

Symmetries and Exact Diagonalization

Peter Brown

November 15, 2018

Suppose we have a cluster of N sites at positions (X_i, Y_i) , $i \in \{0, \dots, N-1\}$ which lie on a periodic lattice and some lattice Hamiltonian describing spins, bosons or fermions. We are interested in some property of this Hamiltonian, such as the ground state energy, conductivity, or correlations between different sites. One approach to determining these properties is to write the Hamiltonian down in a certain basis and then diagonalize it on a computer. If the cluster of sites has a discrete spatial symmetry (such as translation, rotation, reflection, or inversion symmetry), we can take advantage of this to reduce the size of the matrices we need to diagonalize.

In the simplest case, suppose we have a symmetry represented by an operator R such that $R^m = \mathbb{I}$. One example of this situation is four-fold rotation on a square lattice. Then we have $[H, R] = 0$, which implies H and R are simultaneously diagonalizable. In fact, if we choose a basis such that R is diagonal, H is guaranteed to be block diagonal. We will find that such a basis is easy to construct once we have explicitly written down R .

We will also encounter more complicated cases where we have several distinct symmetry operations which do not commute. For example, rotation and reflection symmetry on a square lattice. This situation will naturally lead us towards discussing the symmetry group of a cluster. We will find that we can still find a basis which respects our symmetry group and in which H is block diagonal. The blocks of H are associated with different irreducible representations of the symmetry group.

1 Symmetry Transformations

Suppose our transformation acts on real space coordinates as a linear operator, T .

$$\begin{pmatrix} X_i^t \\ Y_i^t \end{pmatrix} = T \begin{pmatrix} X_i \\ Y_i \end{pmatrix}. \quad (1)$$

We can also think about this transformation as a function acting on site indices, $f_t(i)$, where $f_t(i) = j$ if and only if $X_i^t = X_j$ and $Y_i^t = Y_j$.

In more mathematical language, we would say that f_t is a bijection from the set $\{0, \dots, n-1\}$ to itself, and hence a member of S_n , the n th symmetric group. Often functions like this are written

$$\begin{pmatrix} 0 & 1 & \dots & n-1 \\ f_t(0) & f_t(1) & \dots & f_t(n-1) \end{pmatrix}. \quad (2)$$

For example, suppose we have a collection of four sites arranged in a square, numbered 0, 1, 2, 3 going from the left to right and top to bottom. Suppose that T is a counterclockwise transformation

by $\pi/2$ radians. We would write this

$$f_t = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 3 & 1 \end{pmatrix}.$$

We are interested in the *cycles* of sites that transform into one another as we apply our transformation repeatedly. Let a cycle be an ordered list $c = (i_0, i_1, \dots, i_{n-1})$ where $i_k = f_t(i_{k-1})$, $f_t(i_{n-1}) = i_0$, and $f_t(j) = j$ for any j which doesn't appear in the cycle. We can write our transformation as a collection of disjoint cycles. In the example above, there is only one cycle, $f_t = (0, 2, 3, 1)$.

Consider a 3×3 square, where

$$f_t = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 3 & 0 & 7 & 4 & 1 & 8 & 5 & 2 \end{pmatrix}.$$

This transformation has three cycles, so we can think roughly

$$f_t = (0, 6, 8, 2) \times (1, 3, 7, 5) \times (4).$$

We can think of a cycle of length 2 as a *swap* operation, because it exchanges two sites. Any cycle can be conveniently written as a product of swap operations.

Theorem 1.1 (Cycles decomposed to swap operators). *Any cycle can be written as a collection of two site swap operators, according to*

$$(i_0, i_1, \dots, i_{n-1}) = (i_0, i_1) \times (i_1, i_2) \times \dots \times (i_{n-3}, i_{n-2}) \times (i_{n-2}, i_{n-1}) \quad (3)$$

For example, we can write $(0, 1, 2) = (0, 1) \times (1, 2)$.

Theorem 1.2 (Swap operators decomposed to nearest-neighbor swap operators). *We can check that*

$$(n, n+m) = (n, n+1) \times (n+1, n+2) \times \dots \quad (4)$$

$$\times (n+m-1, n+m) \times (n+m-2, n+m-1) \times \dots \quad (5)$$

$$\times (n, n+1). \quad (6)$$

At this point, we want to start thinking about how the transformation operator R_T related to this transform T operates on our quantum states. Now we construct R by decomposing our cycles into swap operators, our swap operators into nearest neighbor swap operators, and using the expression we found for nearest neighbor swap operators. Suppose our transformation is composed of N cycles c_n , where the n th cycle has length m_n , then we can write

$$f_t = \prod_{n=0}^N c_n = \prod_{n=0}^N (i_0^n, i_1^n, \dots, i_{m_n}^n) \quad (7)$$

$$= \prod_{n=0}^N [(i_0^n, i_1^n) \times \dots \times (i_{m_n-1}^n, i_{m_n}^n)] \quad (8)$$

$$= \prod_{n=0}^N \{[(i_0^n, i_0^n + 1) \times \dots \times (i_0^n, i_0^n + 1)] \times [(i_1^n, i_1^n + 1) \times \dots \times (i_1^n, i_1^n + 1)] \times \dots\} \quad (9)$$

$$R_T = \prod_{n=0}^N \{\text{Swap}(i_0^n, i_0^n + 1) \times \dots \times \text{Swap}(i_0^n, i_0^n + 1) \times \dots\}. \quad (10)$$

I later realized it is possible to write down a general formula for swapping states i and j directly, without doing the final decomposition into nearest neighbor swap operators. In this case, the second to last expression in the above expansion can be used directly to calculate the transformation matrix.

2 Quantum Swap Operators

2.1 Spin Systems

Suppose we have a single spin with basis states $\{|\uparrow\rangle, |\downarrow\rangle\}$. Define the spin operators according to

$$S^+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad S^- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad S^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (11)$$

Consider a system of two spins in the tensor product ordered basis $\{|\uparrow\uparrow\rangle, |\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle, |\downarrow\downarrow\rangle\}$. The operator that swaps these two spins can be written by inspections

$$\text{Swap} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = S^+ \otimes S^- + S^- \otimes S^+ + \frac{1}{2} (\mathbb{I}_{4 \times 4} + S^z \otimes S^z), \quad (12)$$

which leads us to the swap operator for sites m and $m+1$ in a system of n spins

$$\text{Swap}(m, m+1) = \mathbb{I}_1 \otimes \dots \otimes \mathbb{I}_{m-1} \otimes \text{Swap} \otimes \mathbb{I}_{m+2} \otimes \dots \otimes \mathbb{I}_n \quad (13)$$

$$\text{Swap}(i, j) = S_i^+ S_j^- + S_i^- S_j^+ + \frac{1}{2} (\mathbb{I} + S_i^z S_j^z). \quad (14)$$

2.2 Bosons

Use tensor product basis of Fock states. So the basis for a single site is $\{|0\rangle, |1\rangle, \dots, |N\rangle\}$ where we truncate to at most N bosons in the same state. For two sites, the basis is ordered

$$\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, \dots, |0\rangle \otimes |N\rangle, |1\rangle \otimes |1\rangle, \dots, |N\rangle \otimes |N\rangle\}. \quad (15)$$

TODO: What is swap operator?

2.3 Fermions

In the Fermionic case, specifying the number of particles in a certain state does not fully specify that state. This is due to the Fermionic statistics, $\{c_{k\sigma}, c_{k'\sigma'}\} = 0$, which enforce a minus sign when we flip two spins. So, for example, $c_{2\uparrow}^\dagger c_{1\downarrow}^\dagger |0\rangle = -c_{1\downarrow}^\dagger c_{2\uparrow}^\dagger |0\rangle$.

There is still a lot of utility in using binary numbers to express our states, but we must additionally choose an ordering for our Fermion operators. One natural choice is to place our Fermion operators in order of the site number, with the highest number to the left. Using this convention, we would say the binary number 1010 specifies the state $c_3^\dagger c_1^\dagger |0\rangle$.

It is instructive to consider first the case of a single spinless fermion in the basis $\{|0\rangle, c^\dagger |0\rangle\}$. In this space we have operators

$$c^\dagger = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad c = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \text{and } P = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (16)$$

where P measures the parity of the number of Fermions in a given state.

Suppose that we want to consider a space of two fermions. Now we must choose a two Fermion basis. One easy way of doing this is defining a normal ordering of our fermions operators. Once we agree on a normal ordering, we need only specify the number of Fermions on each site as a binary number. Let us first label our fermionic creation operators with integers, and order them by placing operators with higher indices further to the left. For two Fermions we have basis $\{|0\rangle, c_2^\dagger |0\rangle, c_1^\dagger |0\rangle, c_2^\dagger c_1^\dagger |0\rangle\}$. The order of c_2^\dagger and c_1^\dagger is inspired by the natural ordering of tensor product states. Explicitly, our basis is

$$\mathcal{B} = \{00, 01, 10, 11\} \quad (17)$$

$$= \{|0\rangle, c_2^\dagger |0\rangle, c_1^\dagger |0\rangle, c_2^\dagger c_1^\dagger |0\rangle\} \quad (18)$$

In this two Fermion basis our operators are

$$c_1^\dagger = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} = c^\dagger \otimes P \quad (19)$$

$$c_2^\dagger = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \mathbb{I}_{2 \times 2} \otimes c^\dagger \quad (20)$$

$$P_{1 \otimes 2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = P \otimes P \quad (21)$$

$$\text{Swap}(2, 1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = c_2^\dagger c_1 - c_2 c_1^\dagger + (\mathbb{I} - c_1^\dagger c_1 - c_2^\dagger c_2). \quad (22)$$

Generalizing these results to a chain of N fermions, we find

$$\mathcal{B} = \{0\dots 0, 0\dots 01, 0\dots 010, 0\dots 011, \dots, 1\dots 100, 1\dots 101, 1\dots 110, 1\dots 1\} \quad (23)$$

$$= \{|0\rangle, c_N^\dagger |0\rangle, c_{N-1}^\dagger |0\rangle, c_N^\dagger c_{N-1}^\dagger |0\rangle, c_{N-3}^\dagger |0\rangle, c_N^\dagger c_{N-3}^\dagger |0\rangle, \quad (24)$$

$$\dots, c_N^\dagger \dots c_3^\dagger |0\rangle, c_N^\dagger \dots c_3^\dagger c_1^\dagger |0\rangle, c_N^\dagger \dots c_3^\dagger c_2^\dagger |0\rangle, c_N^\dagger \dots c_1^\dagger |0\rangle\}$$

$$c_n^\dagger = \mathbb{I}_{2 \times 2} \otimes \dots \otimes \mathbb{I}_{2 \times 2} \otimes c^\dagger \otimes P \otimes \dots \otimes P \quad (25)$$

$$\text{Swap}(i, i+1) = \mathbb{I}_{2 \times 2} \otimes \dots \otimes \mathbb{I}_{2 \times 2} \otimes \text{Swap}(1, 2) \otimes \mathbb{I}_{2 \times 2} \otimes \dots \otimes \mathbb{I}_{2 \times 2} \quad (26)$$

$$\text{Swap}(i, j) = c_i^\dagger c_j - c_i c_j^\dagger + \left(\mathbb{I} - c_i^\dagger c_i - c_j^\dagger c_j \right) \quad (27)$$

where we have $n-1$ identity matrices to the left of c^\dagger , $N-n$ parity matrices P to the right of c^\dagger , and $i > j$.

If we instead choose to order our Fermion states with *lower* states to the left, we would find c^\dagger as above, but with the identity and parity operators swapped in the tensor product.

$$\mathcal{B} = \{0\dots 0, 0\dots 01, 0\dots 010, 0\dots 011, \dots, 1\dots 100, 1\dots 101, 1\dots 110, 1\dots 1\} \quad (28)$$

$$= \{|0\rangle, c_N^\dagger |0\rangle, c_{N-1}^\dagger |0\rangle, c_{N-1}^\dagger c_N^\dagger |0\rangle, c_{N-3}^\dagger |0\rangle, c_{N-3}^\dagger c_N^\dagger |0\rangle, \quad (29)$$

$$\dots, c_1^\dagger \dots c_{N-2}^\dagger |0\rangle, c_1^\dagger \dots c_{N-2}^\dagger c_N^\dagger |0\rangle, c_1^\dagger \dots c_{N-2}^\dagger c_{N-1}^\dagger |0\rangle, c_1^\dagger \dots c_N^\dagger |0\rangle\}$$

$$c_n^\dagger = P \otimes \dots \otimes P \otimes c^\dagger \otimes \mathbb{I}_{2 \times 2} \otimes \dots \otimes \mathbb{I}_{2 \times 2} \quad (30)$$

$$\text{Swap}(i, i+1) = \mathbb{I}_{2 \times 2} \otimes \dots \otimes \mathbb{I}_{2 \times 2} \otimes \text{Swap}(1, 2) \otimes \mathbb{I}_{2 \times 2} \otimes \dots \otimes \mathbb{I}_{2 \times 2} \quad (31)$$

$$\text{Swap}(i, j) = c_i^\dagger c_j - c_i c_j^\dagger + \left(\mathbb{I} - c_i^\dagger c_i - c_j^\dagger c_j \right). \quad (32)$$

Now we want to consider the situation where we have N sites and two spin states. In the spinless case we signified a basis state with a single binary number. We are free to do the same in the spinful case, but we have some freedom as to how we construct this number. One option is to produce on binary number for the spin up occupations, another for the spin downs, and two concatenate these to form a single composite number.

Using a single binary number to describe the occupations of the spinful system suggests that we are regarding it as a spinless system having twice the number of sites as before ($2N$ instead of N). The spin indices can be thought of extra site indices because at the anti-commutator level the two behave in the same way. If we specify a mapping from our two indices (spin and site) onto a single logical index we can construct creation operators and swap operators as in the spinless case. As in that case, we must also choose how to order the creation operators in our basis states.

We can define one normal ordering for creation operators by placing any up-spins to the left of any down-spins, and then higher site indices to the left of lower indices of the same spin state, e.g. $c_{5\uparrow}^\dagger c_{3\uparrow}^\dagger c_{2\uparrow}^\dagger c_{1\uparrow}^\dagger c_{2\downarrow}^\dagger c_{1\downarrow}^\dagger |0\rangle$. This ordering of creation operators suggests we regard all spin-up sites as having a larger index than any spin down site. This suggests we take our two-index to one-index mapping to be $(i, \uparrow) \rightarrow i + N$ and $(i, \downarrow) \rightarrow i$. This amounts to writing $c_{i\uparrow}^\dagger = \tilde{c}_{N+i}^\dagger$ and $c_{i\downarrow}^\dagger = \tilde{c}_i^\dagger$,

where \tilde{c}^\dagger are the operators in the $2N$ site space. Using these two conventions, we find

$$\mathcal{B} = \{0\dots 0, 0\dots 01, 0\dots 010, 0\dots 011, \dots, 1\dots 100, 1\dots 101, 1\dots 110, 1\dots 1\} \quad (33)$$

$$= \{|0\rangle, c_{N\uparrow}^\dagger |0\rangle, c_{N-1\uparrow}^\dagger |0\rangle, c_{N\uparrow}^\dagger c_{N-1\uparrow}^\dagger |0\rangle, \dots, \quad (34)$$

$$c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{3\downarrow}^\dagger |0\rangle, c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{3\downarrow}^\dagger c_{1\downarrow}^\dagger |0\rangle, \\ c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{3\downarrow}^\dagger c_{2\downarrow}^\dagger |0\rangle, c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{1\downarrow}^\dagger |0\rangle\}$$

$$c_{i\uparrow}^\dagger = \mathbb{I}_{2^{N+i-1} \times 2^{N+i-1}} \otimes c^\dagger \otimes P_{2^{N-i} \times 2^{N-i}} \quad (35)$$

$$c_{i\downarrow}^\dagger = \mathbb{I}_{2^{i-1} \times 2^{i-1}} \otimes c^\dagger \otimes P_{2^{2N-i} \times 2^{2N-i}} \quad (36)$$

$$\text{Swap}(i, j) = \text{Swap}(N+i, N+j) \times \text{Swap}(i, j) \quad (37)$$

We can define another normal ordering by placing higher site indices to the left, and then spin up indices to the left of spin down indices for the same site, e.g. $c_{5\uparrow}^\dagger c_{3\uparrow}^\dagger c_{2\uparrow}^\dagger c_{2\downarrow}^\dagger c_{1\uparrow}^\dagger c_{1\downarrow}^\dagger |0\rangle$. This again suggests a mapping from two-site indices to one-site indices, $(i, \uparrow) \rightarrow 2i$, $(i, \downarrow) \rightarrow 2i-1$. This amounts to writing $c_{i\uparrow}^\dagger = \tilde{c}_{2i}^\dagger$ and $c_{i\downarrow}^\dagger = \tilde{c}_{2i-1}^\dagger$. With this ordering of fermion operators and choice of index mapping we have

$$\mathcal{B} = \{0\dots 0, 0\dots 01, 0\dots 010, 0\dots 011, \dots, 1\dots 100, 1\dots 101, 1\dots 110, 1\dots 1\} \quad (38)$$

$$= \{|0\rangle, \dots\} \quad (39)$$

$$c_{i\uparrow}^\dagger = \mathbb{I}_{2^{2i-1} \times 2^{2i-1}} \otimes c^\dagger \otimes P_{2^{2N-2i} \times 2^{2N-2i}} \quad (40)$$

$$c_{i\downarrow}^\dagger = \mathbb{I}_{2^{2i-2} \times 2^{2i-2}} \otimes c^\dagger \otimes P_{2^{2N-2i-1} \times 2^{2N-2i-1}} \quad (41)$$

$$\text{Swap}(i, j) = \text{Swap}(2i-1, 2j-1) \times \text{Swap}(2i, 2j) \quad (42)$$

What happens if we instead reverse our normal ordering so that *smaller* numbers go to the left of larger number? Then we have to swap the order of P and \mathbb{I} in our tensor products, as we did in the spinless fermion case.

We are free to make many other choices of index mappings and creation operator orderings. We could even take the spin-ordering where we took all spin-ups to the left of any spin-downs but choose the mapping $(i, \uparrow) \rightarrow 2i$, $(i, \downarrow) \rightarrow 2i-1$. But if we make this choice, it is not clear how to map our system back to the spinless fermions. We would need to do more work to find the representations of our creation operators.

For a Fermion system with N sites and s spin states, the Hilbert space scales as 2^{sN} . In many systems of interest, including Hubbard systems, the total number in each spin state is conserved. If we restrict our attention to the subspace with N_s Fermions in spin state s , then the size of the state space is reduced to

$$\prod_s \binom{N}{N_s} = \prod_s \frac{N!}{N_s!(N-N_s)!} < 2^{sN}. \quad (43)$$

We would like to write matrix elements for the operators $c_i^\dagger c_j$ in this space (clearly the matrix elements for c_i^\dagger and c_j on their own are zero). However, this is not a neat tensor product expression. To the best of my knowledge, it requires doing many loops. This can be made efficient in a low level language (e.g. C), but is slow in Matlab or Python.

3 Block Diagonalizing H

3.1 Cyclic Symmetry Groups (A Test Case)

Theorem 3.1 (Eigenstates of R). *Let $z = e^{2\pi i/m}$, then $\{z^0, \dots, z^{m-1}\}$ are the m th roots of unity. Suppose $|\psi\rangle$ is any state that is not an eigenstate of R , then we can construct $|\psi\rangle_k$, which is either an eigenstate of R with eigenvalue z^k , or 0:*

$$|\psi\rangle_k = \sum_{n=0}^{m-1} R^n e^{-i\frac{2\pi}{m}kn} |\psi\rangle \quad (44)$$

$$= R_k |\psi\rangle \quad (45)$$

$$R_k = \sum_{n=0}^{m-1} R^n e^{-i\frac{2\pi}{m}kn} \quad (46)$$

$$R |\psi\rangle_k = e^{i\frac{2\pi}{m}k} |\psi\rangle_k. \quad (47)$$

But properly normalizing $|\psi\rangle_k$ for a general state is challenging.

Theorem 3.2 (R and minimum cycle length). *Suppose $|\psi\rangle$ is an arbitrary state, then $R^l |\psi\rangle = |\psi\rangle$ for some values of l . The smallest value of l that satisfies this relationship must be a divisor of m .*

For example, if $m = 4$, then for any state we pick, either $R |\psi\rangle = |\psi\rangle$, $R^2 |\psi\rangle = |\psi\rangle$, or $R^4 |\psi\rangle = |\psi\rangle$. We can split our space into states that are invariant under R , states that transform back to themselves after two rotations, and states that only transform back to themselves after 4 rotations.

For $m = 5$, we need only consider $R |\psi\rangle = |\psi\rangle$ and $R^5 |\psi\rangle = |\psi\rangle$.

Theorem 3.3 (Projecting onto subspaces). *Suppose that we have our initial basis for the full space $|\psi_j\rangle$ with $j = 1, \dots, N$ and a basis $|\psi_i^k\rangle$ with $i = 1, \dots, M$ for the subspace of eigenstates of R with eigenvalue $e^{i\frac{2\pi}{m}k}$. M is significantly smaller than N in general. We would like to find the operator P which projects a general state to our symmetrized basis. The operator R_k is already very close to this, but there are three difficulties. The first is that R_k is not normalized. The second is that R_k contains duplicate columns, so although it projects any arbitrary state to the correct subspace, it is not written in the correct basis. The third is that R_k contains columns that are all zeros.*

These issues are easily remedied. The unique, non-zero columns of R_k are exactly the basis vectors $|\psi_i^k\rangle$. Construct \tilde{R}_k from R_k by removing duplicate columns of R_k . If a given column is duplicated, we keep only the first instance. Thus, \tilde{R}_k is an $N \times M$ matrix. Next, we must normalize \tilde{R}_k . To this end, define a diagonal matrix by

$$N_{ii} = 1 / \sqrt{\sum_j \tilde{R}_{kji} \tilde{R}_{kji}^*} \quad (48)$$

, which is the norm of the i th column of \tilde{R}_k . Then our projection matrix is

$$P_k = N \tilde{R}_k. \quad (49)$$

As usual, the Hamiltonian projected on this subspace is PHP^\dagger .

3.2 General Symmetry Groups

The approach of the previous section can be cast in the language of representation theory. This is a significant step up in mathematical machinery, but it allows one to handle more general symmetry groups, in particular symmetry groups which contain several non-commuting symmetry operations (such as reflections and rotation). We will see that the previous section essentially constructed the irreducible representations of the cyclic group, C_n , and exploited the orthogonality of irreducible representations. (This approach is described here in more detail.)

Definition 3.4 (Representation). A representation is a map from a group G to a set of invertible n -dimensional matrices which preserves the group operation

$$M : G \rightarrow \text{GL}(n, \mathbb{R}) \quad (50)$$

$$M(g_1 g_2) = M(g_1) M(g_2). \quad (51)$$

Here $\text{GL}(n, \mathbb{R})$ is the set of invertible n -dimensional matrices, which is called the *general-linear group*.

Theorem 3.5 (Block Diagonalizing Hamiltonian). *In general, the technique for block diagonalizing a Hamiltonian is to find a set of projector operators, $\{P_i\}$, satisfying the following*

1. $\sum_i P_i = \mathbb{I}$
2. $P_i P_j = \delta_{ij} P_i$
3. $P_i \mathcal{H} P_j^\dagger = 0, i \neq j$

Theorem 3.6 (Projectors from Irreducible Representations). *Suppose that our Hamiltonian has some discrete symmetry group $G = \{g_1, \dots, g_n\}$ with associated operators $M(g_i) = M_i$ that implement each symmetry operation on our Hamiltonian, i.e. $[M_i, \mathcal{H}] = 0$.*

So, for example if we have a square lattice, then this has symmetry group $G = D_4$ which the Dihedral group of index 4, which has 8 elements. $D_4 = \{e, R, R^2, R^3, \sigma_v, R\sigma_v, R^2\sigma_v, R^3\sigma_v\}$, where e is the identity, R represents counterclockwise rotation, and σ_v represents reflection about a vertical line. We can implement the matrices $M(R)$ and $M(\sigma_v)$ using the techniques described above.

Any linear combination of symmetry matrices commutes with the Hamiltonian, and we can create a projector from this operator using the same technique as theorem 3.3. That is, suppose we define symmetry operator

$$S = \sum_{g \in G} c_g M(g) \quad (52)$$

then we get \tilde{S} by zero columns and duplicate columns from S . Finally we normalize \tilde{S} according to

$$N_{ij} = 1 / \sqrt{\sum_j S_{ji} S_{ji}^*} \quad (53)$$

and define the projector

$$P = N \tilde{S}. \quad (54)$$

Now, we would like to construct a class of these projectors satisfying the properties of theorem 3.5. The main challenge is to figure out how many possible orthogonal projectors we can make.

And the answer is provided by representation theory and in particular the concept of irreducible representations of a group.

We can partition a group into a set of conjugacy classes. We say that two elements $g_1, g_2 \in G$ are in the same conjugacy class if and only if there is a third element $h \in G$ such that $hg_1h^{-1} = g_2$. Further, for a given conjugacy class and a given representation (i.e. a choice of matrices $M(g)$ representing our group), we define the character of any element by

$$\chi_g = \text{Tr}(M(g)). \quad (55)$$

From the cyclical property of the trace, we see that for any two elements in the same conjugacy class

$$\chi_{g_2} = \text{Tr}(g_2) = \text{Tr}(hg_1h^{-1}) = \text{Tr}(g_1) = \chi_{g_1}. \quad (56)$$

For the group D_4 that we described above, the conjugacy classes are

$$\{e\}, \{R, R^3\}, \{R^2\}, \{\sigma_v, R^2\sigma_v\}, \{R\sigma_v, R^3\sigma_v\} \quad (57)$$

Although it is not obvious from what I've described here, the set of characters of a group representation are in some sense a complete characterization of that representation. This is why irreducible representations are typically expressed in terms of a character table. This table lists all possible choices of characters for each conjugacy class for all irreducible representations of a given group. Each row represents a single irreducible representation, and each column represents a single conjugacy class.

For example, the character table for D_4 is

$$\begin{array}{c|ccccc} & \{e\} & \{R, R^3\} & \{R^2\} & \{\sigma_v, R^2\sigma_v\} & \{R\sigma_v, R^3\sigma_v\} \\ \hline A_1 & 1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & -1 & 1 \\ B_1 & 1 & -1 & 1 & 1 & -1 \\ B_2 & 1 & -1 & 1 & -1 & 1 \\ E & 2 & 0 & -2 & 0 & 0 \end{array} \quad (58)$$

Now, the different irreducible representations of G have the property that they are orthogonal. Therefore, if define symmetry operators for each irreducible representation $\alpha = A_1, A_2, \dots$ according to

$$S^\alpha = \sum_g \chi_g(\alpha) M^\alpha(g) \quad (59)$$

then we are guaranteed the orthogonality property from theorem 3.5.

In theorem 3.1 we essentially constructed the character table for a cyclic group \mathbb{Z}_n . Restating our findings in this new language, we found that every element was its own conjugacy class, and that the for the k th representation A_k ($k \in \{0, \dots, n-1\}$) the character of the m th element, $g = R^m$, was $\chi_{R^m} = e^{-i\frac{2\pi}{n}km}$.

For a group with both rotational and translational symmetry, the full symmetry group is the semi-direct product of a point group (such as D_4 for this example) and the spatial translation group ($\mathbb{Z}_n \oplus \mathbb{Z}_n$ for an $n \times n$ square lattice). We would write this

$$G = (\mathbb{Z}_n \oplus \mathbb{Z}_n) \rtimes D_4 \quad (60)$$

which for $n = 4$ is a group with 128 elements.

There is an algorithm for calculating character tables called Burnside's Algorithm, which has been implemented in GAP (Groups, Algorithms, Programming - a System for Computational Discrete Algebra).

To diagonalize our Hamiltonian efficiently we are always interested in producing as small diagonal blocks as possible. In some cases, the full symmetry group does not produce the smallest blocks. Indeed, when working with a lattice with D_4 symmetry, the subspace corresponding to the two dimensional representation (labeled E above) is much larger than the others (roughly half the dimension of the full space). Considering only the rotational symmetry, we have the cyclic symmetry group \mathbb{Z}_4 , and this produces four blocks of roughly equal dimension.

4 Fermi-Hubbard

Theorem 4.1. *Suppose we have a hopping Hamiltonian which is the same for each spin state. We will work with the two-spin basis we defined earlier, and the “spinless” fermions basis for each individual spin state,*

$$\begin{aligned}\mathcal{B}_2 &= \{c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{3\downarrow}^\dagger |0\rangle, c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{3\downarrow}^\dagger c_{1\downarrow}^\dagger |0\rangle, \\ &\quad c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{3\downarrow}^\dagger c_{2\downarrow}^\dagger |0\rangle, c_{N\uparrow}^\dagger \dots c_{1\uparrow}^\dagger c_{N\downarrow}^\dagger \dots c_{1\downarrow}^\dagger |0\rangle\} \\ \mathcal{B}_\sigma &= \{|0\rangle, c_{N\sigma}^\dagger |0\rangle, c_{N-1\sigma}^\dagger |0\rangle, c_{N\sigma}^\dagger c_{N-1\sigma}^\dagger |0\rangle, c_{N-3\sigma}^\dagger |0\rangle, c_{N\sigma}^\dagger c_{N-3\sigma}^\dagger |0\rangle, \\ &\quad \dots, c_{N\sigma}^\dagger \dots c_{3\sigma}^\dagger |0\rangle, c_{N\sigma}^\dagger \dots c_{3\sigma}^\dagger c_{1\sigma}^\dagger |0\rangle, c_{N\sigma}^\dagger \dots c_{3\sigma}^\dagger c_{2\sigma}^\dagger |0\rangle, c_{N\sigma}^\dagger \dots c_{1\sigma}^\dagger |0\rangle\}.\end{aligned}$$

If the hopping Hamiltonian for a single spin state written in basis \mathcal{B}_σ is \mathcal{H} , then the full hopping Hamiltonian in the 2-spin basis is

$$\mathcal{H}_{full} = \mathbb{I} \otimes \mathcal{H} + \mathcal{H} \otimes \mathbb{I}, \quad (61)$$

which fully accounts for Fermion statistics as long as \mathcal{H} does.

Similarly, if the rotation operator on the single spin space is R , then the rotation operator on the full space is

$$R_{full} = \mathbb{I} \otimes R \cdot R \otimes \mathbb{I} = R \otimes R. \quad (62)$$

To generate a hopping Hamiltonian on N sites, we need only generate the much smaller Hamiltonian for a single spin state, then take tensor products. Furthermore, since the interaction term is diagonal, we need only generate a single vector.

Both of these facts limits the amount of work required considerably compared with looping over every entry in the Hamiltonian.