Lecture 7: JW Transform
Reading: Check out here. If you look at here be careful that the definitions of occupied and unoccupied orbitals in this paper are switched compared to most other sources and hence everything else is flipped.

1 Overview of solving quantum problems on a digital QC

Regardless of the particular problem or algorithm used on a QC, there are a few steps that we always need to take to make a problem suitable for a QC. They can be summarized as follows:

1. Define what a qubit means in terms of the physical problem. For instance, for a fermionic problem the 0 and 1 of the qubit could correspond to empty and occupied. Or, each qubit could mean something more complicated. We need a transformation from the relevant Hilbert space of the problem to qubit space.

2. With each qubit having a defined meaning, we need to transform operators in the original Hamiltonian to a tensor product of Pauli matrices.

3. For some algorithms, though not all, we need to implement $U = e^{-iHt}$ in an efficient way. Essentially always, we still need to perform unitaries of the form $U = e^{-i\vec{\sigma}t}$ where $\vec{\sigma}$ is a tensor product of Pauli operators arising from the terms in the Hamiltonian.

We quickly run into problems if we try to carry out these tasks. First, a qubit is a two level system, and when many qubits are placed together in a QC they effectively form a collection of spin-1/2 particles that can be operated on and entangled by the gate set of single and two qubit gates. These gates commute and hence act as bosonic operators. If the system of interest consists of hard core bosons (meaning at most one boson can be present per site) it is easy to perform the mapping: qubits simply represent the sites and creation Q^+ and annihilation Q^- operators can be defined in the usual way, e.g. $Q^{+/-} = X \mp iY$, where X and Y are the usual Pauli matrices (note the switch of $\mp$ in the definition of creation and annihilation operators!)

Now let's consider performing this mapping for fermions. One way to map the Fock space to qubits is straightforward since at most fermion can occupy a given state due to the antisymmetry requirement of the wavefunction. Therefore, the occupation number basis maps the occupation number of a given orbital to a given qubit. (There are other ways that we will talk about later). The real problem is that the gate operators on the QC are bosonic and hence commute, but the fermionic operators in an electronic Hamiltonian anti-commute. Somehow we need to convert fermionic operators into Pauli matrices while still retaining the anti-commutation relations. There are various ways to do that, as well as various ways to encode the occupation of electronic states in the qubits.

2 Jordan-Wigner Transform

2.1 Specification of the transform

The Jordan-Wigner transform employs the occupation number basis and specifies how to map fermionic operators a_i, etc to Pauli matrices X, etc. The occupation number basis is defined as the basis for which the qubit encodes whether a spinorbital is occupied or empty (since we have fermions, those are the only choices). Transforming from occupation number of fermions to qubits is easy since qubits are two-level systems. We make the identification:

$$|f_{n-1}...f_1 f_0\rangle \rightarrow |q_{n-1}\rangle \otimes ... \otimes |q_1\rangle |q_0\rangle \tag{2.1}$$

where $q_j \in \{0, 1\}$ is the state of the qubit. We therefore have translated the occupation number of orbital j to the state of the two-level qubit.

That was the easy part. Now, we have to figure out how to map Fermionic creation and annihilation operators to qubit operators, and the problem is that qubit operators commute but fermionic operators anti-commute. To solve this problem, let's first recall the 1-qubit creation and annihilation operators Q^+ and Q^-. They are defined so that $Q^+ |0\rangle = |1\rangle$, $Q^+ |1\rangle = |0\rangle$, $Q^- |1\rangle = |0\rangle$, $Q^- |0\rangle = |0\rangle$. These operators act almost exactly in the way we would like for fermionic operators, and so an obvious strategy to create p-qubit quantum gates is to make a p-tensor product of gates tensored with identity on the other $n - p$ qubits.

The problem is that such an operator would not obey the fermionic anti-commutation relations. Let's investigate the problem further by writing down the matrix representation of Q^+ and Q^-. Define $Q^+ = |1\rangle \langle 0| = (X - iY)/2 = \sigma^-$ and $Q^- = |0\rangle \langle 1| = (X + iY)/2 = \sigma^+$ where X and Y are the usual Pauli matrices. Note the difference in the plus and minus in the operator definition compared to the actual sign in the operator.

Then

$$Q^+ = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{2.2}$$

$$Q^- = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{2.3}$$

You can check that these operators obey the fermionic anti-commutation relations at the same site, e.g. $\{Q^+, Q^-\} = I$ where I is the identity matrix. However, if these operators are part of a tensor product over all sites and the operators are applied to different sites, they will commute and hence their commutator, NOT the anti-commutator, equals 0: $[Q_i^+, Q_j^+] = 0$ for $i \neq j$ and the same for other combinations of + and −. So Q^+ and Q^- are in-between fermionic and bosonic operators.

To define operators that obey all the fermionic requirements, let us first restate the requirements. We have two choices for how to define a canonical order, but once we choose it we should stick to it. Unfortunately, different references make different choices. For the two qubit example below, canonical order is defined as $a_{n-1}^\dagger ... a_0^\dagger |vac\rangle$. With that definition, we have

$$Q_1^+ |n_1 = 0, n_2 = 1\rangle = Q_1^+ Q_2^+ |vac\rangle = -Q_2^+ Q_1^+ |vac\rangle = -|n_2 = 1, n_1 = 1\rangle \tag{2.4}$$

where we have realized that the first ket was not in canonical order. Therefore, we need to add sign counters that keep track of whether states to the right of the state to which the operator is applied are occupied. We actually know of a matrix that can do that sign-counting: the Z Pauli matrix! When applied to a site, this diagonal matrix gives +1 if the state is empty and −1 is filled. So we can solve our problem by letting $Q_1^+ \rightarrow Q_1^+ \otimes I_2$, and $Q_2^+ \rightarrow Z_1 \otimes Q_2^+$. Then,

$$Q_1^+ Q_2^+ = (Q_1^+ \otimes I_2)(Z_1 \otimes Q_2^+) = Q_1^+ Z_1 \otimes Q_2^+ = -Z_1 Q_1^+ \otimes Q_2^+ = -Q_2^+ Q_1^+ \tag{2.5}$$

So tensoring the operator with Z on sites on index less than that of the site of the operator has produced a new operator that satisfies all the fermionic anti-commutation relations.

Now in the Seeley paper, canonical order is actually defined opposite to what I have written above. Specifically, a state is given as:

$$|f_{n-1} ... f_0\rangle = a_0^\dagger ... a_{n-1}^\dagger |vac\rangle \tag{2.6}$$

Since we follow that paper later for the BK transform, we will use it here for the definition of JW-transformed creation and annihilation operators. The transform is:

$$\begin{aligned} a_j^\dagger &\equiv I^{\otimes n-j-1} \otimes Q^+ \otimes Z^{\otimes j} \equiv Q_j^+ \otimes \overrightarrow{Z}_{j-1} && (2.7) \\ a_j &\equiv I^{\otimes n-j-1} \otimes Q^- \otimes Z^{\otimes j} \equiv Q_j^- \otimes \overrightarrow{Z}_{j-1} && (2.8) \end{aligned}$$

where the Z string is defined as $\overrightarrow{Z}_i \equiv Z_i \otimes Z_{i-1}...Z_0$.

With these operators, we have now successfully transformed our fermionic operators and state to Pauli matrices and qubits! Note that the Jordan-Wigner transformation can also be used to exactly solve certain spin Hamiltonians, which was the reason it was originally developed. The one problem with the JW transform is that it turns local operators into non-local operators - a local creation operator on orbital j has now turned into a tensor product of operators that in general has length $O(n)$. Such a non-local Pauli string is undesirable in practice because it leads to large gate depth in the actual quantum algorithm. There are some ways around that, as we will see, but for now we examine exactly how the electronic structure Hamiltonian transforms under the JW transform.

2.2 Application to electronic structure Hamiltonian

2.2.1 One-body terms

The electronic structure Hamiltonian can be divided into 1-body and 2-body terms. Let's first consider the 1-body terms of the form:

$$H_1 = \sum_i h_{ii} a_i^\dagger a_i + \sum_{i<j} h_{ij}(a_i^\dagger a_j + a_j^\dagger a_i) \quad (2.9)$$

Let's start with the diagonal terms. We have:

$$\begin{aligned} a_i^\dagger a_i &= (\prod_{k=1}^{i-1} Z_k) Q_i^+ (\prod_{k=1}^{i-1} Z_k) Q_i^- = Q_i^+ Q_i^- && (2.10) \\ \frac{1}{4}(X - iY)(X + iY) &= \frac{1}{4}(I + iXY - iYX + I) && (2.11) \\ &= \frac{1}{4}(2I + i[X, Y]) = \frac{1}{4}(2I - 2Z) = \frac{1}{2}(I - Z) && (2.12) \end{aligned}$$

To implement the exponentiation of this operator, we have:

$$\begin{aligned} U &= e^{-i/2(I-Z)h_{ii}t} = e^{-iIh_{ii}t/2} e^{iZh_{ii}t/2} && (2.13) \\ &= \begin{pmatrix} 1 & 0 \\ 0 & e^{-ih_{ii}\Delta t} \end{pmatrix} && (2.14) \end{aligned}$$

where in the first line we used the fact that I and Z commute. So this diagonal term is just a phase gate.

Now let's take care of the non-diagonal terms. As an example, let $n = 5$, $i = 2$, $j = 4$ so that

$$\begin{aligned} a_2^\dagger &= Z \otimes Q^+ \otimes I \otimes I \otimes I && (2.15) \\ a_4 &= Z \otimes Z \otimes Z \otimes Q^- \otimes I && (2.16) \\ a_2^\dagger a_4 &= I \otimes Q^+ Z \otimes Z \otimes Q^- \otimes I && (2.17) \end{aligned}$$

A typical string would consist of this term plus its Hermitian conjugate.

With this result as an example, in general, we have the following (leaving off the strings of identity operators):

$$a_i^\dagger a_j + a_j^\dagger a_i \;=\; Q_i^+ Z_i(\prod_{k=i+1}^{j-1} Z_k)Q_j^- + Z_i Q_i^-(\prod_{k=i+1}^{j-1} Z_k)Q_j^+ \tag{2.18}$$

Now in this string we have terms of the form $Q_i^+ Z_i \otimes ... \otimes Q_j^- + Z_i Q_i^- \otimes ... \otimes Q_j^+$. Use the definitions of $Q^+ = (X - iY)/2$, $Q^- = (X + iY)/2$ in the tensor product, add the terms, and use the anti-commutator and commutator properties of Pauli matrices to find the final result:

$$a_i^\dagger a_j + a_j^\dagger a_i \;=\; -\frac{1}{2}(X_i(...)X_j + Y_i(...)Y_j) \tag{2.19}$$

where (...) is shorthand for the Z string. A little shortcut is to realize that $Q^+Z = -Q^+$ and $ZQ^- = -Q^-$ to remove the Z gates prior to evaluating the sum.

So we see that the one-body term can be generically written in the form:

$$H_1 = \sum_i h_{ii}(I_i - Z_i) - \frac{1}{2}\sum_{i<j} h_{ij}(X_i(...)X_i + Y_i(...)Y_i) = \sum_l H_l \tag{2.20}$$

This term consists of a sum of many Pauli strings that are in general nonlocal. We have to exponentiate this sum to implement it as a unitary on our qubits. To deal with the sum of non-commuting operators, we have to Trotterize. Another problem is that we have to somehow implement nonlocal strings of Pauli operators with only two-qubit operations that we can actually implement in a real QC.

There are a few ways to see how to treat these strings. The first, ad-hoc way is as follows. Consider first the unitary:

$$U = e^{-i\theta/2(Z\otimes Z)} = \begin{pmatrix} e^{-i\theta/2} & & & \\ & e^{i\theta/2} & & \\ & & e^{i\theta/2} & \\ & & & e^{-i\theta/2} \end{pmatrix} \tag{2.21}$$

The circuit that implements this unitary can be figured out to be: CNOT, $I \otimes R_z$, CNOT. Here $R_z(\theta) = \exp(-i\theta/2Z)$ is the rotation operator about the Z axis. For instance, checking the operations on the second qubit, we see they are XR_zX which flips the diagonal entries of R_z, giving us the correct two-qubit gate. It's best if you explicitly check this statement yourself.

This result can be generalized to longer Z strings by adding more CNOTs, e.g. for 3 qubits the circuit would be CNOT, CNOT, R_z, CNOT, CNOT.

This is all well and good, but in our Pauli string we have X and Y gates at the beginning and end of the string. How to deal with those? It turns out to be easy: simply change basis first, then apply the circuit we just discussed, then change the basis back. Explicitly, for the $X(...)X$ term, we apply the Hadamard operator on the first and last qubits, then the circuit above, then $H^\dagger = H$ again. For the $Y(...)Y$ term, we apply a rotation gate about the x axis, $R_x(-\pi/2) = \exp(i(\pi/4)X)$ to rotate the y axis to the z axis. We will denote this gate as RY and is given by:

$$RY = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \tag{2.22}$$

To return to the Z basis we apply $RY^\dagger$.

The more mathematical way to get this result is to figure out sequences of unitaries that can be applied to yield the desired string. For this procedure we will need the important property of the Pauli matrices that:

$$e^{i\theta(\hat{n}\cdot\vec{\sigma})} = I\cos\theta + (\hat{n}\cdot\vec{\sigma})\sin\theta \tag{2.23}$$

Now, let's say we want the string $X_1X_2X_3$ where subscript indicates qubit number. Let's start with a Z_1 gate and apply the following unitaries:

$$\begin{aligned} e^{-i(\pi/4)Y_1}Z_1e^{i(\pi/4)Y_1} &= \frac{1}{2}(I-iY_1)Z_1(I+iY_1) && (2.24)\\ &= \frac{1}{2}(Z_1+i[Z_1,Y_1]+Y_1Z_1Y_1) = X_1 && (2.25)\end{aligned}$$

where we used $YZY=-Z$ and $[Z,Y]=-2iX$.

Now apply the following unitaries to X_1:

$$\begin{aligned} e^{-i\pi/4Z_1Z_2}X_1e^{i\pi/4Z_1Z_2} &= \frac{1}{2}(I-iZ_1Z_2)X_1(I+iZ_1Z_2) && (2.26)\\ &= \frac{1}{2}i[X_1,Z_1]Z_2 = Y_1Z_2 && (2.27)\end{aligned}$$

Similarly, we have:

$$\begin{aligned} e^{+i\pi/4Z_1Z_3}Y_1Z_2e^{-i\pi/4Z_1Z_3} &= X_1Z_2Z_3 && (2.28)\\ e^{-i\pi/4Y_3}X_1Z_2Z_3e^{i\pi/4Y_1} &= X_1Z_2X_3 && (2.29)\end{aligned}$$

Note the sign change in the first equation, designed to avoid a sign flip. Therefore, we have

$$X_1Z_2X_3 = U_4^\dagger U_3^\dagger U_2^\dagger U_1^\dagger Z_1U_1U_2U_3U_4 \tag{2.30}$$

Using the fact that $U^\dagger e^{-iZ\theta}U = \exp(iU^\dagger ZU\theta)$, we see that

$$e^{-iX_1Z_2\ldots Z_{j-1}X_j\theta} = \prod_k U_k^\dagger e^{-iZ\theta}\prod_{k'} U_{k'} \tag{2.31}$$

implements the term.

2.2.2 Two-body terms

The two-body terms get quite complicated. We will show a few cases but refer you to the reference for the full details. The general form of the two-body term is:

$$H_2 = \sum_{ijkl} V_{ijkl} a_i^\dagger a_j^\dagger a_l a_k \tag{2.32}$$

Note the order of the indices is different between the coefficient and the operator (physicist notation). Also $i \neq j$ and $k \neq l$ for fermions.

There are many possible cases that can occur. If $i = l$, $j = k$ or $i = k$, $j = l$, the term is diagonal. Let's consider the case for $a_i^\dagger a_j^\dagger a_i a_j$, $i < j$:

$$\begin{aligned} a_i^\dagger a_j^\dagger a_i a_j &= \prod_{s=1}^{i-1} Z_s(Q_i^+) \prod_{t=1}^{j-1} Z_t(Q_j^+) \prod_{t=1}^{j-1} Z_t(Q_j^-) \prod_{s=1}^{i-1} Z_s(Q_i^-) & (2.33) \\ &= \prod_{s=1}^{i-1} Z_s^4 Q_i^+ Z_i Z_i Q_i^- \prod_{t=i+1}^{j-1} Z_t^2 Q_j^+ Q_j^- & (2.34) \\ &= Q_i^+ Q_i^- Q_j^+ Q_j^- = \frac{1}{16}(I - Z_i)(I - Z_j) & (2.35) \end{aligned}$$

Considering that there are four permutations of i and j that yield the same state, the final result is multiplied by 4.

Now let's consider the general case. The general form of the term is

$$H_{ijkl} = \sum_{i<k} \sum_{j<l} V_{ijkl} (a_i^\dagger a_j^\dagger a_k a_l + a_l^\dagger a_k^\dagger a_j a_i) \tag{2.36}$$

where we used that $V_{ijkl} = V_{lkji}$ again due to fact that switching the order of the operators has no effect on the final state. There end up being 12 cases for the ordering of the indices (e.g. $i < j < l < k$, etc) with the second term in the above equation giving 24 possibilities for the ordering. When we switch the ordering of indices, we have to be careful to account for the sign change that accompanies the operators.

It gets to be a pretty horrible mess. The upshot is we get Pauli strings of the form $XZ..ZXYZ...ZY$. These strings look similar to the ones we had for the one-body term but are more precisely two of the strings put together. Using similar logic to the one-body case, we find that we can use two sets of unitary transformations consisting of CNOTs and single-qubit rotations to implement the exponentiation of the string:

$$e^{-iXZ...ZXXZ...ZX\theta} = \prod_s V_s^\dagger \prod_k U_k^\dagger e^{-iZ_a Z_b \theta} \prod_s V_s \prod_k U_k \tag{2.37}$$

where a and b correspond to some of the i, j, k, l depending on the particular permutation.

2.2.3 Complexity

You can see that this calculation ends up with a lot of terms pretty fast. For both the one and two body terms, for each term in the Hamiltonian we need $O(n)$ operations mainly due to the Z strings. The main trouble is that there are $O(n^4)$ terms in H_2 leading to an overall scaling of $O(n^5)$! The gate depth required to handle all of these terms is quite substantial. As a result, using QC for actual electronic structure Hamiltonians of physical interest is quite some time away.