

1 Momentum distributions

2 Second quantization

This section will be somewhat over-elaborated. But it can serve as a recapitulation of second quantization.

The one body momentum distribution operator is defined as,

$$\hat{n}(p) = \frac{1}{(2\pi)^3} \int d^2\Omega_{\mathbf{p}} a_{\mathbf{p}}^\dagger a_{\mathbf{p}} \quad (1)$$

It's action on a multi particle ground state $|\Phi\rangle$,

$$\langle\Phi|\hat{n}(p)|\Phi\rangle = \frac{1}{(2\pi)^3} \int d^2\Omega_{\mathbf{p}} \langle\Phi|a_{\mathbf{p}}^\dagger a_{\mathbf{p}}|\Phi\rangle \quad (2)$$

The creation and annihilation operators $a_{\mathbf{p},\mathbf{p}}^\dagger, a_{\mathbf{p}}$ have only meaning working on particles with definite momentum or the vacuum state $|0\rangle$.

$$\langle\Phi|a_{\mathbf{p}}^\dagger a_{\mathbf{p}}|\Phi\rangle = \int d^3\mathbf{p}_1 \dots d^3\mathbf{p}_A \langle\Phi|\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_A\rangle \langle\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_A|a_{\mathbf{p}}^\dagger a_{\mathbf{p}}|\Phi\rangle \quad (3)$$

$$= \int d^A\mathbf{p}_1 \dots d^3\mathbf{p}_A \langle\Phi|\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_A\rangle \langle 0|a_{\mathbf{p}_1} a_{\mathbf{p}_2} \dots a_{\mathbf{p}_A} a_{\mathbf{p}}^\dagger a_{\mathbf{p}}|\Phi\rangle \quad (4)$$

Using the anticommutation relation $\{a_{\mathbf{p}}, a_{\mathbf{q}}^\dagger\} = \delta(\mathbf{p} - \mathbf{q})$, we get

$$\langle 0|a_{\mathbf{p}_1} a_{\mathbf{p}_2} \dots a_{\mathbf{p}_A} a_{\mathbf{p}}^\dagger a_{\mathbf{p}}|\Phi\rangle = \langle 0|a_{\mathbf{p}_1} a_{\mathbf{p}_2} \dots \delta(\mathbf{p} - \mathbf{p}_A) a_{\mathbf{p}}|\Phi\rangle - \langle 0|a_{\mathbf{p}_1} a_{\mathbf{p}_2} \dots a_{\mathbf{p}_{A-1}} a_{\mathbf{p}}^\dagger a_{\mathbf{p}_A} a_{\mathbf{p}}|\Phi\rangle \quad (5)$$

$$= \delta(\mathbf{p} - \mathbf{p}_A) \langle\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}|\Phi\rangle - \delta(\mathbf{p} - \mathbf{p}_{A-1}) \langle 0|a_{\mathbf{p}_1} \dots a_{\mathbf{p}_{A-2}} a_{\mathbf{p}_A} a_{\mathbf{p}}|\Phi\rangle \quad (6)$$

$$+ \langle 0|a_{\mathbf{p}_1} \dots a_{\mathbf{p}_{A-2}} a_{\mathbf{p}}^\dagger a_{\mathbf{p}_{A-1}} a_{\mathbf{p}_A} a_{\mathbf{p}}|\Phi\rangle \quad (7)$$

$$= \delta(\mathbf{p} - \mathbf{p}_A) \langle\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_A|\Phi\rangle + \delta(\mathbf{p} - \mathbf{p}_{A-1}) \langle\mathbf{p}_1 \dots \mathbf{p}_{A-2} \mathbf{p}_{A-1} \mathbf{p}_A|\Phi\rangle \quad (8)$$

$$+ \langle 0|a_{\mathbf{p}_1} \dots a_{\mathbf{p}_{A-2}} a_{\mathbf{p}}^\dagger a_{\mathbf{p}_{A-1}} a_{\mathbf{p}_A} a_{\mathbf{p}}|\Phi\rangle = \dots \quad (9)$$

$$= \sum_{i=1}^A \delta(\mathbf{p} - \mathbf{p}_i) \langle\mathbf{p}_1 \dots \mathbf{p}_A|\Phi\rangle + (-1)^A \underbrace{\langle 0|a_{\mathbf{p}}^\dagger a_{\mathbf{p}_1} \dots a_{\mathbf{p}_A} a_{\mathbf{p}}|\Phi\rangle}_{=0} \quad (10)$$

Hence,

$$\langle\Phi|a_{\mathbf{p}}^\dagger a_{\mathbf{p}}|\Phi\rangle = \int d^3\mathbf{p}_1 \dots d^3\mathbf{p}_A \langle\Phi|\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_A\rangle \sum_{i=1}^A \delta(\mathbf{p} - \mathbf{p}_i) \langle\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_A|\Phi\rangle \quad (11)$$

If $|\Phi\rangle$ is a slater determinant of orthonormal single particle wave functions $|\phi_{\alpha_i}\rangle$ we get,

$$\langle\Phi|a_{\mathbf{p}}^\dagger a_{\mathbf{p}}|\Phi\rangle = \sum_{i=1}^A |\langle\mathbf{p}|\phi_{\alpha_i}\rangle|^2 = \sum_{i=1}^A \phi_{\alpha_i}^\dagger(\mathbf{p}) \phi_{\alpha_i}(\mathbf{p}) \quad (12)$$

Note that we also could have derived this result by instead of inserting the unit $\prod_{i=1}^A d^3\mathbf{p}_i |\mathbf{p}_i\rangle \langle\mathbf{p}_i|$ we expand $|\Phi\rangle$ in terms of single particle creation operators,

$$a_{\mathbf{p}}^\dagger a_{\mathbf{p}} |\Phi\rangle = a_{\mathbf{p}}^\dagger a_{\mathbf{p}} |\alpha_1 \alpha_2 \dots \alpha_A\rangle = a_{\mathbf{p}}^\dagger a_{\mathbf{p}} a_{\alpha_1}^\dagger a_{\alpha_2}^\dagger \dots a_{\alpha_A}^\dagger |0\rangle \quad (13)$$

The commutation relations between $a_{\mathbf{p}}$ and a_{α_i} are easily derived by expanding a_{α_i} in momentum creation operators,

$$a_{\alpha_i}^\dagger = \int d^3\mathbf{k} \phi_{\alpha_i}(\mathbf{k}) a_{\mathbf{k}}^\dagger \quad (14)$$

$$\Rightarrow a_{\mathbf{p}} a_{\alpha_i}^\dagger = \int d^3\mathbf{k} \phi_{\alpha_i}(\mathbf{k}) a_{\mathbf{p}} a_{\mathbf{k}}^\dagger = \phi_{\alpha_i}(\mathbf{p}) - a_{\alpha_i}^\dagger a_{\mathbf{p}} \quad (15)$$

So,

$$a_{\mathbf{p}} |\Phi\rangle = a_{\mathbf{p}} a_{\alpha_1}^\dagger a_{\alpha_2}^\dagger \dots a_{\alpha_A}^\dagger |0\rangle = (\phi_{\alpha_1}(\mathbf{p}) - a_{\alpha_1}^\dagger a_{\mathbf{p}}) a_{\alpha_2}^\dagger \dots a_{\alpha_A}^\dagger |0\rangle \quad (16)$$

$$= \sum_{i=1}^A (-1)^{i-1} \phi_{\alpha_i}(\mathbf{p}) |\alpha_1 \dots \alpha_{i-1} \alpha_{i+1} \dots \alpha_A\rangle \quad (17)$$

The conjugate gives,

$$\langle \Phi | a_{\mathbf{p}}^\dagger = \sum_{j=1}^A (-1)^{j-1} \langle \alpha_1 \dots \alpha_{j-1} \alpha_{j+1} \dots \alpha_A | \phi_{\alpha_j}^\dagger(\mathbf{p}) \quad (18)$$

Hence,

$$\langle \Phi | a_{\mathbf{p}}^\dagger a_{\mathbf{p}} | \Phi \rangle = \sum_{i,j=1}^A (-1)^{i+j} \phi_{\alpha_j}^\dagger(\mathbf{p}) \phi_{\alpha_i}(\mathbf{p}) \underbrace{\langle \alpha_1 \dots \alpha_{j-1} \alpha_{j+1} \dots \alpha_A | \alpha_1 \dots \alpha_{i-1} \alpha_{i+1} \dots \alpha_A \rangle}_{=\delta_{ij}} \quad (19)$$

$$= \sum_i \phi_{\alpha_i}^\dagger(\mathbf{p}) \phi_{\alpha_i}(\mathbf{p}) \quad (20)$$

Which is exactly the same result as before.

So the one body momentum distribution is given by,

$$\langle \Phi | \hat{n}(p) | \Phi \rangle = \sum_{i=1}^A \frac{1}{(2\pi)^3} \int d^2\Omega_{\mathbf{p}} \phi_{\alpha_i}^\dagger(\mathbf{p}) \phi_{\alpha_i}(\mathbf{p}) \quad (21)$$

Note that this distribution is normed to the number of particles A . To get the momentum distribution normed to unity we have to divide by A ,

$$\langle \Phi | \hat{n}(p) | \Phi \rangle = \frac{1}{A} \sum_{i=1}^A \frac{1}{(2\pi)^3} \int d^2\Omega_{\mathbf{p}} \phi_{\alpha_i}^\dagger(\mathbf{p}) \phi_{\alpha_i}(\mathbf{p}) \quad (22)$$

3 Nucleus

3.1 shell.h

This class contains the quantum number of a shell nlj . It has two (proton & neutron) static arrays containing all the shells.

```
shellsN = [ Shell(n1,l1,j1), Shell(n2,l2,j2), ... ]
shellsP = [ Shell(n1,l1,j1), Shell(n2,l2,j2), ... ]
```

These two arrays are initialised and deleted by the static methods `Shell::initialiseShells`, `Shell::deleteShells`.

3.2 nucleus.h

Main method here is `Nucleus::makePairs`. Note that this relies on overloaded virtual functions to function. It iterates over the quantum numbers, $n_1 l_1 j_1 m_{j_1}, n_2 l_2 j_2 m_{j_2}$ and makes a pair for each of these combinations: `Pair::Pair(mosh, n1, l1, j1, mj1, t1, n2, l2, j2, mj2, t2)`. `mosh` is the return value of `RecMosh::createRecMosh(n1, l1, n2, l2, inputdir, outputdir)`, being a `RecMosh` object. The moshinsky brackets $\langle n_1 l_1 n_2 l_2; \Lambda | nlNL; \Lambda \rangle$ can be accessed by calling `RecMosh::getCoefficient(n, l, N, L, Lambda)`. Open shells are taken care of by calculating a open shell correction factor and applying it to the pair via `Pair::setfnorm(factor)`.

4 Pair coupling

4.1 pair.h

This class represents the state

$$|\alpha_1, \alpha_2\rangle_{\text{nas}}, |\alpha\rangle \equiv |nljm_j tm_t\rangle \quad (23)$$

The class calculates all the coefficients,

$$C_{\alpha_1 \alpha_2}^A = \langle A \equiv \{nlSjm_j, NLM_L TM_T\} | \alpha_1 \alpha_2 \rangle \quad (24)$$

The main method here is `Pair::makecoeflist()`. It loops over all possible values of $A \equiv \{S, T, n, l, N, M_L, j, m_j\}$. Where in the summation over $\{n, l, N, L\}$ the energy conservation $2n_1 + l_1 + 2n_2 + l_2 = 2n + l + 2N + L$ is taken into account to eliminate one of the summation loops, $L = 2n_1 + l_1 + 2n_2 + l_2 - 2n - l - 2N$. Note that M_T is also fixed by $M_T = m_{t_1} + m_{t_2}$ and no summation over this is performed, as we want to keep the contribution from different pairs separated. For each A a new object `Newcoef` is generated and stored in the member `std::vector<NewCoef*> coeflist`.

4.2 newcoef.h

This class takes the parameters $n_1 l_1 j_1 m_{j_1} m_{t_1} n_2 l_2 j_2 m_{j_2} m_{t_2} NLM_L nlSjm_j TM_T$, and calculates the coefficient given in Eq. (24). It takes also a pointer to a `RecMosh` object that holds the Moshinsky brackets. The only function in this class is to calculate $C_{\alpha_1 \alpha_2}^A$ using the formula,

$$\sum_{JM_J} \sum_{\Lambda} [1 - (-1)^{L+S+T}] \langle t_1 m_{t_1} t_2 m_{t_2} | TM_T \rangle \langle j_1 m_{j_1} j_2 m_{j_2} | JM_J \rangle \langle j m_j LM_L | JM_J \rangle \\ \langle nlNL; \Lambda | n_1 l_1 n_2 l_2; \Lambda \rangle_{\text{SMB}} \sqrt{2\Lambda + 1} \sqrt{2j + 1} \left\{ \begin{matrix} j & L & J \\ \Lambda & S & l \end{matrix} \right\} \\ \sqrt{2j_1 + 1} \sqrt{2j_2 + 1} \sqrt{2S + 1} \sqrt{2\Lambda + 1} \left\{ \begin{matrix} l_1 & s_1 & j_1 \\ l_2 & s_2 & j_2 \\ \Lambda & S & J \end{matrix} \right\} \quad (25)$$

It is easy to check that the result indeed depends on α_1, α_2, A . Note that it is always assumed that $s_i, t_i \equiv \frac{1}{2}$ as we are dealing with protons and neutrons. This class also defines a ‘‘key’’ to be able to index the coefficients, `key = ‘‘nlSjm_j.NLM_L.TM_T’’`.

5 paircoef.h

This is a very thin class designed to do some bookkeeping. As outlined in Maartens thesis pg 156, different $|\alpha_1 \alpha_2\rangle$ combinations will sometimes map to the same ‘‘rcm’’ states $A = |nlSjm_j NLM_L TM_T\rangle$. In matrix element calculations,

$$\langle \alpha_1 \alpha_2 | \hat{O} | \alpha_1 \alpha_2 \rangle = \sum_{AB} C_{\alpha_1 \alpha_2}^{A\dagger} C_{\alpha_1 \alpha_2}^B \langle A | \hat{O} | B \rangle \quad (26)$$

We want to calculate matrix elements as $\langle A|\hat{O}|B\rangle$ only once. $|\alpha_1\alpha_2\rangle$ that map to the same A, B states should lookup the earlier calculated values for $\langle A|\hat{O}|B\rangle$. In general the matrix element $\langle A|\hat{O}|B\rangle$ is not diagonal. A `Paircoef` object has all the quantum numbers in a rcm state A . In addition it holds a value and a map `std::map<Paircoef*, double>`. The map is used to link a rcm state $|A\rangle$ to all other rcm states $|B\rangle$ which yield a non zero contribution for $\langle A|\hat{O}|B\rangle$. The value for the transformation coefficients $C_{\alpha_1, \alpha_2}^{A, \dagger} C_{\alpha_1, \alpha_2}^B$ is stored in the second field of the map (`double`). So that the the summation over B (Eq. 26) is replaced by,

$$\langle \alpha_1 \alpha_2 | \hat{O} | \alpha_1 \alpha_2 \rangle = \sum_A \sum_{\text{Paircoef}(A).links} \text{link.strength} \langle A | \hat{O} | B \rangle \quad (27)$$

`Paircoef::add(double val)` adds `val` to private member `value` but as far as I can see this private member `value` is NEVER used!