Fock calculation, that is

$$E_{ref} = \langle \Phi_0|\hat{H}|\Phi_0\rangle.$$

FCI equation and the coefficients

However, as we have seen, even for a small case like the four first major shells and a nucleus like oxygen-16, the dimensionality becomes quickly intractable. If we wish to include single-particle states that reflect weakly bound systems, we need a much larger single-particle basis. We need thus approximative methods that sum specific correlations to infinite order.

Popular methods are

- Many-body perturbation theory (in essence a Taylor expansion)
- Coupled cluster theory (coupled non-linear equations)
- Green's function approaches (matrix inversion)
- Similarity group transformation methods (coupled ordinary differential equations)

All these methods start normally with a Hartree-Fock basis as the calculational basis.

Important ingredients to have in codes

- Be able to validate and verify the algorithms.
- Include concepts like unit testing. Gives the possibility to test and validate several or all parts of the code.
- Validation and verification are then included *naturally* and one can develop a better attitude to what is meant with an ethically sound scientific approach.

A structured approach to solving problems

In the steps that lead to the development of clean code you should think of

1. How to structure a code in terms of functions (use IDEs or advanced text editors like sublime or atom)
2. How to make a module
3. How to read input data flexibly from the command line or files
4. How to create graphical/web user interfaces
5. How to write unit tests
6. How to refactor code in terms of classes (instead of functions only)
7. How to conduct and automate large-scale numerical experiments
8. How to write scientific reports in various formats (LaTeX, HTML, doconce)

Additional benefits

Many of the above aspetcs will save you a lot of time when you incrementally extend software over time from simpler to more complicated problems. In particular, you will benefit from many good habits:

1. New code is added in a modular fashion to a library (modules)
2. Programs are run through convenient user interfaces
3. It takes one quick command to let all your code undergo heavy testing
4. Tedious manual work with running programs is automated,
5. Your scientific investigations are reproducible, scientific reports with top quality typesetting are produced both for paper and electronic devices. Use version control software like git and repositories like github

Unit Testing

Unit Testing is the practice of testing the smallest testable parts, called units, of an application individually and independently to determine if they behave exactly as expected.

Unit tests (short code fragments) are usually written such that they can be preformed at any time during the development to continually verify the behavior of the code.

In this way, possible bugs will be identified early in the development cycle, making the debugging at later stages much easier.

Unit Testing, benefits

There are many benefits associated with Unit Testing, such as

- It increases confidence in changing and maintaining code. Big changes can be made to the code quickly, since the tests will ensure that everything still is working properly.
- Since the code needs to be modular to make Unit Testing possible, the code will be easier to reuse. This improves the code design.
- Debugging is easier, since when a test fails, only the latest changes need to be debugged.
  - Different parts of a project can be tested without the need to wait for the other parts to be available.
- A unit test can serve as a documentation on the functionality of a unit of the code.

Simple example of unit test

Look up the guide on how to install unit tests for c++ at course webpage. This is the version with classes.

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
include <unittest++/UnitTest++.h>
class MyMultiplyClass public: double multiply(double x, double y) return x * y; ;
TEST(MyMath) MyMultiplyClass my; CHECK$_EQUAL(56, my.multiply(7,8))$;
int main() return UnitTest::RunAllTests();

Simple example of unit test

And without classes

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
include <unittest++/UnitTest++.h>
double multiply(double x, double y) return x * y;
TEST(MyMath) CHECK$_EQUAL(56, multiply(7,8))$;
int main() return UnitTest::RunAllTests();

For Fortran users, the link at `http://sourceforge.net/projects/fortranxunit/` contains a similar software for unit testing. For Python go to `https://docs.python.org/2/library/unittest.html`.

Unit tests

There are many types of **unit test** libraries. One which is very popular with C++ programmers is Catch

Catch is header only. All you need to do is drop the file(s) somewhere reachable from your project - either in some central location you can set your header search path to find, or directly into your project tree itself!

This is a particularly good option for other Open-Source projects that want to use Catch for their test suite.

Examples

Computing factorials

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
inline unsigned int Factorial(unsigned int number) return number > 1 ? Factorial(number-1)*number : 1;

Factorial Example

Simple test where we put everything in a single file

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
define CATCH$_{CONFIG_MAIN}$//ThistellsCatchtoprovideamain()include"catch.hpp"inlineunsignedintFactorial(unsignedintnumber)returnn

TEST$_{CASE}$("Factorialsarecomputed","[factorial]")REQUIRE(Factorial(0) == 1);REQUIRE(Factorial(1) == 1);REQUIRE(Factoria

This will compile to a complete executable which responds to command line arguments. If you just run it with no arguments it will execute all test cases (in this case there is just one), report any failures, report a summary of how many tests passed and failed and return the number of failed tests.

What did we do (1)?

All we did was

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
define

one identifier and

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
include

one header and we got everything - even an implementation of main() that will respond to command line arguments. Once you have more than one file with unit tests in you'll just need to

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
include "catch.hpp"

and go. Usually it's a good idea to have a dedicated implementation file that just has

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
define CATCH$_{C}$$_{O}$$_{N}$$_{F}$$_{I}$$_{G}$$_{M}$$_{A}$$_{I}$$_{N}$$_{i}$$_{n}$$_{c}$$_{l}$$_{u}$$_{d}$$_{e}$”$_{c}$$_{a}$$_{t}$$_{c}$$_{h}$.$_{h}$$_{p}$$_{p}$”.

You can also provide your own implementation of main and drive Catch yourself.

What did we do (2)?

We introduce test cases with the

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
TEST$_{C}$$_{A}$$_{S}$$_{E}$

macro.

The test name must be unique. You can run sets of tests by specifying a wildcarded test name or a tag expression. All we did was **define** one identifier and **include** one header and we got everything.

We write our individual test assertions using the

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]c++
REQUIRE

macro.

Unit test summary and testing approach

Three levels of tests

1. Microscopic level: testing small parts of code, use often unit test libraries
2. Mesoscopic level: testing the integration of various parts of your code
3. Macroscopic level: testing that the final result is ok

Coding Recommendations

Writing clean and clear code is an art and reflects your understanding of

1. derivation, verification, and implementation of algorithms
2. what can go wrong with algorithms
3. overview of important, known algorithms
4. how algorithms are used to solve mathematical problems
5. reproducible science and ethics
6. algorithmic thinking for gaining deeper insights about scientific problems

Computing is understanding and your understanding is reflected in your abilities to write clear and clean code.

Summary and recommendations

Some simple hints and tips in order to write clean and clear code

1. Spell out the algorithm and have a top-down approach to the flow of data
2. Start with coding as close as possible to eventual mathematical expressions
3. Use meaningful names for variables
4. Split tasks in simple functions and modules/classes
5. Functions should return as few as possible variables
6. Use unit tests and make sure your codes are producing the correct results
7. Where possible use symbolic coding to autogenerate code and check results
8. Make a proper timing of your algorithms
9. Use version control and make your science reproducible
10. Use IDEs or smart editors with debugging and analysis tools.
11. Automatize your computations interfacing high-level and compiled languages like C++ and Fortran.
12. .....

Building a many-body basis

Here we will discuss how we can set up a single-particle basis which we can use in the various parts of our projects, from the simple pairing model to infinite nuclear matter. We will use here the simple pairing model to illustrate in particular how to set up a single-particle basis. We will also use this do discuss standard FCI approaches like:

1. Standard shell-model basis in one or two major shells
2. Full CI in a given basis and no truncations
3. CISD and CISDT approximations
4. No-core shell model and truncation in excitation energy

Building a many-body basis

An important step in an FCI code is to construct the many-body basis.

While the formalism is independent of the choice of basis, the **effectiveness** of a calculation will certainly be basis dependent.

Furthermore there are common conventions useful to know.

First, the single-particle basis has angular momentum as a good quantum number. You can imagine the single-particle wavefunctions being generated by a one-body Hamiltonian, for example a harmonic oscillator. Modifications include harmonic oscillator plus spin-orbit splitting, or self-consistent mean-field potentials, or the Woods-Saxon potential which mocks up the self-consistent mean-field. For nuclei, the harmonic oscillator, modified by spin-orbit splitting, provides a useful language for describing single-particle states.

Building a many-body basis

Each single-particle state is labeled by the following quantum numbers:

- Orbital angular momentum l
- Intrinsic spin s = 1/2 for protons and neutrons
- Angular momentum $j = l \pm 1/2$
- z-component j_z (or m)
- Some labeling of the radial wavefunction, typically n the number of nodes in the radial wavefunction, but in the case of harmonic oscillator one can also use the principal quantum number N, where the harmonic oscillator energy is $(N+3/2)\omega$.

In this format one labels states by $n(l)_j$, with (l) replaced by a letter: s for $l = 0$, p for $l = 1$, d for $l = 2$, f for $l = 3$, and thenceforth alphabetical.

Building a many-body basis

In practice the single-particle space has to be severely truncated. This truncation is typically based upon the single-particle energies, which is the effective energy from a mean-field potential.

Sometimes we freeze the core and only consider a valence space. For example, one may assume a frozen ^{4}He core, with two protons and two neutrons in the $0s_{1/2}$ shell, and then only allow active particles in the $0p_{1/2}$ and $0p_{3/2}$ orbits.

Another example is a frozen ^{16}O core, with eight protons and eight neutrons filling the $0s_{1/2}$, $0p_{1/2}$ and $0p_{3/2}$ orbits, with valence particles in the $0d_{5/2}, 1s_{1/2}$ and $0d_{3/2}$ orbits.

Sometimes we refer to nuclei by the valence space where their last nucleons go. So, for example, we call ^{12}C a p-shell nucleus, while ^{26}Al is an sd-shell nucleus and ^{56}Fe is a pf-shell nucleus.

Building a many-body basis

There are different kinds of truncations.

- For example, one can start with ‘filled’ orbits (almost always the lowest), and then allow one, two, three... particles excited out of those filled orbits. These are called 1p-1h, 2p-2h, 3p-3h excitations.
- Alternately, one can state a maximal orbit and allow all possible configurations with particles occupying states up to that maximum. This is called *full configuration*.
- Finally, for particular use in nuclear physics, there is the *energy* truncation, also called the $N\Omega$ or N_{max} truncation.

Building a many-body basis

Here one works in a harmonic oscillator basis, with each major oscillator shell assigned a principal quantum number $N = 0, 1, 2, 3, \ldots$. The $N\Omega$ or N_{max} truncation: Any configuration is given an noninteracting energy, which is the sum of the single-particle harmonic oscillator energies. (Thus this ignores spin-orbit splitting.)

Excited state are labeled relative to the lowest configuration by the number of harmonic oscillator quanta.

This truncation is useful because if one includes *all* configuration up to some N_{max}, and has a translationally invariant interaction, then the intrinsic motion and the center-of-mass motion factor. In other words, we can know exactly the center-of-mass wavefunction.

In almost all cases, the many-body Hamiltonian is rotationally invariant. This means it commutes with the operators $\hat{J}^2, \hat{J}_z$ and so eigenstates will have good J, M. Furthermore, the eigenenergies do not depend upon the orientation M.

Therefore we can choose to construct a many-body basis which has fixed M; this is called an M-scheme basis.

Alternately, one can construct a many-body basis which has fixed J, or a J-scheme basis.

Building a many-body basis

The Hamiltonian matrix will have smaller dimensions (a factor of 10 or more) in the J-scheme than in the M-scheme. On the other hand, as we'll show in the next slide, the M-scheme is very easy to construct with Slater determinants, while the J-scheme basis states, and thus the matrix elements, are more complicated, almost always being linear combinations of M-scheme states. J-scheme bases are important and useful, but we'll focus on the simpler M-scheme.

The quantum number m is additive (because the underlying group is Abelian): if a Slater determinant $\hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_k^\dagger \ldots |0\rangle$ is built from single-particle states all with good m, then the total

$$M = m_i + m_j + m_k + \ldots$$

This is *not* true of J, because the angular momentum group SU(2) is not Abelian.

Building a many-body basis

The upshot is that

- It is easy to construct a Slater determinant with good total M;
- It is trivial to calculate M for each Slater determinant;
- So it is easy to construct an M-scheme basis with fixed total M.

Note that the individual M-scheme basis states will *not*, in general, have good total J. Because the Hamiltonian is rotationally invariant, however, the eigenstates will have good J. (The situation is muddied when one has states of different J that are nonetheless degenerate.)

Building a many-body basis

Example: two $j = 1/2$ orbits

Index	n	l	j	m_j
1	0	0	1/2	-1/2
2	0	0	1/2	1/2
3	1	0	1/2	-1/2
4	1	0	1/2	1/2

Note that the order is arbitrary.

Building a many-body basis

There are $\binom{4}{2} = 6$ two-particle states, which we list with the total M:

Occupied	M
1,2	0
1,3	-1
1,4	0
2,3	0
2,4	1
3,4	0

There are 4 states with $M = 0$, and 1 each with $M = \pm 1$.

Building a many-body basis

As another example, consider using only single particle states from the $0d_{5/2}$ space. They have the following quantum numbers

Index	n	l	j	m_j
1	0	2	5/2	-5/2
2	0	2	5/2	-3/2
3	0	2	5/2	-1/2
4	0	2	5/2	1/2
5	0	2	5/2	3/2
6	0	2	5/2	5/2

Building a many-body basis

There are $\binom{6}{2} = 15$ two-particle states, which we list with the total M:

Occupied	M	Occupied	M	Occupied	M
1,2	-4	2,3	-2	3,5	1
1,3	-3	2,4	-1	3,6	2
1,4	-2	2,5	0	4,5	2
1,5	-1	2,6	1	4,6	3
1,6	0	3,4	0	5,6	4

There are 3 states with $M = 0$, 2 with $M = 1$, and so on.

Shell-model project

The first step is to construct the M-scheme basis of Slater determinants. Here M-scheme means the total J_z of the many-body states is fixed.

The steps could be:

- Read in a user-supplied file of single-particle states (examples can be given) or just code these internally;
- Ask for the total M of the system and the number of particles N;
- Construct all the N-particle states with given M. You will validate the code by comparing both the number of states and specific states.

Shell-model project

The format of a possible input file could be

Index	n	l	$2j$	$2m_j$
1	1	0	1	-1
2	1	0	1	1
3	0	2	3	-3
4	0	2	3	-1
5	0	2	3	1
6	0	2	3	3
7	0	2	5	-5
8	0	2	5	-3
9	0	2	5	-1
10	0	2	5	1
11	0	2	5	3
12	0	2	5	5

This represents the $1s_{1/2}0d_{3/2}0d_{5/2}$ valence space, or just the sd-space. There are twelve single-particle states, labeled by an overall index, and which have associated quantum numbers the

number of radial nodes, the orbital angular momentum l, and the angular momentum j and third component j_z. To keep everything as integers, we could store $2\times j$ and $2\times j_z$.

Shell-model project

To read in the single-particle states you need to:

- Open the file
  - Read the number of single-particle states (in the above example, 12); allocate memory; all you need is a single array storing $2\times j_z$ for each state, labeled by the index.
- Read in the quantum numbers and store $2\times j_z$ (and anything else you happen to want).

Shell-model project

The next step is to read in the number of particles N and the fixed total M (or, actually, $2\times M$). For this project we assume only a single species of particles, say neutrons, although this can be relaxed. **Note**: Although it is often a good idea to try to write a more general code, given the short time alloted we would suggest you keep your ambition in check, at least in the initial phases of the project.

You should probably write an error trap to make sure N and M are congruent; if N is even, then $2\times M$ should be even, and if N is odd then $2\times M$ should be odd.

Shell-model project

The final step is to generate the set of N-particle Slater determinants with fixed M. The Slater determinants will be stored in occupation representation. Although in many codes this representation is done compactly in bit notation with ones and zeros, but for greater transparency and simplicity we will list the occupied single particle states.

Hence we can store the Slater determinant basis states as $sd(i,j)$, that is an array of dimension N_{SD}, the number of Slater determinants, by N, the number of occupied state. So if for the 7th Slater determinant the 2nd, 3rd, and 9th single-particle states are occupied, then $sd(7,1)=2$, $sd(7,2)=3$, and $sd(7,3)=9$.

Shell-model project

We can construct an occupation representation of Slater determinants by the *odometer* method. Consider $N_{sp} = 12$ and $N = 4$. Start with the first 4 states occupied, that is:

- $sd(1,:) = 1,2,3,4$ (also written as $|1,2,3,4\rangle$)

Now increase the last occupancy recursively:

- $sd(2,:) = 1,2,3,5$
- $sd(3,:) = 1,2,3,6$
- $sd(4,:) = 1,2,3,7$
- ...
- $sd(9,:) = 1,2,3,12$

Then start over with

- $sd(10,:) = 1,2,4,5$

and again increase the rightmost digit

- $sd(11,:) = 1,2,4,6$
- $sd(12,:) = 1,2,4,7$
- ...
- $sd(17,:) = 1,2,4,12$

Shell-model project

When we restrict ourselves to an M-scheme basis, we could choose two paths. The first is simplest (and simplest is often best, at least in the first draft of a code): generate all possible Slater determinants, and then extract from this initial list a list of those Slater determinants with a given M. (You will need to write a short function or routine that computes M for any given occupation.)

Alternately, and not too difficult, is to run the odometer routine twice: each time, as as a Slater determinant is calculated, compute M, but do not store the Slater determinants except the current one. You can then count up the number of Slater determinants with a chosen M. Then allocated storage for the Slater determinants, and run the odometer algorithm again, this time storing Slater determinants with the desired M (this can be done with a simple logical flag).

Shell-model project

Some example solutions: Let's begin with a simple case, the $0d_{5/2}$ space containing six single-particle states

Index	n	l	j	m_j
1	0	2	5/2	-5/2
2	0	2	5/2	-3/2
3	0	2	5/2	-1/2
4	0	2	5/2	1/2
5	0	2	5/2	3/2
6	0	2	5/2	5/2

For two particles, there are a total of 15 states, which we list here with the total M:

- $|1,2\rangle$, $M=-4$, $|1,3\rangle$, $M=-3$
- $|1,4\rangle$, $M=-2$, $|1,5\rangle$, $M=-1$
- $|1,5\rangle$, $M=0$, $vert2,3\rangle$, $M=-2$
- $|2,4\rangle$, $M=-1$, $|2,5\rangle$, $M=0$
- $|2,6\rangle$, $M=1$, $|3,4\rangle$, $M=0$
- $|3,5\rangle$, $M=1$, $|3,6\rangle$, $M=2$
- $|4,5\rangle$, $M=2$, $|4,6\rangle$, $M=3$
- $|5,6\rangle$, $M=4$

Of these, there are only 3 states with $M=0$.

Shell-model project

You should try by hand to show that in this same single-particle space, that for $N=3$ there are 3 states with $M=1/2$ and for $N=4$ there are also only 3 states with $M=0$.

To test your code, confirm the above.

Also, for the sd-space given above, for $N=2$ there are 14 states with $M=0$, for $N=3$ there are 37 states with $M=1/2$, for $N=4$ there are 81 states with $M=0$.

Shell-model project

For our project, we will only consider the pairing model. A simple space is the $(1/2)^2$ space with four single-particle states

Index	n	l	s	m_s
1	0	0	1/2	-1/2
2	0	0	1/2	1/2
3	1	0	1/2	-1/2
4	1	0	1/2	1/2

For $N=2$ there are 4 states with $M=0$; show this by hand and confirm your code reproduces it.

Shell-model project

Another, slightly more challenging space is the $(1/2)^4$ space, that is, with eight single-particle states we have

Index	n	l	s	m_s
1	0	0	1/2	-1/2
2	0	0	1/2	1/2
3	1	0	1/2	-1/2
4	1	0	1/2	1/2
5	2	0	1/2	-1/2
6	2	0	1/2	1/2
7	3	0	1/2	-1/2
8	3	0	1/2	1/2

For $N = 2$ there are 16 states with $M = 0$; for $N = 3$ there are 24 states with $M = 1/2$, and for $N = 4$ there are 36 states with $M = 0$.

Shell-model project

In the shell-model context we can interpret this as 4 $s_{1/2}$ levels, with $m = \pm 1/2$, we can also think of these are simple four pairs, $\pm k, k = 1,2,3,4$. Later on we will assign single-particle energies, depending on the radial quantum number n, that is, $\varepsilon_k = |k|\delta$ so that they are equally spaced.

Shell-model project

For application in the pairing model we can go further and consider only states with no "broken pairs," that is, if $+k$ is filled (or $m = +1/2$, so is $-k$ ($m = -1/2$). If you want, you can write your code to accept only these, and obtain the following six states:

- $|1,2,3,4\rangle$,
- $|1,2,5,6\rangle$,
- $|1,2,7,8\rangle$,
- $|3,4,5,6\rangle$,
- $|3,4,7,8\rangle$,
- $|5,6,7,8\rangle$

Shell-model project

Hints for coding.

- Write small modules (routines/functions) ; avoid big functions that do everything. (But not too small.)
- Use Unit tests! Write lots of error traps, even for things that are 'obvious.'
- Document as you go along. The Unit tests serve as documentation. For each function write a header that includes:

  1. Main purpose of function and/or unit test
  2. names and brief explanation of input variables, if any
  3. names and brief explanation of output variables, if any
  4. functions called by this function
  5. called by which functions

Shell-model project

Hints for coding

- Unit tests will save time. Use also IDEs for debugging. If you insist on brute force debugging, print out intermediate values. It's almost impossible to debug a code by looking at it–the code will almost always win a 'staring contest.'
- Validate code with SIMPLE CASES. Validate early and often. Unit tests!!

The number one mistake is using a too complex a system to test. For example , if you are computing particles in a potential in a box, try removing the potential–you should get particles in a box. And start with one particle, then two, then three... Don't start with eight particles.

Shell-model project

Our recommended occupation representation, e.g. $|1,2,4,8\rangle$, is easy to code, but numerically inefficient when one has hundreds of millions of Slater determinants.

In state-of-the-art shell-model codes, one generally uses bit representation, i.e. $|1101000100...\rangle$ where one stores the Slater determinant as a single (or a small number of) integer.

This is much more compact, but more intricate to code with considerable more overhead. There exist bit-manipulation functions. We will discuss these in more detail at the beginning of the third week.

Example case: pairing Hamiltonian

We consider a space with 2Ω single-particle states, with each state labeled by $k = 1,2,3,\Omega$ and $m = \pm 1/2$. The convention is that the state with $k > 0$ has $m = +1/2$ while $-k$ has $m = -1/2$.

The Hamiltonian we consider is

$$\hat{H} = -G\hat{P}_+\hat{P}_-,$$

where

$$\hat{P}_+ = \sum_{k>0} \hat{a}_k^\dagger \hat{a}_{-k}^\dagger.$$

and $\hat{P}_- = (\hat{P}_+)^\dagger$.

This problem can be solved using what is called the quasi-spin formalism to obtain the exact results. Thereafter we will try again using the explicit Slater determinant formalism.

Example case: pairing Hamiltonian

One can show (and this is part of the project) that

$$\left[\hat{P}_+, \hat{P}_-\right] = \sum_{k>0} \left(\hat{a}_k^\dagger \hat{a}_k + \hat{a}_{-k}^\dagger \hat{a}_{-k} - 1\right) = \hat{N} - \Omega.$$

Now define

$$\hat{P}_z = \frac{1}{2}(\hat{N} - \Omega).$$

Finally you can show

$$\left[\hat{P}_z, \hat{P}_\pm\right] = \pm \hat{P}_\pm.$$

This means the operators $\hat{P}_\pm, \hat{P}_z$ form a so-called $SU(2)$ algebra, and we can use all our insights about angular momentum, even though there is no actual angular momentum involved.

So we rewrite the Hamiltonian to make this explicit:

$$\hat{H} = -G\hat{P}_+\hat{P}_- = -G\left(\hat{P}^2 - \hat{P}_z^2 + \hat{P}_z\right)$$

Example case: pairing Hamiltonian

Because of the SU(2) algebra, we know that the eigenvalues of $\hat{P}^2$ must be of the form $p(p+1)$, with p either integer or half-integer, and the eigenvalues of $\hat{P}_z$ are m_p with $p \geq |m_p|$, with m_p also integer or half-integer.

But because $\hat{P}_z = (1/2)(\hat{N} - \Omega)$, we know that for N particles the value $m_p = (N - \Omega)/2$. Furthermore, the values of m_p range from $-\Omega/2$ (for $N = 0$) to $+\Omega/2$ (for $N = 2\Omega$, with all states filled).

We deduce the maximal $p = \Omega/2$ and for a given n the values range of p range from $|N - \Omega|/2$ to $\Omega/2$ in steps of 1 (for an even number of particles)

Following Racah we introduce the notation $p = (\Omega - v)/2$ where $v = 0, 2, 4, ..., \Omega - |N - \Omega|$ With this it is easy to deduce that the eigenvalues of the pairing Hamiltonian are

$$-G(N-v)(2\Omega+2-N-v)/4$$

This also works for N odd, with $v = 1, 3, 5, \ldots$.

Example case: pairing Hamiltonian

Let's take a specific example: $\Omega = 3$ so there are 6 single-particle states, and $N = 3$, with $v = 1, 3$. Therefore there are two distinct eigenvalues,

$$E = -2G, 0$$

Now let's work this out explicitly. The single particle degrees of freedom are defined as

Index	k	m
1	1	-1/2
2	-1	1/2
3	2	-1/2
4	-2	1/2
5	3	-1/2
6	-3	1/2

There are $\binom{6}{3} = 20$ three-particle states, but there are 9 states with $M = +1/2$, namely $|1,2,3\rangle, |1,2,5\rangle, |1,4,6\rangle, |2,3,4\rangle, |2,3,6\rangle, |2,4,5\rangle, |2,5,6\rangle, |3,4,6\rangle, |4,5,6\rangle$.

Example case: pairing Hamiltonian

In this basis, the operator

$$\hat{P}_+ = \hat{a}_1^\dagger \hat{a}_2^\dagger + \hat{a}_3^\dagger \hat{a}_4^\dagger + \hat{a}_5^\dagger \hat{a}_6^\dagger$$

From this we can determine that

$$\hat{P}_-|1,4,6\rangle = \hat{P}_-|2,3,6\rangle = \hat{P}_-|2,4,5\rangle = 0$$

so those states all have eigenvalue 0.

Example case: pairing Hamiltonian

Now for further example,

$$\hat{P}_-|1,2,3\rangle = |3\rangle$$

so

$$\hat{P}_+\hat{P}_-|1,2,3\rangle = |1,2,3\rangle + |3,4,3\rangle + |5,6,3\rangle$$

The second term vanishes because state 3 is occupied twice, and reordering the last term we get

$$\hat{P}_+\hat{P}_-|1,2,3\rangle = |1,2,3\rangle + |3,5,6\rangle$$

without picking up a phase.

Example case: pairing Hamiltonian

Continuing in this fashion, with the previous ordering of the many-body states ($|1,2,3\rangle, |1,2,5\rangle, |1,4,6\rangle, |2,3,4\rangle, |2,3,6\rangle,$ the Hamiltonian matrix of this system is

$$H = -G \begin{pmatrix} 1&0&0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0&0&1 \end{pmatrix}$$

This is useful for our project. One can by hand confirm that there are 3 eigenvalues $-2G$ and 6 with value zero.

Example case: pairing Hamiltonian

Another example Using the $(1/2)^4$ single-particle space, resulting in eight single-particle states

Index	n	l	s	m_s
1	0	0	1/2	-1/2
2	0	0	1/2	1/2
3	1	0	1/2	-1/2
4	1	0	1/2	1/2
5	2	0	1/2	-1/2
6	2	0	1/2	1/2
7	3	0	1/2	-1/2
8	3	0	1/2	1/2

and then taking only 4-particle, $M = 0$ states that have no 'broken pairs', there are six basis Slater determinants:

- $|1,2,3,4\rangle$,
- $|1,2,5,6\rangle$,

- $|1,2,7,8\rangle$,
- $|3,4,5,6\rangle$,
- $|3,4,7,8\rangle$,
- $|5,6,7,8\rangle$

Example case: pairing Hamiltonian

Now we take the following Hamiltonian

$$\hat{H} = \sum_n n\delta \hat{N}_n - G\hat{P}^\dagger \hat{P}$$

where

$$\hat{N}_n = \hat{a}^\dagger_{n,m=+1/2}\hat{a}_{n,m=+1/2} + \hat{a}^\dagger_{n,m=-1/2}\hat{a}_{n,m=-1/2}$$

and

$$\hat{P}^\dagger = \sum_n \hat{a}^\dagger_{n,m=+1/2}\hat{a}^\dagger_{n,m=-1/2}$$

We can write down the 6×6 Hamiltonian in the basis from the prior slide:

$$H = \begin{pmatrix} 2\delta-2G & -G & -G & -G & -G & 0 \\ -G & 4\delta-2G & -G & -G & -0 & -G \\ -G & -G & 6\delta-2G & 0 & -G & -G \\ -G & -G & 0 & 6\delta-2G & -G & -G \\ -G & 0 & -G & -G & 8\delta-2G & -G \\ 0 & -G & -G & -G & -G & 10\delta-2G \end{pmatrix}$$

(You should check by hand that this is correct.)

For $\delta = 0$ we have the closed form solution of the g.s. energy given by $-6G$.

Building a Hamiltonian matrix

The goal is to compute the matrix elements of the Hamiltonian, specifically matrix elements between many-body states (Slater determinants) of two-body operators

$$\sum_{p<q,r<s} V_{pqr}\hat{a}^\dagger_p\hat{a}^\dagger_q\hat{a}_s\hat{a}_r$$

In particular we will need to compute

$$\langle\beta|\hat{a}^\dagger_p\hat{a}^\dagger_q\hat{a}_s\hat{a}_r|\alpha\rangle$$

where α,β are indices labeling Slater determinants and p,q,r,s label single-particle states.

Building a Hamiltonian matrix

Note: there are other, more efficient ways to do this than the method we describe, but you will be able to produce a working code quickly.

As we coded in the first step, a Slater determinant $|\alpha\rangle$ with index α is a list of N occupied single-particle states $i_1 < i_2 < i_3 \dots i_N$.

Furthermore, for the two-body matrix elements V_{pqrs} we normally assume $p < q$ and $r < s$. For our specific project, the interaction is much simpler and you can use this to simplify considerably the setup of a shell-model code for project 2.

What follows here is a more general, but still brute force, approach.

Building a Hamiltonian matrix

Write a function that:

1. Has as input the single-particle indices p,q,r,s for the two-body operator and the index α for the ket Slater determinant;
2. Returns the index β of the unique (if any) Slater determinant such that

$$|\beta\rangle = \pm \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r |\alpha\rangle$$

as well as the phase

This is equivalent to computing

$$\langle\beta| \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r |\alpha\rangle$$

Building a Hamiltonian matrix, first step

The first step can take as input an initial Slater determinant (whose position in the list of basis Slater determinants is α) written as an ordered listed of occupied single-particle states, e.g. $1,2,5,8$, and the indices p,q,r,s from the two-body operator.

It will return another final Slater determinant if the single-particle states r and s are occupied, else it will return an empty Slater determinant (all zeroes).

If r and s are in the list of occupied single particle states, then replace the initial single-particle states ij as $i \to r$ and $j \to r$.

Building a Hamiltonian matrix, second step

The second step will take the final Slater determinant from the first step (if not empty), and then order by pairwise permutations (i.e., if the Slater determinant is $i_1, i_2, i_3, \ldots$, then if $i_n > i_{n+1}$, interchange $i_n \leftrightarrow i_{n+1}$.

Building a Hamiltonian matrix

It will also output a phase. If any two single-particle occupancies are repeated, the phase is 0. Otherwise it is +1 for an even permutation and -1 for an odd permutation to bring the final Slater determinant into ascending order, $j_1 < j_2 < j_3 \ldots$.

Building a Hamiltonian matrix

Example: Suppose in the *sd* single-particle space that the initial Slater determinant is $1,3,9,12$. If $p,q,r,s = 2,8,1,12$, then after the first step the final Slater determinant is $2,3,9,8$. The second step will return $2,3,8,9$ and a phase of -1, because an odd number of interchanges is required.

Building a Hamiltonian matrix

Example: Suppose in the *sd* single-particle space that the initial Slater determinant is $1,3,9,12$. If $p,q,r,s = 3,8,1,12$, then after the first step the final Slater determinant is $3,3,9,8$, but after the second step the phase is 0 because the single-particle state 3 is occupied twice.

Lastly, the final step takes the ordered final Slater determinant and we search through the basis list to determine its index in the many-body basis, that is, β.

Building a Hamiltonian matrix

The Hamiltonian is then stored as an $N_{SD} \times N_{SD}$ array of real numbers, which can be allocated once you have created the many-body basis and know N_{SD}.

Building a Hamiltonian matrix

1. Initialize $H(\alpha,\beta) = 0.0$
2. Set up an outer loop over β
3. Loop over $\alpha = 1, NSD$
4. For each α, loop over $a = 1, ntbme$ and fetch $V(a)$ and the single-particle indices p,q,r,s
5. If $V(a) = 0$ skip. Otherwise, apply $\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r$ to the Slater determinant labeled by α.
6. Find, if any, the label β of the resulting Slater determinant and the phase (which is 0, +1, -1).
7. If phase $\neq 0$, then update $H(\alpha,\beta)$ as $H(\alpha,\beta) + phase * V(a)$. The sum is important because multiple operators might contribute to the same matrix element.
8. Continue loop over a
9. Continue loop over α.
10. End the outer loop over β.

You should force the resulting matrix H to be symmetric. To do this, when updating $H(\alpha,\beta)$, if $\alpha \neq \beta$, also update $H(\beta,\alpha)$.

Building a Hamiltonian matrix

You will also need to include the single-particle energies. This is easy: they only contribute to diagonal matrix elements, that is, $H(\alpha,\alpha)$. Simply find the occupied single-particle states i and add the corresponding $\varepsilon(i)$.

Hamiltonian matrix without the bit representation

Consider the many-body state Ψ_λ expressed as linear combinations of Slater determinants (SD) of orthonormal single-particle states $\phi(\mathbf{r})$:

$$\Psi_\lambda = \sum_i C_{\lambda i} SD_i \tag{3.2}$$

Using the Slater-Condon rules the matrix elements of any one-body ($\mathscr{O}_\infty$) or two-body ($\mathscr{O}_\in$) operator expressed in the determinant space have simple expressions involving one- and two-fermion integrals in our given single-particle basis. The diagonal elements are given by:

$$\begin{aligned} \langle SD|\mathscr{O}_\infty|\mathscr{S}\mathscr{D}\rangle &= \sum_{i\in SD} \langle \phi_i|\mathscr{O}_\infty|\phi_\rangle \rangle \\ \langle SD|\mathscr{O}_\in|\mathscr{S}\mathscr{D}\rangle &= \frac{1}{2} \sum_{(i,j)\in SD} \langle \phi_i\phi_j|\mathscr{O}_\in|\phi_\rangle\phi_|\rangle - \\ &\quad \langle \phi_i\phi_j|\mathscr{O}_\in|\phi_|\phi_\rangle\rangle \end{aligned} \tag{3.3}$$

Hamiltonian matrix without the bit representation, one and two-body operators

For two determinants which differ only by the substitution of single-particle states i with a single-particle state j:

$$\langle SD|\mathscr{O}_\infty|\mathscr{S}\mathscr{D}_\rangle^|\rangle = \langle\phi_i|\mathscr{O}_\infty|\phi_|\rangle \tag{3.4}$$
$$\langle SD|\mathscr{O}_\in|\mathscr{S}\mathscr{D}_\rangle^|\rangle = \sum_{k\in SD}\langle\phi_i\phi_k|\mathscr{O}_\in|\phi_|\phi_\|\rangle - \langle\phi_\rangle\phi_\||\mathscr{O}_\in|\phi_\|\phi_|\rangle$$

For two determinants which differ by two single-particle states

$$\langle SD|\mathscr{O}_\infty|\mathscr{S}\mathscr{D}_{\rangle\|}^{|\updownarrow}\rangle = 0 \tag{3.5}$$
$$\langle SD|\mathscr{O}_\in|\mathscr{S}\mathscr{D}_{\rangle\|}^{|\updownarrow}\rangle = \langle\phi_i\phi_k|\mathscr{O}_\in|\phi_|\phi_\updownarrow\rangle - \langle\phi_\rangle\phi_\||\mathscr{O}_\in|\phi_\updownarrow\phi_|\rangle$$

All other matrix elements involving determinants with more than two substitutions are zero.

Strategies for setting up an algorithm

An efficient implementation of these rules requires

- to find the number of single-particle state substitutions between two determinants
- to find which single-particle states are involved in the substitution
- to compute the phase factor if a reordering of the single-particle states has occured

We can solve this problem using our odometric approach or alternatively using a bit representation as discussed below and in more detail in

- Scemama and Gimer's article (Fortran codes)
- Simen Kvaal's article on how to build an FCI code (C++ code)

We recommend in particular the article by Simen Kvaal. It contains nice general classes for creation and annihilation operators as well as the calculation of the phase (see below).

Computing expectation values and transitions in the shell-model

When we diagonalize the Hamiltonian matrix, the eigenvectors are the coefficients $C_{\lambda i}$ used to express the many-body state Ψ_λ in terms of a linear combinations of Slater determinants (SD) of orthonormal single-particle states $\phi(\mathbf{r})$.

With these eigenvectors we can compute say the transition likelyhood of a one-body operator as

$$\langle\Psi_\lambda|\mathscr{O}_\infty|\ominus_\sigma\rangle = \sum_{\rangle|}\mathscr{C}^*_{\lambda\rangle}\mathscr{C}_{\sigma|}\langle\mathscr{S}\mathscr{D}_\rangle|\mathscr{O}_\infty|\mathscr{S}\mathscr{D}_|\rangle.$$

Writing the one-body operator in second quantization as

$$\mathscr{O}_\infty = \sum_{\blacksquare} \langle \blacksquare | \wr_\infty | \blacksquare \rangle \dashv^\dagger \dashv_\blacksquare,$$

we have

$$\langle \Psi_\lambda | \mathscr{O}_\infty | \ominus_\sigma \rangle = \sum_{\blacksquare} \langle \blacksquare | \wr_\infty | \blacksquare \rangle \sum_{|} \mathscr{C}^*_\lambda \mathscr{C}_\sigma \langle \mathscr{S} \mathscr{D} | \dashv^\dagger \dashv_\blacksquare | \mathscr{S} \mathscr{D} \rangle.$$

Computing expectation values and transitions in the shell-model and spectroscopic factors

The terms we need to evalute then are just the elements

$$\langle SD_i | a_p^\dagger a_q | SD_j \rangle,$$

which can be rewritten in terms of spectroscopic factors by inserting a complete set of Slater determinats as

$$\langle SD_i | a_p^\dagger a_q | SD_j \rangle = \sum_l \langle SD_i | a_p^\dagger | SD_l \rangle \langle SD_l | a_q | SD_j \rangle,$$

where $\langle SD_l | a_q(a_p^\dagger) | SD_j \rangle$ are the spectroscopic factors. These can be easily evaluated in *m*-scheme. Using the Wigner-Eckart theorem we can transform these to a *J*-coupled scheme through so-called reduced matrix elements.

Operators in second quantization

In the build-up of a shell-model or FCI code that is meant to tackle large dimensionalities we need to deal with the action of the Hamiltonian $\hat{H}$ on a Slater determinant represented in second quantization as

$$|\alpha_1 \dots \alpha_n\rangle = a^\dagger_{\alpha_1} a^\dagger_{\alpha_2} \dots a^\dagger_{\alpha_n} |0\rangle.$$

The time consuming part stems from the action of the Hamiltonian on the above determinant,

$$\left(\sum_{\alpha\beta} \langle \alpha | t + u | \beta \rangle a^\dagger_\alpha a_\beta + \frac{1}{4} \sum_{\alpha\beta\gamma\delta} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle a^\dagger_\alpha a^\dagger_\beta a_\delta a_\gamma \right) a^\dagger_{\alpha_1} a^\dagger_{\alpha_2} \dots a^\dagger_{\alpha_n} |0\rangle.$$

A practically useful way to implement this action is to encode a Slater determinant as a bit pattern.

Operators in second quantization

Assume that we have at our disposal n different single-particle states $\alpha_0, \alpha_2, \ldots, \alpha_{n-1}$ and that we can distribute among these states $N \leq n$ particles.

A Slater determinant can then be coded as an integer of n bits. As an example, if we have $n = 16$ single-particle states $\alpha_0, \alpha_1, \ldots, \alpha_{15}$ and $N = 4$ fermions occupying the states α_3, α_6, α_{10} and α_{13} we could write this Slater determinant as

$$\Phi_\Lambda = a^\dagger_{\alpha_3} a^\dagger_{\alpha_6} a^\dagger_{\alpha_{10}} a^\dagger_{\alpha_{13}} |0\rangle.$$

The unoccupied single-particle states have bit value 0 while the occupied ones are represented by bit state 1. In the binary notation we would write this 16 bits long integer as

α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}
0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0

which translates into the decimal number

$$2^3 + 2^6 + 2^{10} + 2^{13} = 9288.$$

We can thus encode a Slater determinant as a bit pattern.

Operators in second quantization

With N particles that can be distributed over n single-particle states, the total number of Slater determinats (and defining thereby the dimensionality of the system) is

$$\dim(\mathscr{H}) = \binom{n}{N}.$$

The total number of bit patterns is 2^n.

Operators in second quantization

We assume again that we have at our disposal n different single-particle orbits $\alpha_0, \alpha_2, \ldots, \alpha_{n-1}$ and that we can distribute among these orbits $N \leq n$ particles. The ordering among these states is important as it defines the order of the creation operators. We will write the determinant

$$\Phi_\Lambda = a^\dagger_{\alpha_3} a^\dagger_{\alpha_6} a^\dagger_{\alpha_{10}} a^\dagger_{\alpha_{13}} |0\rangle,$$

in a more compact way as

$$\Phi_{3,6,10,13} = |0001001000100100\rangle.$$

The action of a creation operator is thus

$$a^\dagger_{\alpha_4}\Phi_{3,6,10,13} = a^\dagger_{\alpha_4}|0001001000100100\rangle = a^\dagger_{\alpha_4}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_3}a^\dagger_{\alpha_4}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = -|0001101000100100\rangle.$$

Operators in second quantization

Similarly

$$a^\dagger_{\alpha_6}\Phi_{3,6,10,13} = a^\dagger_{\alpha_6}|0001001000100100\rangle = a^\dagger_{\alpha_6}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_4}(a^\dagger_{\alpha_6})^2a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = 0!$$

This gives a simple recipe:

- If one of the bits b_j is 1 and we act with a creation operator on this bit, we return a null vector
- If $b_j = 0$, we set it to 1 and return a sign factor $(-1)^l$, where l is the number of bits set before bit j.

Operators in second quantization

Consider the action of $a^\dagger_{\alpha_2}$ on various slater determinants:

$$\begin{aligned}
a^\dagger_{\alpha_2}\Phi_{00111} &= a^\dagger_{\alpha_2}|00111\rangle &&= 0\times|00111\rangle\\
a^\dagger_{\alpha_2}\Phi_{01011} &= a^\dagger_{\alpha_2}|01011\rangle &&= (-1)\times|01111\rangle\\
a^\dagger_{\alpha_2}\Phi_{01101} &= a^\dagger_{\alpha_2}|01101\rangle &&= 0\times|01101\rangle\\
a^\dagger_{\alpha_2}\Phi_{01110} &= a^\dagger_{\alpha_2}|01110\rangle &&= 0\times|01110\rangle\\
a^\dagger_{\alpha_2}\Phi_{10011} &= a^\dagger_{\alpha_2}|10011\rangle &&= (-1)\times|10111\rangle\\
a^\dagger_{\alpha_2}\Phi_{10101} &= a^\dagger_{\alpha_2}|10101\rangle &&= 0\times|10101\rangle\\
a^\dagger_{\alpha_2}\Phi_{10110} &= a^\dagger_{\alpha_2}|10110\rangle &&= 0\times|10110\rangle\\
a^\dagger_{\alpha_2}\Phi_{11001} &= a^\dagger_{\alpha_2}|11001\rangle &&= (+1)\times|11101\rangle\\
a^\dagger_{\alpha_2}\Phi_{11010} &= a^\dagger_{\alpha_2}|11010\rangle &&= (+1)\times|11110\rangle
\end{aligned}$$

What is the simplest way to obtain the phase when we act with one annihilation(creation) operator on the given Slater determinant representation?

Operators in second quantization

We have an SD representation

$$\Phi_\Lambda = a^\dagger_{\alpha_0} a^\dagger_{\alpha_3} a^\dagger_{\alpha_6} a^\dagger_{\alpha_{10}} a^\dagger_{\alpha_{13}} |0\rangle,$$

in a more compact way as

$$\Phi_{0,3,6,10,13} = |1001001000100100\rangle.$$

The action of

$$a^\dagger_{\alpha_4} a_{\alpha_0} \Phi_{0,3,6,10,13} = a^\dagger_{\alpha_4} |0001001000100100\rangle = a^\dagger_{\alpha_4} a^\dagger_{\alpha_3} a^\dagger_{\alpha_6} a^\dagger_{\alpha_{10}} a^\dagger_{\alpha_{13}} |0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_3} a^\dagger_{\alpha_4} a^\dagger_{\alpha_6} a^\dagger_{\alpha_{10}} a^\dagger_{\alpha_{13}} |0\rangle = -|0001101000100100\rangle.$$

Operators in second quantization

The action

$$a_{\alpha_0} \Phi_{0,3,6,10,13} = |0001001000100100\rangle,$$

can be obtained by subtracting the logical sum (AND operation) of $\Phi_{0,3,6,10,13}$ and a word which represents only α_0, that is

$$|1000000000000000\rangle,$$

from $\Phi_{0,3,6,10,13} = |1001001000100100\rangle$.

This operation gives $|0001001000100100\rangle$.

Similarly, we can form $a^\dagger_{\alpha_4} a_{\alpha_0} \Phi_{0,3,6,10,13}$, say, by adding $|0000100000000000\rangle$ to $a_{\alpha_0} \Phi_{0,3,6,10,13}$, first checking that their logical sum is zero in order to make sure that the state α_4 is not already occupied.

Operators in second quantization

It is trickier however to get the phase $(-1)^l$. One possibility is as follows

- Let S_1 be a word that represents the 1-bit to be removed and all others set to zero.

In the previous example $S_1 = |1000000000000000\rangle$

- Define S_2 as the similar word that represents the bit to be added, that is in our case

$S_2 = |0000100000000000\rangle$.

- Compute then $S = S_1 - S_2$, which here becomes

$$S = |0111000000000000\rangle$$

- Perform then the logical AND operation of S with the word containing

$$\Phi_{0,3,6,10,13} = |1001001000100100\rangle,$$

which results in $|0001000000000000\rangle$. Counting the number of 1-bits gives the phase. Here you need however an algorithm for bitcounting.

Bit counting

We include here a python program which may aid in this direction. It uses bit manipulation functions from `http://wiki.python.org/moin/BitManipulation`.

[fontsize=,linenos=false,mathescape,baselinestretch=1.0,fontfamily=tt,xleftmargin=7mm]python
import math

""" A simple Python class for Slater determinant manipulation Bit-manipulation stolen from:

http://wiki.python.org/moin/BitManipulation """

bitCount() counts the number of bits set (not an optimal function)

def bitCount(int$_type$) : """$Countbitssetininteger$"""$count = 0 while(int_type) : int_type = int_type - 1 count+ = 1 return(count)$

testBit() returns a nonzero result, 2**offset, if the bit at 'offset' is one.

def testBit(int$_type, offset$) : $mask = 1 << offset return(int_type mask) >> offset$

setBit() returns an integer with the bit at 'offset' set to 1.

def setBit(int$_type, offset$) : $mask = 1 << offset return(int_type|mask)$

clearBit() returns an integer with the bit at 'offset' cleared.

def clearBit(int$_type, offset$) : $mask = (1 << offset) return(int_type mask)$

toggleBit() returns an integer with the bit at 'offset' inverted, 0 -> 1 and 1 -> 0.

def toggleBit(int$_type, offset$) : $mask = 1 << offset return(int_type^mask)$

binary string made from number

def bin0(s): return str(s) if s<=1 else bin0(s»1) + str(s1)

def bin(s, L = 0): ss = bin0(s) if L > 0: return '0'*(L-len(ss)) + ss else: return ss

class Slater: """ Class for Slater determinants """ def $_init_{(self):self.word=int(0)}$

def create(self, j): print "c$^+_{..}$ $+ str(j) + "|" + bin(self.word) + " >= ", Assumebit jisset, thenwereturnzero.s = 0 Checkifbit jisset.isset = testBit(self.word, j) if isset == 0 : bits = bitCount(self.word((1 << j) - 1)) s = pow(-1, bits) self.word = setBit(self.word, j)$

print str(s) + " x |" + bin(self.word) + ">" return s

def annihilate(self, j): print "c$_{..}$ $+ str(j) + "|" + bin(self.word) + " >= ", Assumebit jisnotset, thenwereturnzero.s = 0 Checkifbit jisset.isset = testBit(self.word, j) if isset == 1 : bits = bitCount(self.word((1 << j) - 1)) s = pow(-1, bits) self.word = clearBit(self.word, j)$

print str(s) + " x |" + bin(self.word) + ">" return s

Do some testing:

phi = Slater() phi.create(0) phi.create(1) phi.create(2) phi.create(3)

print

s = phi.annihilate(2) s = phi.create(7) s = phi.annihilate(0) s = phi.create(200)

Eigenvalue problems, basic definitions

Let us consider the matrix $\mathbf{A}$ of dimension n. The eigenvalues of $\mathbf{A}$ are defined through the matrix equation

$$\mathbf{A}\mathbf{x}^{(\nu)} = \lambda^{(\nu)}\mathbf{x}^{(\nu)},$$

where $\lambda^{(\nu)}$ are the eigenvalues and $\mathbf{x}^{(\nu)}$ the corresponding eigenvectors. Unless otherwise stated, when we use the wording eigenvector we mean the right eigenvector. The left eigen-

value problem is defined as

$$\mathbf{x}_L^{(\nu)}\mathbf{A} = \lambda^{(\nu)}\mathbf{x}_L^{(\nu)}$$

The above right eigenvector problem is equivalent to a set of n equations with n unknowns x_i.

Eigenvalue problems, basic definitions

The eigenvalue problem can be rewritten as

$$\left(\mathbf{A} - \lambda^{(\nu)}\mathbf{I}\right)\mathbf{x}^{(\nu)} = 0,$$

with $\mathbf{I}$ being the unity matrix. This equation provides a solution to the problem if and only if the determinant is zero, namely

$$\left|\mathbf{A} - \lambda^{(\nu)}\mathbf{I}\right| = 0,$$

which in turn means that the determinant is a polynomial of degree n in λ and in general we will have n distinct zeros.

Eigenvalue problems, basic definitions

The eigenvalues of a matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$ are thus the n roots of its characteristic polynomial

$$P(\lambda) = det(\lambda\mathbf{I} - \mathbf{A}),$$

or

$$P(\lambda) = \prod_{i=1}^{n}(\lambda_i - \lambda).$$

The set of these roots is called the spectrum and is denoted as $\lambda(\mathbf{A})$. If $\lambda(\mathbf{A}) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ then we have

$$det(\mathbf{A}) = \lambda_1\lambda_2\ldots\lambda_n,$$

and if we define the trace of $\mathbf{A}$ as

$$Tr(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$$

then

$$Tr(\mathbf{A}) = \lambda_1 + \lambda_2 + \cdots + \lambda_n.$$

Abel-Ruffini Impossibility Theorem

The *Abel-Ruffini* theorem (also known as Abel's impossibility theorem) states that there is no general solution in radicals to polynomial equations of degree five or higher.

The content of this theorem is frequently misunderstood. It does not assert that higher-degree polynomial equations are unsolvable. In fact, if the polynomial has real or complex coefficients, and we allow complex solutions, then every polynomial equation has solutions; this is the fundamental theorem of algebra. Although these solutions cannot always be computed exactly with radicals, they can be computed to any desired degree of accuracy using numerical methods such as the Newton-Raphson method or Laguerre method, and in this way they are no different from solutions to polynomial equations of the second, third, or fourth degrees.

The theorem only concerns the form that such a solution must take. The content of the theorem is that the solution of a higher-degree equation cannot in all cases be expressed in terms of the polynomial coefficients with a finite number of operations of addition, subtraction, multiplication, division and root extraction. Some polynomials of arbitrary degree, of which the simplest nontrivial example is the monomial equation $ax^n = b$, are always solvable with a radical.

Abel-Ruffini Impossibility Theorem

The *Abel-Ruffini* theorem says that there are some fifth-degree equations whose solution cannot be so expressed. The equation $x^5 - x + 1 = 0$ is an example. Some other fifth degree equations can be solved by radicals, for example $x^5 - x^4 - x + 1 = 0$. The precise criterion that distinguishes between those equations that can be solved by radicals and those that cannot was given by Galois and is now part of Galois theory: a polynomial equation can be solved by radicals if and only if its Galois group is a solvable group.

Today, in the modern algebraic context, we say that second, third and fourth degree polynomial equations can always be solved by radicals because the symmetric groups S_2, S_3 and S_4 are solvable groups, whereas S_n is not solvable for $n \geq 5$.

Eigenvalue problems, basic definitions

In the present discussion we assume that our matrix is real and symmetric, that is $\mathbf{A} \in \mathbb{R}^{n\times n}$. The matrix $\mathbf{A}$ has n eigenvalues $\lambda_1 \dots \lambda_n$ (distinct or not). Let $\mathbf{D}$ be the diagonal matrix with the eigenvalues on the diagonal

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \lambda_{n-1} & \\ 0 & \dots & \dots & \dots & \dots & 0 & \lambda_n \end{pmatrix}.$$

If $\mathbf{A}$ is real and symmetric then there exists a real orthogonal matrix $\mathbf{S}$ such that

$$\mathbf{S}^T\mathbf{A}\mathbf{S} = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n),$$

and for $j = 1 : n$ we have $\mathbf{AS}(:, j) = \lambda_j \mathbf{S}(:, j)$.

Eigenvalue problems, basic definitions

To obtain the eigenvalues of $\mathbf{A} \in \mathbb{R}^{n \times n}$, the strategy is to perform a series of similarity transformations on the original matrix $\mathbf{A}$, in order to reduce it either into a diagonal form as above or into a tridiagonal form.

We say that a matrix $\mathbf{B}$ is a similarity transform of $\mathbf{A}$ if

$$\mathbf{B} = \mathbf{S}^T \mathbf{AS}, \qquad \text{where} \qquad \mathbf{S}^T \mathbf{S} = \mathbf{S}^{-1} \mathbf{S} = \mathbf{I}.$$

The importance of a similarity transformation lies in the fact that the resulting matrix has the same eigenvalues, but the eigenvectors are in general different.

Eigenvalue problems, basic definitions

To prove this we start with the eigenvalue problem and a similarity transformed matrix $\mathbf{B}$.

$$\mathbf{Ax} = \lambda \mathbf{x} \qquad \text{and} \qquad \mathbf{B} = \mathbf{S}^T \mathbf{AS}.$$

We multiply the first equation on the left by $\mathbf{S}^T$ and insert $\mathbf{S}^T \mathbf{S} = \mathbf{I}$ between $\mathbf{A}$ and $\mathbf{x}$. Then we get

$$(\mathbf{S}^T \mathbf{AS})(\mathbf{S}^T \mathbf{x}) = \lambda \mathbf{S}^T \mathbf{x}, \tag{3.6}$$

which is the same as

$$\mathbf{B}\left(\mathbf{S}^T \mathbf{x}\right) = \lambda \left(\mathbf{S}^T \mathbf{x}\right).$$

The variable λ is an eigenvalue of $\mathbf{B}$ as well, but with eigenvector $\mathbf{S}^T \mathbf{x}$.

Eigenvalue problems, basic definitions

The basic philosophy is to

- Either apply subsequent similarity transformations (direct method) so that

$$\mathbf{S}_N^T \ldots \mathbf{S}_1^T \mathbf{AS}_1 \ldots \mathbf{S}_N = \mathbf{D}, \tag{3.7}$$

- Or apply subsequent similarity transformations so that $\mathbf{A}$ becomes tridiagonal (Householder) or upper/lower triangular (the QR method to be discussed later).
- Thereafter, techniques for obtaining eigenvalues from tridiagonal matrices can be used.
- Or use so-called power methods
- Or use iterative methods (Krylov, Lanczos, Arnoldi). These methods are popular for huge matrix problems.

Discussion of methods for eigenvalues

The general overview.

One speaks normally of two main approaches to solving the eigenvalue problem.

- The first is the formal method, involving determinants and the characteristic polynomial. This proves how many eigenvalues there are, and is the way most of you learned about how to solve the eigenvalue problem, but for matrices of dimensions greater than 2 or 3, it is rather impractical.
- The other general approach is to use similarity or unitary tranformations to reduce a matrix to diagonal form. This is normally done in two steps: first reduce to for example a *tridiagonal* form, and then to diagonal form. The main algorithms we will discuss in detail, Jacobi's and Householder's (so-called direct method) and Lanczos algorithms (an iterative method), follow this methodology.

Eigenvalues methods

Direct or non-iterative methods require for matrices of dimensionality $n \times n$ typically $O(n^3)$ operations. These methods are normally called standard methods and are used for dimensionalities $n \sim 10^5$ or smaller. A brief historical overview

Year	n	
1950	$n = 20$	(Wilkinson)
1965	$n = 200$	(Forsythe et al.)
1980	$n = 2000$	Linpack
1995	$n = 20000$	Lapack
This decade	$n \sim 10^5$	Lapack

shows that in the course of 60 years the dimension that direct diagonalization methods can handle has increased by almost a factor of 10^4 (note this is for serial versions). However, it pales beside the progress achieved by computer hardware, from flops to petaflops, a factor of almost 10^{15}. We see clearly played out in history the $O(n^3)$ bottleneck of direct matrix algorithms.

Sloppily speaking, when $n \sim 10^4$ is cubed we have $O(10^{12})$ operations, which is smaller than the 10^{15} increase in flops.

Discussion of methods for eigenvalues

If the matrix to diagonalize is large and sparse, direct methods simply become impractical, also because many of the direct methods tend to destroy sparsity. As a result large dense matrices may arise during the diagonalization procedure. The idea behind iterative methods is to project the $n-$dimensional problem in smaller spaces, so-called Krylov subspaces. Given a matrix $\mathbf{A}$ and a vector $\mathbf{v}$, the associated Krylov sequences of vectors (and thereby subspaces) $\mathbf{v}$, $\mathbf{Av}$, $\mathbf{A}^2\mathbf{v}$, $\mathbf{A}^3\mathbf{v}, \ldots$, represent successively larger Krylov subspaces.

Matrix	$\mathbf{Ax}=\mathbf{b}$	$\mathbf{Ax}=\lambda\mathbf{x}$
$\mathbf{A}=\mathbf{A}^*$	Conjugate gradient	Lanczos
$\mathbf{A}\neq\mathbf{A}^*$	GMRES etc	Arnoldi

Eigenvalues and Lanczos' method

Basic features with a real symmetric matrix (and normally huge $n > 10^6$ and sparse) $\hat{A}$ of dimension $n \times n$:

- Lanczos' algorithm generates a sequence of real tridiagonal matrices T_k of dimension $k \times k$ with $k \leq n$, with the property that the extremal eigenvalues of T_k are progressively better estimates of $\hat{A}$' extremal eigenvalues.* The method converges to the extremal eigenvalues.
- The similarity transformation is

$$\hat{T} = \hat{Q}^T \hat{A} \hat{Q},$$

with the first vector $\hat{Q}\hat{e}_1 = \hat{q}_1$.

We are going to solve iteratively

$$\hat{T} = \hat{Q}^T \hat{A} \hat{Q},$$

with the first vector $\hat{Q}\hat{e}_1 = \hat{q}_1$. We can write out the matrix $\hat{Q}$ in terms of its column vectors

$$\hat{Q} = [\hat{q}_1 \hat{q}_2 \dots \hat{q}_n].$$

Eigenvalues and Lanczos' method, tridiagonal matrix

The matrix

$$\hat{T} = \hat{Q}^T \hat{A} \hat{Q},$$

can be written as

$$\hat{T} = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \dots & \dots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 & \dots & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & \dots & \dots & 0 & \beta_{n-1} & \alpha_n \end{pmatrix}$$

Eigenvalues and Lanczos' method, tridiagonal and orthogonal matrices

Using the fact that

$$\hat{Q}\hat{Q}^T = \hat{I},$$

we can rewrite

$$\hat{T} = \hat{Q}^T \hat{A} \hat{Q},$$

as

$$\hat{Q}\hat{T} = \hat{A}\hat{Q}.$$

Eigenvalues and Lanczos' method

If we equate columns

$$\hat{T} = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \dots & \dots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 & \dots & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & \dots & \dots & 0 & \beta_{n-1} & \alpha_n \end{pmatrix}$$

we obtain

$$\hat{A}\hat{q}_k = \beta_{k-1}\hat{q}_{k-1} + \alpha_k \hat{q}_k + \beta_k \hat{q}_{k+1}.$$

Eigenvalues and Lanczos' method, defining the Lanczos' vectors

We have thus

$$\hat{A}\hat{q}_k = \beta_{k-1}\hat{q}_{k-1} + \alpha_k \hat{q}_k + \beta_k \hat{q}_{k+1},$$

with $\beta_0 \hat{q}_0 = 0$ for $k = 1 : n-1$. Remember that the vectors $\hat{q}_k$ are orthornormal and this implies

$$\alpha_k = \hat{q}_k^T \hat{A} \hat{q}_k,$$

and these vectors are called Lanczos vectors.

Eigenvalues and Lanczos' method, basic steps

We have thus

$$\hat{A}\hat{q}_k = \beta_{k-1}\hat{q}_{k-1} + \alpha_k \hat{q}_k + \beta_k \hat{q}_{k+1},$$

with $\beta_0 \hat{q}_0 = 0$ for $k = 1 : n-1$ and

$$\alpha_k = \hat{q}_k^T \hat{A} \hat{q}_k.$$

If

$$\hat{r}_k = (\hat{A} - \alpha_k \hat{I})\hat{q}_k - \beta_{k-1}\hat{q}_{k-1},$$

is non-zero, then

$$\hat{q}_{k+1} = \hat{r}_k / \beta_k,$$

with $\beta_k = \pm||\hat{r}_k||_2$.