Nuclear Shell Model, how to develop a shell-model code

Morten Hjorth-Jensen[1]

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

Spring 2016

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

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

Product wavefunction + antisymmetry = Slater determinant.

Slater determinants as basis states

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

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, 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 $\hat{a}^\dagger_i$ and an annihilation operator $\hat{a}_i$.

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

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

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,

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

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

Full Configuration Interaction Theory

We have defined the ansatz for the ground state as

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

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

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

Full Configuration Interaction Theory

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

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

where we have introduced the so-called correlation operator

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

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

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

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 $0, 1, \dots, n$ hole states and P for $0, 1, \dots, n$ 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

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

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

Since the coefficients $\delta[C_H^{*P}]$ and $\delta[C_{H'}^{P'}]$ are complex conjugates it

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

$$\sum_{P'H'} \langle \Phi_H^P | \hat{H} - E | \Phi_{H'}^{P'} \rangle = 0. \quad (1)$$

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

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

$$\left(\begin{array}{c} n \\ N \end{array} \right) = \frac{n!}{(n-N)!N!}.$$

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

$$\left(\begin{array}{c} 40 \\ 8 \end{array} \right) = \frac{40!}{(32)!8!} \sim 10^9,$$

and multiplying this with the number of proton Slater determinants we end up with approximately witha 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. (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

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

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. (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

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

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

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

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

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

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

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

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

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

$$C_{ij}^{ab}=\frac{\langle ij|\hat{v}|ab\rangle}{\epsilon_i+\epsilon_j-\epsilon_a-\epsilon_b}.$$

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.

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 $(N + 3/2)\hbar\omega$.

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 $N\hbar\Omega$ 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 $N\hbar\Omega$ 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.

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 $\begin{pmatrix} 4 \\ 2 \end{pmatrix} = 6$ 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.

Project 2

The basic goal of this project is for you to build your own configuration-interaction shell-model code. The code will be fairly basic; it will assume that we have a single species of particles, e.g. only neutrons, and you could, if you wish to, read in uncoupled two-body matrix elements. Furthermore the pieces of the code will not be the most efficient. Nonetheless it will be usable; most importantly, you will gain a good idea of what goes into a many-body shell-model code.

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.

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 $sd(7,3) = 9$.

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

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

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).

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:

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

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 $M=0$.

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, $\epsilon_k = |k|\delta$ 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.)
- Write lots of error traps, even for things that are 'obvious.'
- Document as you go along. For each function write a header that includes:
  1. Main purpose of function
  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

- When 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.

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. This is left as a challenge for those of you who would like to study this topic further for the final project to be presented for the oral examination.

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

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

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

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

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 (this is similar to project 1).
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 $v = 1, 3, 5, \dots$.

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

Example case: pairing Hamiltonian

In this basis, the operator

$$\hat{P}_+ = \hat{a}^\dagger_1\hat{a}^\dagger_2 + \hat{a}^\dagger_3\hat{a}^\dagger_4 + \hat{a}^\dagger_5\hat{a}^\dagger_6$$

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 ($|1,2,3\rangle, |1,2,5\rangle, |1,4,6\rangle, |2,3,4\rangle, |2,3,6\rangle, |2,4,5\rangle, |2,5,6\rangle, |3,4,6\rangle, |4,5,6\rangle$ 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:

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

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 $i_1 < i_2 < i_3 \ldots i_N$.
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

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

as well as the phase
This is equivalent to computing

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

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 $i \rightarrow r$ and $j \rightarrow r$.

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. Loop over $\alpha = 1, NSD$
3. For each α, loop over $a = 1, ntbme$ and fetch $V(a)$ and the single-particle indices p, q, r, s
4. 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.
5. Find, if any, the label β of the resulting Slater determinant and the phase (which is 0, +1, -1).
6. 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.
7. Continue loop over a
8. Continue loop over α.

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 $\epsilon(i)$.