Exploration of Fermion Representations and Hamiltonians on Quantum Computers

Benjamin Schreyer

University of Maryland, College Park, Maryland

May 9th 2024

ABSTRACT

Fermionic particles are one of the two quantizations of particle number, as opposed to bosonic particles. Famous examples of fermions are the electron and proton, and the most famous boson the photon, light, or the carrier of the electromagnetic force. Such particles make up our physical world and as a model have allowed great advance in scientific understanding and technology. Quantum mechanical models of our world, and thus often fermions when chemistry or solid state physics is of concern, suffer from being famously difficult to simulate on classical computers due to the size of a many particle entangled state space. The ability to simulate a system of fermionic particles (electrons) on a quantum computer would be immediately desirable in physical or chemical research of things like semiconductor and superconductor systems. Here the basics of representing such simulations on a quantum computer, Hamiltonian evolution, the machinery that determines the simulation dynamics are explored. New results in more qubit efficient ways to specify our simulated physical system on a quantum computer are discussed.

1. INTRODUCTION

How does a physicist dealing with phenomena on the scale of electrons, atoms, and molecules describe such a system? The mathematical structure is exactly that of unitary operations on quantum states with the phenomenon of collapse, as is also the case for description of quantum computation. The main difference being the labels of our quantum mechanical states. Instead of the abstraction of qubits used in quantum computers, where states are labeled with binary strings:

$$|0110\rangle \equiv |6\rangle, |0001\rangle \equiv |1\rangle ... \tag{1}$$

physicists instead give a quantum state to all mutually exclusive states of a physical system:

$$|Schroedinger's\ cat\ is\ dead\rangle, |Schroedinger's\ cat\ is\ alive\rangle, |The\ electron\ of\ a\ hydrogen\ atom\ is\ energized\rangle ... \tag{2}$$

In probably the most famous case of Schrödinger's cat, Schrödinger gave a quantum amplitude to the life of the cat. In another more serious example figure 1, a physicist may give a photon an amplitude for traveling one of two directions after entering a beam splitter. One may be disappointed in the ingenuity of physicists here, anyone could have told you a model of the world was writing down all possible things. There are of course more concerns, for how does one specify a physical system based on what a scientists see in experiments or observations? How does one predict the effect of external influence on such a system, or even just the passing of time's effect?

Correspondence E-mail: bschrey1@terpmail.umd.edu

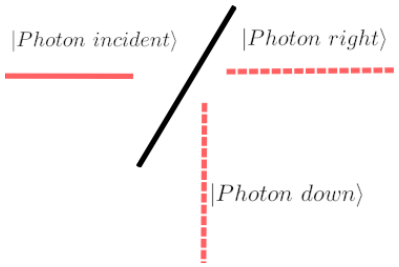

Figure 1. A photons state before and after passing through a beam splitter (solid and dashed respectively).

1.1 Quantum Mechanical Time Evolution

The theory of classical mechanics, or in more friendly terms Newton's laws, have a lot to say about the structure of quantum mechanics. This may come as a surprise, but the overarching idea is that a quantum mechanical model of the world should match up with the classical Newtonian description if one does not make overly precise measurements or predictions. If I throw a baseball, it better follows Newton's laws beyond any perceivable doubt, and this applies just as well to quantum mechanics, the path of the baseball flying through the air, should be equally if not better predicted by quantum mechanics. Classical mechanics says that the total energy determines how a Newtonian system changes in time, specifically its degrees of freedom. In a human scale case, the degrees of freedom of a thrown baseball are its location relative to the pitcher, and its velocity. In this context the total energy is called H after Hamilton who discovered the mathematical formulation of Newton's laws in which the total energy plays the star role. This mathematical formulation leads to the following differential equation after being souped up to handle quantum states, named Schrödinger's equation after is creator. This equations follows the principle that quantum physics maintains agreement with Newton's laws on the large scale. Note that once one has the derivative of our physical state in time, one can find out what the physical state looks like a short time later, and repeat this process to evolve our state any given amount of time into the future.

$$i\hbar\frac{d}{dt}|\psi\rangle = \hat{H}|\psi\rangle \tag{3}$$

This resembles the first order (only involving the first time derivative) formulation of Newtonian mechanics given in (4) which also relates the evolution of a system, in this case a coordinate q (such as the position or momentum) to the energy operator H. The addition of $\hbar$ and i are the main differences, imaginary constant i assures the physical laws to be unitary, which has the classical analog of Newtonian systems being deterministic (having no random outcome). The reason i is not needed in the classical formulation there are two real valued variables the momentum and position, which are replaced in quantum mechanics by a complex scalar composed of two real values.

$$\frac{d}{dt}q = \{q, H\} \equiv \frac{d}{dt}q = -\{H, \_\}q \tag{4}$$

This equation representing Newton's laws in terms of the total energy H is not immediately enlightening without mathematical buildup. This is mainly due to the curly braces in (4) which denote a mathematical structure including a summation and partial derivatives which encode the same mathematical structure that moving to the complex numbers encodes.

Below is the most commonly recognized consequence of Schödinger's equation that shows the first known success of Schrödinger's extension of classical mechanics to quantum mechanical systems. The Planck-Einstein equation for the energy of a photon can be seen as a result of Schrödinger's equation, despite probably also inspiring Schrödinger towards his equation:

$$E = \hbar\omega \tag{5}$$

On the right hand side (RHS) of the equation one sees the photons frequency, and on the left hand side(LHS) the energy of the photon. The frequency of the photon is in proportion to the rate of change of the photon's state. One may show that the photon follows Schrödinger's equation just given the wavelike assumption for light, and the Planck-Einstein equation for photon energy.

$$|photon\rangle = e^{-i\omega t}|photon_0\rangle \tag{6}$$

where $|photon_0\rangle$ is the initial photon's state at time zero. Then multiply by the Planck-Einstein equation to reach

$$E|photon\rangle = \hbar\omega e^{-i\omega t}|photon_0\rangle \tag{7}$$

Now using the simple rule for the time derivative on an exponential find the following expression

$$E|photon\rangle = i\hbar\frac{d}{dt}e^{-i\omega t}|photon_0\rangle \tag{8}$$

$$E|photon\rangle = i\hbar\frac{d}{dt}|photon\rangle \tag{9}$$

$$H|photon\rangle = i\hbar\frac{d}{dt}|photon\rangle \tag{10}$$

Where I have taken the energy scalar E on the photon state to the Hamiltonian operator H which multiplies each specific state vector by its energy, since this is just a specific case where there is only a single energy, the energy of a photon. The Schrödinger equation, applicable beyond just photons, has been applied much further in the hundred years following its inception, leading to the modern age of electronics and computers and many advances in the field of chemistry.

1.2 Quantum States Built from Empty Space

Now that the classically inspired time evolution of some quantum state have introduced, it is further discussed how to describe physics where particles may be added or removed to our system. In a quantum computer, there is the abstraction of qubits, and so all that is necessary is to give a quantum amplitude for each configuration of our qubit register. In the most general physics problems, there is in a sense, a more convoluted set of possible states, although as will be seen these states can be expressed on a quantum computer. The most simple state in the general quantum computer is the so-called "vacuum of space". This state represents a portion of space that is empty of any particles, like atoms, electrons, photons, or cats. It will turn out that in the formalized representation, this state can be thought of as all 0 qubits, or all 1 qubits.

$$|00000000...\rangle \tag{11}$$

$$|Empty\ space/vacuum\rangle \tag{12}$$

The empty space state, the first state considered due to its simplicity, given a representation on a quantum computer.

It is from this state that the physicist can model any nice enough quantum mechanical system (nice enough meaning something they can construct in a laboratory). The most immediate concern for this system is how to introduce some particle, like a cat, electron etc. so that one can see how physics predicts its future. The creatively named creation $a^\dagger$ and annihilation a operators, act on the empty space state, to introduce, or remove particles. For example:

$$a^\dagger_{electron\ at\ position\ x_0}|Empty\ space\rangle = |electron\ at\ position\ x_0\rangle \tag{13}$$

One may recognize the dagger, or conjugation of a linear operator from the mathematics of quantum computing. This is because creating and destroying particles are unsurprisingly inverse operations of each other, and changing our quantum state for space should be unitary (the total probability should be 1). Note now I choose the example of creating an electron of specified energy. The way a physicist or chemist chooses to enumerate the possible states for a quantum system, in terms of energy, position, or other variables, is often a mathematical convenience. In some cases physicists may carefully avoid using one quantity, because of a phenomenon called degeneracy, where two physical states share a descriptive variable such as momentum or energy.

$$a_{electron\ of\ energy\ E_0} a^{\dagger}_{electron\ of\ energy\ E_0} |Empty\ space\rangle = I|Empty\ space\rangle \tag{14}$$

This example demonstrates use of the annihilation operator to remove a particle from a physical state it previously occupied. This should be equivalent to the identity operation, which is denoted I. In addition to building a system, it is also useful to be able to count the number of particles, which in the case of fermions, will take a very simple mathematical form, as to be discussed. This lets a physicist figure out, by simply counting, whether or not a particle reaches some state. In the example of a baseball game, the physicist wants to know whether or not the baseball ever flew into the strike zone, or instead does not count towards fouls on the batter. In real examples a scientist may want to know how many atoms of hydrogen are available in a chemical reaction to act as acidic reagents, counting the number of ionized hydrogen atoms. Counting is also important in specifying the Hamiltonian for a system, which as discussed determines the time evolution of the system. Being able to count particles in certain states allows us to calculate the total energy, or Hamiltonian, roughly:

$$H = \sum_{j \in possible\ states} \#_j \times E_j \tag{15}$$

Where E_j is the energy of the state j. Of which there are $\#_j$ fermions in this state ($\#_j$ will be 1 or 0 for fermions).

1.3 Discrete Physical States

For a quantum computer with bits, one can only label positions, or otherwise enumerated physical states discretely with integers. In the most general case, it is possible to enumerate the states available to a particle with integers, and can apply a creation operator with any integer label defined to realize a particle (in the case of a fermion, usually an electron) in that state. For many cases one might be concerned that with specifying a finite number of possible positions, even though variables like positions have infinite feasible values, there are two usual rhetorical solutions.

First take enough integer labeled states (for example particle at position x_j), such that any x in the true space of continuous positions is immeasurably close to some discrete particle position. Additionally one may consider the system to be trapped in an unimaginably large confinement potential ("force field"), which requires the possible states to become discrete, in analogy with the notes playable on a violin string shown in figure 2 being integer spaced (quantized) frequencies. These frequencies are found by chopping up the set length of the string (analogous to the size of confinement of our system) into an integer number of sections. Another example of confined states, are the orbitals for electrons on atoms, which is one recalls from chemistry are labeled s,p,d etc., these labels may easily be mapped to integers. Note the violin modes (or possible particle states) are all formed by breaking up the space into an integer number of sections, possibly infinite. In the case of a quantum computer, one cannot represent an infinite number of possibilities or physical states. with a finite number of qubits. Some infinity of physical states is cut off to accommodate only a finite number of possible physical states (each being roughly analogous to a violin mode). In the case of a violin string, this is a valid approximation, because humans cannot hear (measure) any note outside of the human audible band, so the listener cant tell the ultra high frequencies are missing. In analogy, the energy of the infinitely labeled physical states of a physical system are usually too energetic to exist, or are so energetic they pass through a detector unmeasured. Having established a labeling for physical states, and a formalism to create and characterize models of physical states, one big question remains. This question is simply put: can two identical particles exist in the same physical

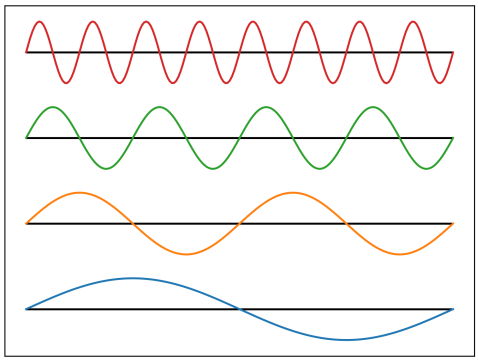

Figure 2. Allowable vibration modes on a violin string, as an analogy for physical states for a confined quantum mechanical wave. Note that although the string may be thought of continuous, the vibrational frequencies of the string are integer spaced. The same transition to discrete description occurs when a quantum state is confined.

state? For example, in the case of bosons, like a photon, I may place many photons into a physical state, if I have a laser pointer, I may make it twice as intense at the point of measurement by simply shining a second one on the detector, causing twice the heating or energy transfer. The bosonic property of photons means I can just keep increasing the brightness of the laser source, by say doubling the number of photons that are in the state of traveling towards the target at the frequency of the laser. As a complete opposite, fermions, cannot be placed in the same physical state, meaning in the same example with multiple lasers, multiple electron beams cannot simply be overlapped to increase their intensity. The electrons must instead be spread over a larger area, or become more energetic, so as to not occupy the same physical states. This phenomena is exactly what gives a chunk of material its hardness. If I try to squeeze a steel block into a smaller size, I will face a lot of resistance from the electrons of the metal, which cannot simply overlap each other and thus let me crush the steel block into a smaller shape. Many other interesting physical phenomena only occur because of this exclusion of electrons from occupying the same state.

Electrons have been mentioned plenty of times, this is because the electron and atoms, molecules, and pseudo particles resulting from electron structure are the foundation of chemistry and solid state physics. When studying chemical reactions, the electrons (which are fermions) change state to form or break chemical bonds in complex chemical reactions. In solid state physics electron dynamics determine the properties of important materials like semi-conductors, super conductors, and more.

2. FERMIONS ON A QUANTUM COMPUTER

Fermions are described as a subset of quantum mechanical operators on the vacuum state. Quantum computers are described by another set of operations on qubit states. The overarching goal of this section will be to show these two collections of operations may be appropriately utilized to represent each other. First the mathematical formalism of Fermions must be addressed, starting from the notation introduced during the introduction.

2.1 Basic Fermion Operational Notation

A system of identical fermions is a collection of physical states defined by the action of two operators on the vacuum state. These operators being the creation operator $a_j^\dagger$ which creates a fermion in the physical state labeled by integer j. The operation to remove a fermion from a state is called annihilation, and has operator a_j for removing a fermion from the state labeled by integer j. Because for most systems, like a bath of chemicals reactions, or a semiconductor crystal conducting electrons, the physicist does not want to worry about the detail of building a crystal or chemical solution out of creation operators, the vacuum state is often replaced with a

different state, such as a neutrally charged crystal lattice (see figure 5), which would approximately occur in a semi-conductor material. For example:

$$a_j^\dagger |\textit{empty crystal lattice}\rangle = |\textit{crystal lattice with electron on site } j\rangle \tag{16}$$

Because the mathematics do not care for the label "crystal lattice" or "vacuum", they are often omitted, and the creation operators are left on their own:

$$a_j^\dagger \tag{17}$$

This may seem confusing at first, but as long as all the operations performed have the properties appropriate to fermion annihilation and creation, one can be sure the physics is correctly modeled. If one ever has a worry about such notation, they are free to right multiply all their equations on the vacuum state of their choice to make sure things are in order. So, as a physicist, the omitted state is the vacuum, for a quantum computer, the omitted state is something much more tangible, the qubit registers. The important part is that our qubit registers encode enough information to differentiate all the different states we desire. Again one could choose to include the basic state in our calculations and that would look like this:

$$a_j^\dagger a_k^\dagger |00000...\rangle \tag{18}$$

This is simply expressing creating two fermions, one at position j, one at position k. Where it has been chosen that the qubit state of all 0s is the vacuum or empty state because it makes the mathematics cleaner. If one so desired $|000100110...\rangle$ could just as well be the ground state, but this would require much uglier notation and circuits when one goes to implement their fermionic operators on a quantum computer.

2.2 Fermion Commutations

Fermion notation and interpretation has been discussed at length. It is now appropriate to introduce the immediate mathematical structure of fermions. The physical assumptions that the vacuum or basic state is the lowest energy state of a fermion system, and that two ore more fermions cannot mutually occupy the same physical state, are beautifully the only things needed to take count of a state of many fermions. The distinctive properties of states and time evolution of the system are all abstracted into the energy assigned to each state, and the sum of these energies, the Hamiltonian H, as previously discussed.

The fundamental algebraic relations between creation and annihilation operators are as follows, again induced by the above assumptions.

$$a_i^\dagger a_j^\dagger + a_j^\dagger a_i^\dagger = 0 \implies a_i^\dagger a_i^\dagger = 0, a_i^\dagger a_j^\dagger = -a_j^\dagger a_i^\dagger \tag{19}$$

$$a_i a_j + a_j a_i = 0 \implies a_i a_i = 0, a_i a_j = -a_j a_i \tag{20}$$

$$a_i a_j^\dagger + a_j^\dagger a_i = \delta_{i,j} \implies a_i a_i^\dagger = I - a_i^\dagger a_i \equiv a_i^\dagger a_i + a_i a_i^\dagger = I \tag{21}$$

(19) Implies the fermion counting statistics, that is applying a single operator to create a fermion in the same physical state is the zero operator, or outside of the space of unitary evolution of the physical system. This equally applies for removing a fermion twice from a physical state, since there is no possible way for two fermions to have occupied the singular state they were removed from. These relations also define the swapping of any two fermions between their physical states as resulting in a negative sign as a relative phase between the two resulting identical state vectors. (21) Expresses the fact that the counting operators for filled and unoccupied fermion states $a_j^\dagger a_j$ and $a_j a_j^\dagger$ respectively are a complete description of counting all possible fermion states. This again indicates there is no fermion occupation of some physical states besides either no occupation, or a single fermion occupation. The counting operators can be seen as have eigenvalue of 0 or 1 on different states, states with an occupied fermion get 1 from the filled state counting operator, and 0 if empty, the reverse is true for the empty state counting operator. The symmetry between these two counting operators can be seen as the most fundamental reason that there exists n and p type semiconductor materials, that is materials where electrons conduct electricity, versus where electron holes (the lack of an electron) conduct electricity.

2.3 Qubits as Physical States?

There is an immediately obvious scheme to try to represent which physical states of our system are occupied by fermions. Qubits are conveniently already of two states 1 and 0. Relabeling the 1 state to "one fermion in this physical state" and 0 to "no fermion in this physical state" is the first thing one would try. Consider now the qubit z-increment operators that under such a scheme become the annihilation and creation operators. Let us investigate if this scheme follows the algebraic relations (19-21). Let $a_?, a_?^\dagger$ be the spin decrement and increment operators, one should check that they follow the prescription for fermions. For some Pauli gate P ($P \in X, Y, Z, I$) it is denoted acting on the $j'th$ qubit as P_j. It is chosen to ignore normalization for convenience.

$$a_{?j}^\dagger = X_j + iY_j \tag{22}$$

$$a_{?j} = X_j - iY_j \tag{23}$$

These are the familiar spin increase/decrease operators if one is familiar from physics. Given that quantum computers are most often formulated on spin qubits, these operators are very easy to implement, in computer science terms, these are hardware level operations. Now check the swap relation (19)

$$(X_i + iY_i)(X_j + iY_j) + (X_j + iY_j)(X_i + iY_i)? = 0 \tag{24}$$

One knows from dealing with quantum circuits that gates acting on only one qubit commute with each other, so one may freely move X_j, Y_j past X_i, Y_i without worry. Then it is found:

$$(X_j + iY_j)(X_i + iY_i) + (X_j + iY_j)(X_i + iY_i) = 2(X_j + iY_j)(X_i + iY_i) \neq 0 \tag{25}$$

This is unacceptable, so while the spin operators do provide two possibilities, and increment and decrement (one would hope these would serve as creation and annihilation), they do not complete the last step of obeying correct phase under swapping of indices. This property is crucial in enforcing the physical fact that two fermions cannot occupy the same state. If the phase had swapped when the spin increment and decrement operators were exchanged, then the swap relation (19) would have been fulfilled as the two summed terms would have canceled to zero.

A Resolution The solution to this phase issue is due to Jordan and Wigner, who early on in the formulation of quantum mechanics discovered the equivalence of fermions and spins (spins are equivalent to the qubits of a quantum computer). Jordan and Wigner formulated the Jordan-Wigner encoding on the spin increment and decrement operators such that they obey the mathematical structure of fermion swapping. On top of the simple almost working idea of (22,23), Jordan and Wigner added operations to correct for the incorrect behavior of spin operations when they are commuted.

$$a_j^\dagger = e^{i\pi \sum_{k=1}^{j-1}(X_j+iY_j)(X_j-iY_j)}(X_j + iY_j) \tag{26}$$

$$a_j = e^{-i\pi \sum_{k=1}^{j-1}(X_j+iY_j)(X_j-iY_j)}(X_j - iY_j) \tag{27}$$

The spin increment and decrement operators still remain on the far right of the creation and annihilation operator. However each qubit's creation and annihilation picks up a phase of $+1$ or -1 depending on how many occupied orbits come before it in the integer enumeration of states, enforced by using counting operators as the phase of a complex exponent. If one swaps two creation or annihilation operators, the number of preceding occupied states the swapped operator will see is one less than previously, which means the phase the combination of the two operators creates be reversed. The expression above can be simplified to not contain the exponential, since the exponent is just $i\pi$ ($e^{i\pi} = -1$) multiplied by some integer (the number of occupied physical states below index j), which is the same as just applying a Z operator to all the qubits below j, if they are occupied they get a negative phase, if not they are undisturbed. (Recall the action of Z on a qubit: $Z(a|0\rangle + b|1\rangle) = a|0\rangle - b|1\rangle$).

$$a_j^\dagger = (\Pi_{k=1}^{j-1} - Z_k)(X_j + iY_j) \tag{28}$$

$$a_j = (\Pi_{k=1}^{j-1} - Z_k)(X_j - iY_j) \tag{29}$$

The reader is encouraged to try the example or comparing $a_2 a_4$ and $a_4 a_2$ (assume that j runs from 1...5) as formulated in (29), to see that indeed the addition of the Z gates remedies the swap problem in this example.

2.4 The Fermion Hamiltonian on a Quantum Computer

In the introduction the quantity or operator of the Hamiltonian (operator when dealing with quantum mechanics) was given as crucial in understanding or simulating a physical system. Here some simple Hamiltonians for simple systems will be given, building up to expressing a Hamiltonian which with simple extensions becomes the basis for some important unsolved problems in physics.

Elementary Hamiltonian: Stationary Fermion Particle in a Magnetic Field Consider a lone electron in an external magnet field imposed by something like a bar magnet. This electron has its own magnetic field, which tends to align with the external magnetic field, this is similar to when a collection of iron fillings magnetize under a bar magnet held up to them, and become aligned with the magnetic field, or the needle of a compass, which is happiest pointing towards the north pole along earth's magnetic field. The easiest way to physically model this system is by completely ignoring the motion of the electron through space, and only think about its intrinsic magnetic field aligning with the external magnetic field. In this formulation, the typical quantum computer already provides the gates needed to express the energy. Without loss of generality assume the external magnetic field points in the Z direction so that one may write the Hamiltonian (energy) as follows:

$$H = \mu B_z Z \tag{30}$$

Where μ is the strength of the intrinsic magnetic field of the electron, and B_z is the strength of the external magnetic field. The Hamiltonian acts on the one qubit states and determines each states energy. Note $|\uparrow_z\rangle$ and $|\downarrow_z\rangle$ represent the possible spin states of an electron's intrinsic magnetic field, which behaves in the same way as a qubit. One may ask where the other directions x and y went. It turns out that the classical degree of freedom of rotating our electrons axis of intrinsic magnetic moment is already accounted for in the coefficients of $|\uparrow_z\rangle$ and $|\downarrow_z\rangle$, a beautiful result that is too complex to discuss now.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \equiv \alpha|\uparrow_z\rangle + \beta|\downarrow_z\rangle \tag{31}$$

Given the Hamiltonian, as discussed in 1.1, it is now a matter of number crunching to calculate how our electrons tiny magnetic field will change as time passes under the influence of the external magnet. It turns out this Hamiltonian is exactly solvable, although it will also be evaluated with approximate methods as follows:

$$|\psi(T)\rangle = (I + \frac{-i}{\hbar} H \Delta t)^{\frac{T}{\Delta t}} |\psi(t=0)\rangle \tag{32}$$

This scheme is employed here in preparation for harder problems, since it is applicable to much harder to solve Hamiltonians to be discussed. The exponent of the operator $(I + \frac{-i}{\hbar} H \Delta t)$ appears because this operator evolves our state a tiny Δt forward in time, so if we repeatedly apply this operator the time coordinate can be increased arbitrarily. As is seen, this problem is nicely formulated using the existing Z gate of the quantum computer, but it is illuminating to express this system using the annihilation and creation operators, which are harder to express but necessary for other highly applicable Hamiltonians. Using the counting operators on two states with labels $\uparrow$ and $\downarrow$ the Hamiltonian is equivalently expressed as follows.

$$H_{alternative} = \mu B_z (a_\uparrow^\dagger a_\uparrow) - \mu B_z (a_\downarrow^\dagger a_\downarrow) \tag{33}$$

The structure of the Z operator is recreated via the counting operators, and the flip of the negative sign on the second term. We can simulate with either of these Hamiltonians and get the same answer.

This Hamiltonian was formulated and solved in Qiskit code the result of a constant magnetic field strength through time is shown in figure 3.

Simulating this system for the case when the external magnetic field is variable in time, the result for one case is shown in figure 4. Specifically the convoluted example (34) as follows is chosen, to illustrate the advantage of solving the system more generally as in (32).

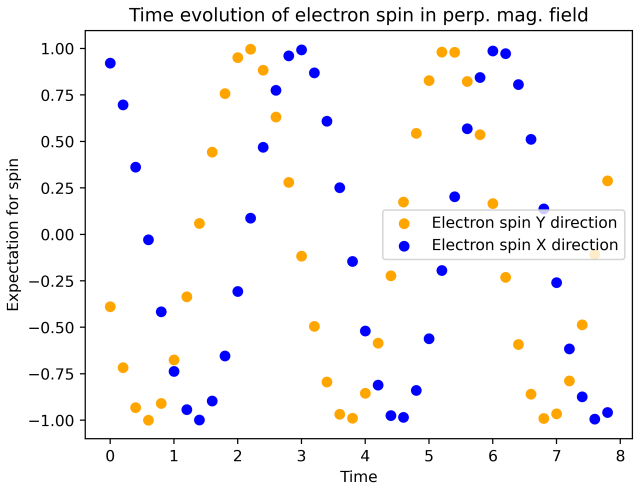

Figure 3. The electron is placed such that its initial intrinsic magnetic moment is in the x direction at time zero. The external magnetic field, as expected form the analytical solution, causes the electron's magnetic field to rotate about the z axis.

$$B_z(t) = (\frac{t}{4})^2 \sinh^{\pi}(t/10) \tag{34}$$

Where units have been ignored, but could easily be corrected by introducing a constant with units of magnetic field strength, and parameterize in terms of t as a fraction of some specific time scale τ.

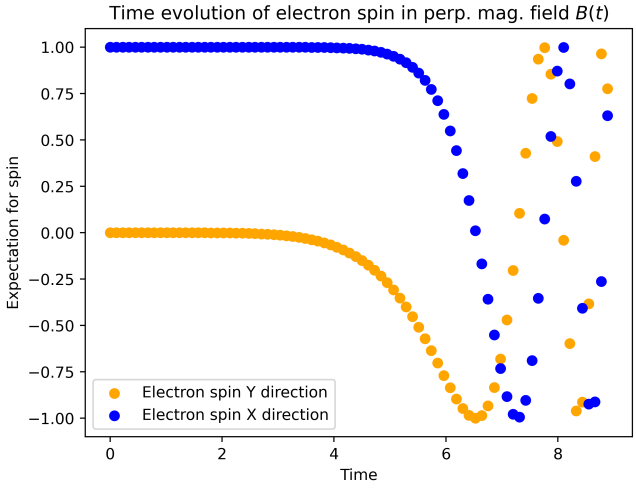

Figure 4. The electron is placed such that its initial intrinsic magnetic moment is in the x direction at time zero. The external magnetic field is now changing, so the rate of evolution of the system changes.

The physical system has been described in it's Hamiltonian, and immediately admits solutions in short notation. As will be shown, actually formulating a Hamiltonian is usually the most complex step, although this Hamiltonian does not even admit any specifically fermionic (impossibility of **two** particles being in the same

physical state) structure because there is only one particle. Of course there is also the caveat of number of qubits needed when crunching the numbers. The system just described is of size a single qubit, so simulating it on a classical computer is easy. The following more general fermion Hamiltonian will express fermionic structure and in cases where it is applied require many qubits.

Nontrivial Hamiltonian: The Hubbard Model The Hubbard model is a quantum mechanical model of a crystal structure. In physics an elementary model of a solid material like a pure metal, semi-conductor, or super conductor is most simply modeled by assuming that the atoms are held firmly in place, and electrons in the valence shell are free to move around the crystal structure. If one is not familiar with the valence shell from chemistry, it is ok to imagine the valence as simply the electrons associated with an atom that are free to move (conduct) while the other electrons are too tightly bound to have any chance of moving throughout the greater material. These valence electrons are the only quantum particles considered in a Hamiltonian formulated for such problems. The nuclei of atoms, and more tightly bound electrons contribute negligibly small dynamical effect so they are considered as a static background that only serve to determine the allowable positions (enumerated creation and annihilation operators) of conducting electrons. Figure 5 shows a sketch of what such a crystal "looks" like in the case of a semi conductor, where the crystal is largely made of one element like silicon, with the addition of a small amount of different atoms (called doping) added, to introduce some free electrons that allow conduction in an otherwise solid crystal. To solve a system without involving computations beyond those done by hand, a physicist will usually assume that the free electrons of a crystal do not interact with each other. The Hubbard model takes the first step out of the approximation of zero interaction between electrons conducting about a crystal. The Hubbard model assumes that conducting electrons interact only with electrons one crystal spacing away from them, and electrons of opposite spin on the same atom of the crystal. Since electrons have an intrinsic magnetic spin, two electrons may be located in space at the same location, so long as they have differing intrinsic magnetic spin, differentiating their physical states in line with the property of fermions of never sharing one physical state between more than one fermion. The Hubbard model assumes two interactions, which are

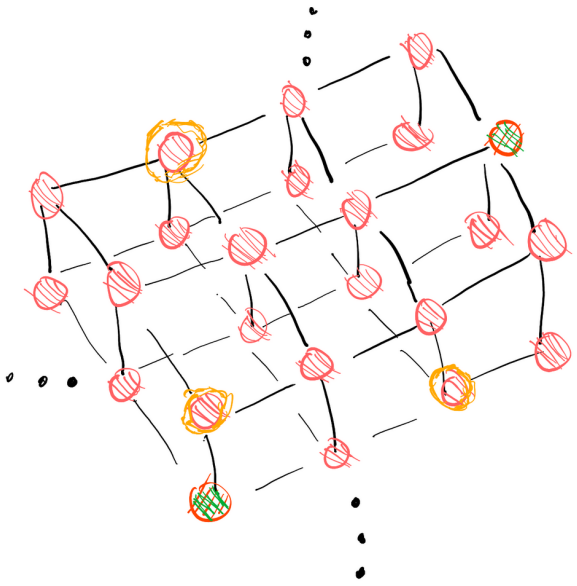

Figure 5. A few electrons on the lattice of a crystal. These electrons interact with each other via close quarters electromagnetic repulsion, and local spin alignment. Yellow coloring shows an electron occupying one position on the crystal lattice, and green coloring shows randomly distributed atoms of the dopant element among the red bulk atoms of a different element, which is the element that contributes electrons that are free to conduct. Note the crystal is always much larger than can be pictured on paper, hence the ellipses continuing the crystal in space.

significant enough to generate entanglement, and move the system into a regime where simulation by a classical

computer can become infeasible.

The Hubbard model we will formulate is 1 dimensional, and has actually been solved analytically. Although adding additional dimensions immediately makes the problem require computational methods. Physicists believe that better simulation of such higher dimensional Hubbard models would be illuminating to many physical and chemical phenomena, most notably super conducting materials.

$$H = -t \sum_{j \times \sigma \in \{\uparrow,\downarrow\}} (a^\dagger_{j,\sigma} a_{j+1,\sigma} + a^\dagger_{j+1,\sigma} a_{j,\sigma}) + U \sum_j a^\dagger_{j,\uparrow} a_{j,\uparrow} a^\dagger_{j,\downarrow} a_{j,\uparrow} \tag{35}$$

This formula has more complexity into two manners, first one has to label states by an index j that denotes the position along the 1 dimensional crystal lattice, and σ which denotes the intrinsic magnetic spin direction of an electron state. There are still a countable number of states, since we can find some integer labeling of all our states. The symbols U and t are just physical constants. The first summation represents interactions of neighboring particles, hence the indexing j, and $j+1$. Cooking up a higher dimensional version of this Hamiltonian (ie an everyday three dimensional crystal, like a chunk of iron) just requires adding more indices than just j, and adding more terms to the first sum representing the more than two neighbors an electron can have if it lives on a higher dimensional lattice (6 closest neighbors in the case of 3 dimensions).

This Hamiltonian is the reason for all the trouble of writing down the Jordan-Wigner encoding. It induces sufficient quantum mechanical complexity that simulation on a classical computer is not feasible, but is a system of great theoretical and practical interest to physicists, chemists, and engineers. We can express this Hamiltonian as a quantum circuit easily, since we have already prepared (29).

This Hamiltonian was expressed in the qiskit language which is shown in the appendix. The simulation of the Hubbard model with this implementation in a simulation of a quantum computer admitted two important properties that verify it is probably correct. The first is that the total probability was conserved under this nontrivial Hamiltonian, indicating a unitary operation as is expected for any time evolution by any physically accurate Hamiltonian that models an isolated state. Secondly the system undergoes oscillation analogues to the oscillations seen in the most simple Hamiltonian of the lone electron in a magnetic field. This is to be expected when the initial conditions are to the Hubbard model are such that there is only one fermions that has no interactions. A plot of such oscillations can be seen in figure 6. On my classical computer the size of this model grows exponentially, and so trying to have even 10 different crystal positions is infeasible, requiring matrices of dimension 2^{20} (2 spins for each of the 10 positions). If the same code was run on a quantum computer, the amount of qubits would only only be 20 perfect computational qubits, much more feasible.

2.5 Optimizations

Even a linear amount of qubits in the number of fermion states available may be considered to be alot to require in the representation of a fermion system. Consider a chunk of crystal that could fit in your hand that is already of order 10^{23} (Avogadro's number) crystal sites electrons could occupy. There is a quite clever way to achieve a logarithmic growth in the number of qubits required ($log(10^{23}) = 23 << 10^{23}$). In many scenarios, like the semiconductor crystal discussed, or systems like a chemical reaction, the number of dynamical electrons (conducting electrons in the case of the crystal) is much smaller than the number of physical states available to the electrons. This means having a qubit dedicated to each possible physical state is massively inefficient, if it is known only a small number like 0.1% or less of these physical states will be occupied. Instead of storing a qubit for each state, instead store a quantum pointer for each electron denoting its physical state. If we have F electrons, and M possible physical states, then the number of qubits required for this scheme will be $O(Flog(M))$, a huge improvement if $F << M$, as is in the case in many problems in solid state physics and chemistry. The reality for such schemes is not all sunshine an rainbows, as extra computations need to be done to wrangle more complicated ways of storing the physical state, while also maintaining the fermion algebra (19, 21...). Carolan and Schaeffer [5] show how to efficiently implement such a scheme on a quantum computer, and even further optimize it with

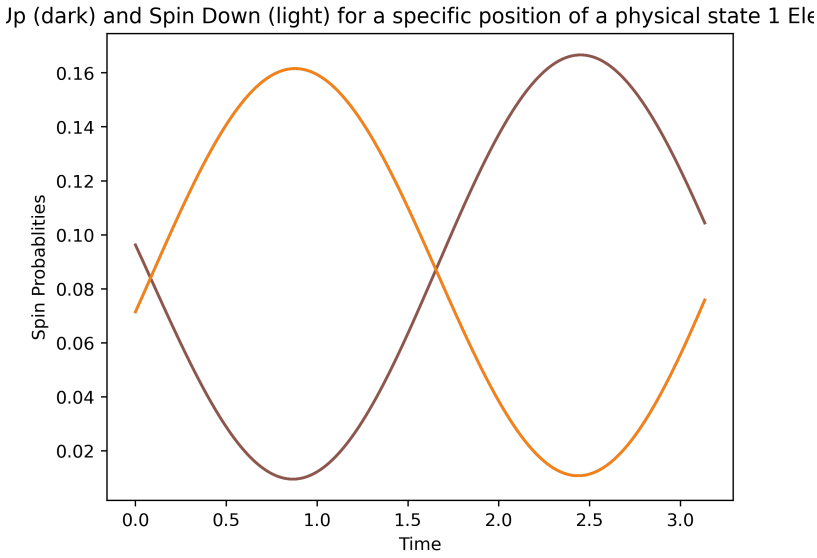

Figure 6. Probability of an electron on some crystal lattice site having spin up or spin down. Note the probabilities do not add up to 1, since these are probabilities for only one crystal site, and the lone electron is spread over many crystal sites which are not plotted.

respect to encoding, and gate operations, to achieve a computational scheme for fermions that is amenable to physics problems having few fermions occupying many physical states. Whitfield et al. [5] also investigated expressing local Hamiltonians more effectively on quantum computers. Local Hamiltonians are those where only states nearby to each other in some euclidean formulation of state space interact, a nontrivial example would be the Hubbard model discussed here, but in 2 or 3 dimensions. This process is extremely nontrivial and is detailed in their paper. There is large trade off in achieving a decreased number of qubits, namely in gate complexity, which is already large because of the Jordan Wigner encoding, and becomes larger when one stores a list of pointers (or more complex data structure) for their fermions. Due to time constraints, I have not implemented their proposed more space efficient fermion representation, however it is well illustrated in their paper. Instead I will talk about the information theoretic basis for their work, and concepts needed to understand it.

Consider a system of F fermions, confined to some amount of possible physical states (also called modes) M. Noting that there will never be two fermions in the same mode, this is just counting how many ways there are to place F objects into M locations, given that the objects are indistinguishable. This is a classic combinatorial problem, and the number of arrangements is $\binom{M}{F}$. If one wishes to represent all this physical states with different combinations of qubits (equivalent in a classical setting to labeling each physical state with a binary number), at least $log(\binom{M}{F})$ qubits will be used, otherwise by the pigeonhole principle, some state will have its representation overlap with a non identical state. Carolan and Schaeffer nearly acheive this information theoretical number of qubits. However applicable to physics problems where there are polynomially less fermions than physical states to occupy (as discussed here) a small inefficency is allowed in the number of qubits needed above the information theoretic lower bound, which allows for better gate complexity of fermion operations. This tradeoff is very useful in the specific field of solid state or chemical simulation, however, there are a wide swath of other encodings of fermions on spin quantum computers that are amenable to other problems, where there are more fermions relative to the number of orbitals, or different Hamiltonian structure.

3. CONCLUSIONS

An introduction to quantum mechanical evolution, and the nature of fermionic particles have been given, to allow motivation and understanding of problems physicists and chemists are concerned with. Then the migration of

such problems of real physical systems onto quantum computers was described in its simplest form,the Jordan-Wigner encoding, and implemented in code. Finally a brief introduction is given to more efficient ways some "sparse" physical systems can be represented on a quantum computer, a rich space for innovation.

4. APPENDIX SELECTED CODE

```python
#Now create elements we will use to create the Jordan-Wigner Fermion operators on the quantum computer
def get_creation_spin_representation_JW(L):
#L is the number of qubits or number of orbitals

    #We are returning a collection of creation and annihalation operators
    ret_plus = []
    ret_minus = []

    for i in range(0, L):
        #"Make the minus Z operators to do phase correction"
        #Note the string multiplication, since we need i z gates
        #Note -1**Z is for the negative 1 mult. on each z gate
        Z_tuples = [("Z" * i, list(range(0,i)), (-1)**i)]


        #Generate the phase correcting Z operators for the ith orbital
        op = SparsePauliOp.from_sparse_list(Z_tuples, num_qubits=L)

        #The normal qubit raising/lower operator elements, we need (X +iY)/2 = Spin Increase
        #Create raising or lowering operators for the ith spin
        plus_op = SparsePauliOp.sum(SparsePauliOp.from_sparse_list([("X",[i],1/2),("Y",[i],1j/2)], num_qubits=L))
        minus_op = SparsePauliOp.sum(SparsePauliOp.from_sparse_list([("X",[i],1/2),("Y",[i],-1j/2)], num_qubits=L))

        #Compose spin raising and lowering and phase corrections to obtain the JW raising and lower fermionic operato
        plus_op = op.compose(plus_op)
        minus_op = op.compose(minus_op)

        #Save the ith raising and lowering
        ret_minus.append(minus_op)
        ret_plus.append(plus_op)
    return ret_minus,ret_plus
```

Figure 7. Code snippet to create fermion annihilation and creation operators from qiskit Pauli gates. These follow the mathematical description in code. SparsePauli op enables efficent formulation of gates made of sums and compositions of Pauli gates.

```python
t = 1 #t is typically positive
U = 1#U is positive for electronic ssystems
#https://en.wikipedia.org/wiki/Hubbard_model
#second quantized form of hubbard hamiltonian
ops_all = []
for i in range(fermion_spatial_orbitals):
    #the end fermion has no right partner interaction
    if i != fermion_spatial_orbitals-1:
        for s in range(2):
            #neighbour coupling
            #need L* (2,) to specify dimension of scalar op
            ops_all.append(minus_JW[2*(i + 1) + s].compose(plus_JW[2*i + s]).compose(ScalarOp(L * (2,),coeff = -t)))
            ops_all.append(minus_JW[2*(i) + s].compose(plus_JW[2*(i+1) + s]).compose(ScalarOp(L * (2,),coeff = -t)))
    #spin pairing potential
    nudown = minus_JW[2*(i) + 1].compose(plus_JW[2*(i) + 1])
    nuup = minus_JW[2*(i) + 0].compose(plus_JW[2*(i) + 0])
    ops_all.append(nudown.compose(nuup).compose(ScalarOp(L * (2,),coeff = U)))
hubbard = SparsePauliOp.sum(ops_all)
```

Figure 8. Code snippet to build a Hubbard Hamiltonian from Jordan-Wigner operators. Again this is a program realizing the mathematical description given.

```
s0  = Statevector.from_label("1000")#
def simulate(s0):
    print(s0)
    #Normalize input state
    s0 = s0 / np.sqrt(np.sum(np.square(np.abs(s0))))
    print(np.sum(np.square(np.abs(s0))),"norm s")
    #Stepping parameters
    t = 1 * np.pi
    steps = 400
    dt = t/steps

    evolution_operator = SparsePauliOp.from_list([("I" * L,1)]).to_matrix()
    print(evolution_operator)
    delt = hubbard.to_matrix() *  dt*(-1.0j)
    evolution_operator += delt

    ts = np.zeros((steps,int(2**L)))
    As = np.zeros((steps,int(2**L))) + 1.0j

    #evolve the state dt each step
    for i in range(steps):
        s0 = np.matmul(evolution_operator,s0)
        ts[i] = i *dt
        As[i] = s0
    return As,ts
simulate(s0)
```

Figure 9. Code to simulate the Hubbard Hamiltonian in qiskit using small time stepping.

5. CITATIONS

Arovas, Daniel P., et al. "The Hubbard Model." Annual Review of Condensed Matter Physics, vol. 13, no. 1, Mar. 2022, pp. 239–74. arXiv.org, https://doi.org/10.1146/annurev-conmatphys-031620-102024.

Carolan, Joseph, et al., "Succinct Fermion Data Structures" (Pending publication) https://quics.umd.edu/events/succinct-fermion-data-structures

Whitfield, James D., et al. "Local Spin Operators for Fermion Simulations." Physical Review A, vol. 94, no. 3, Sept. 2016, p. 030301. arXiv.org, https://doi.org/10.1103/PhysRevA.94.030301.