ADVANCES IN SIMULATING FERMIONS ON A QUANTUM COMPUTER

By

Gabriel Given

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Physics—Doctor of Philosophy

2024

ABSTRACT

One of the principal challenges in simulating fermions on a quantum computer is that qubits lack the anti-symmetry of fermions. The simplest solution, the Jordan-Wigner transformation, converts local interactions into non-local ones. I will describe a method based on Majorana fermions that preserves locality, and propose some improvements to it that reduce the CNOT gate cost and make the algorithm more suited to simulating nuclear matter. I will also suggest how a perturbation theory-based approach can be useful for studies in nuclear physics. Finally, I will discuss contributions I have made involving time fractals and quantum algorithms such as the rodeo algorithm, an eigenvalue estimation algorithm that can obtain precise results even on noisy quantum computers.

ACKNOWLEDGEMENTS

Thank you to everyone at FRIB for being part of a wonderful community to work in. Thank you also to my mom, Hannah, and Chessie for your support over the past several years.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CNOT	Controlled Not Gate
CZ	Controlled Z Gate
NISQ	Noisy Intermediate-Scale Quantum (Computers)

CHAPTER 1

INTRODUCTION

In this chapter, I will introduce nuclear lattice effective field theory, and Hamiltonians used in it. I will also provide some background about circuits used on quantum computers. In the second chapter, I will discuss various algorithms for converting Hamiltonians into quantum circuits. In the third chapter, I will discuss one particular algorithm in depth, including some improvements I have made to it and future applications of it. In the final chapter, I will discuss contributions that I have made to other papers authored by members of Dean Lee's research group.

1.1 Mathematical Preliminaries

Here I will define some mathematical notation that I will use.

The commutator of two operators is defined

$$[A, B] = AB - BA. \tag{1.1}$$

The commutator is zero if the two operators commute.

The anti-commutator of two operators is defined

$$\{A, B\} = AB + BA. \tag{1.2}$$

The anti-commutator is zero if the two operators anti-commute, i.e. $AB = -BA$.

The Kronecker delta function is defined

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{1.3}$$

An operator U acting on a Hilbert space H is unitary if $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the adjoint, i.e. complex conjugate transpose, and I is the identity matrix. Unitary operators have the property that they preserve inner products.

1.2 Quantum Mechanics

In classical mechanics, the time evolution of an object is given by Newton's 2nd law of motion,

$$\mathbf{F} = m\mathbf{a}. \tag{1.4}$$

If we can write down the forces acting on an object, we can solve this differential equation for the position of the object at any future time. Alternatively, we can write down kinetic energy T and potential energy V as functions of some generalized coordinates q_j and their time derivatives (generalized velocities) $\dot{q}_j$. Then we can solve the (mathematically equivalent to Newton's 2nd law) Euler-Lagrange equations,

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}_j} = \frac{\partial \mathcal{L}}{\partial q_j}, \tag{1.5}$$

where $\mathcal{L} = T - V$. Solving either of these differential equations is not always practical. One example where they don't work well is chaotic systems, systems that have exponential dependence on the initial conditions. This means that the initial conditions must be known exactly to get accurate predictions of the future behavior of the system. Additionally, there are situations where classical mechanics does not provide accurate predictions, such as in double-slit interference patterns of electrons.

This is because, to the best of our knowledge, the world obeys quantum mechanics instead of classical mechanics. Classical mechanics remains useful as a limiting case of quantum mechanics for large systems. While classical mechanics is deterministic, quantum mechanics is probabilistic. Instead of solving for position as a function of time, we solve for a complex-valued wave function that represents a probability amplitude, $\Psi(x, t)$. From the wave function we can calculate a probability density,

$$\rho(x) = |\Psi(x, t)|^2, \tag{1.6}$$

which leads to the natural normalization of the wave function

$$\int_{-\infty}^{\infty} |\Psi(x, t)|^2 = 1. \tag{1.7}$$

The state of a system need not be written as a wave function in the position basis; we can write the general state as $|\Psi(t)\rangle$. The dynamics of the state are given by the time-dependent Schrödinger equation,

$$i\frac{d}{dt}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle, \tag{1.8}$$

where $\hat{H} = \hat{T} + \hat{V}$ is the sum of kinetic and potential energy operators. If the dependence of $|\Psi\rangle$ on position and time are independent, we can use separation of variables to obtain the time-independent Schrödinger equation,

$$\hat{H} |\Psi\rangle = E |\Psi\rangle , \tag{1.9}$$

and obtain the time evolution operator,

$$U(t) = e^{-i\hat{H}t}. \tag{1.10}$$

1.3 Variational Principle

Suppose that we have some Hamiltonian H with eigenstates $|n\rangle$. Then we can define the eigenvalues (energies) of the eigenstates as

$$H |n\rangle = E_n |n\rangle . \tag{1.11}$$

If we don't know the ground state, we can still get an upper bound on the ground state energy E_0 using the variational principle. We can write down a state that depends on some parameters θ, $|\psi(\theta)\rangle$. Since the eigenstates form a complete basis, we can decompose $|\psi(\theta)\rangle$ into a linear combination over the eigenstates,

$$|\psi(\theta)\rangle = \sum_n |n\rangle \langle n|\psi(\theta)\rangle = \sum_n c_n |n\rangle . \tag{1.12}$$

Then the expectation value of the Hamiltonian will give an upper bound on the ground state energy:

$$\begin{aligned} \langle\psi(\theta)| H |\psi(\theta)\rangle &= \sum_{nm} c_m^* c_n \langle m| H |n\rangle & (1.13) \\ &= \sum_{nm} c_m^* c_n E_n \delta_{nm} & (1.14) \\ &= \sum_n |c_n|^2 E_n & (1.15) \\ &\geq E_0. & (1.16) \end{aligned}$$

So minimizing the expectation value of the Hamiltonian over θ gives an estimate of the ground state energy. If there exist θ such that $|\psi(\theta)\rangle$ is the ground state, then the ground state energy will be exact.

1.4 Many-Body Wavefunctions

A single particle has a wave function $\psi_i(x_i)$ that depends on the potential that the particle is in. If there are multiple particles, the product

$$\prod_i \psi_i(x_i) \tag{1.17}$$

approximates a solution to the Schrödinger equation [41]. For bosons, this is satisfactory. However, fermions obey the Pauli exclusion principle through wave functions that are antisymmetric under particle exchange. This requires, for example, that

$$\psi_1(x_1)\psi_2(x_2) = -\psi_1(x_2)\psi_2(x_1). \tag{1.18}$$

Slater notes that any permutation of the x_i still results in an approximate solution to the Schrödinger equation. It is possible to construct a unique antisymmetric linear combination of these permutations. This is most conveniently written in terms of a Slater determinant,

$$\frac{1}{\sqrt{N!}}\begin{vmatrix} \psi_1(x_1) & \psi_2(x_1) & \dots & \psi_N(x_1) \\ \psi_1(x_2) & \psi_2(x_2) & \dots & \psi_N(x_2) \\ \vdots & \vdots & & \vdots \\ \psi_1(x_N) & \psi_2(x_N) & \dots & \psi_N(x_N) \end{vmatrix}. \tag{1.19}$$

Permuting two x_i exchanges two columns of the matrix. By the properties of the determinant, this flips its sign. Thus the Slater determinant wave function has the antisymmetry we desire. It is also easy to verify that the Slater determinant satisfies the Pauli exclusion principle: if two particles are in the same state, two rows of the matrix will be identical, causing the determinant to be zero.

It is possible to construct more complex many-body wave functions. For example, the Pfaffian wave function is an antisymmetric product that contains two-particle pairing functions as well as single-particle states [1].

1.5 Second Quantization

Before I introduce nuclear lattice effective field theory, I will introduce the language that its Hamiltonians are written in, second quantization. Second quantization describes many-body systems by the number of particles in each single-particle state. This is very convenient for defining simulations on a lattice.

Suppose that we have some (possibliy infinite) set of orthonormal single-particle states. These states could be, for example, lattice sites or electron orbitals. Then the many-body state can be written as

$$|n_1, n_2, n_3, \ldots\rangle\,, \tag{1.20}$$

where n_i is the number of particles in state i. For bosons (particles with integer spin), n_i can be any nonnegative integer, but for fermions (particles with half-integer spin) n_i is restricted to be 0 or 1 by the Pauli exclusion principle. The total number of particles in the system can be found by summing the occupation numbers of each state, $N = \sum_i n_i$.

We introduce creation and annihilation operators that add or remove one particle from a many-body state. For bosons, we denote the creation operator as $b^\dagger$ and the annihilation operator as b, and they are defined such that

$$b_i^\dagger|\ldots, n_{i-1}, n_i, n_{i+1}, \ldots\rangle = \sqrt{n_i + 1}|\ldots, n_{i-1}, n_i + 1, n_{i+1}, \ldots\rangle \tag{1.21}$$

$$b_i|\ldots, n_{i-1}, n_i, n_{i+1}, \ldots\rangle = \sqrt{n_i}|\ldots, n_{i-1}, n_i - 1, n_{i+1}, \ldots\rangle. \tag{1.22}$$

Note that if we try to annihilate a particle that doesn't exist, the state is replaced by 0. Due to the symmetry of bosonic wavefunctions under particle exchange, the creation and annihilation operators obey the following commutation relations:

$$[b_i^\dagger, b_j^\dagger] = 0, \tag{1.23}$$

$$[b_i, b_j] = 0, \tag{1.24}$$

$$[b_i, b_j^\dagger] = \delta_{ij}. \tag{1.25}$$

Nuclear matter is made of protons and neutrons, both particles with spin $\frac{1}{2}$ and thus fermions. So we will primarily deal with fermionic creation and annihilation operators, denoted $a^\dagger$ and a or $f^\dagger$ and f, respectively. Fermion wavefunctions are antisymmetric under particle exchange, so these fermionic operators obey the anti-commutation relations

$$\{f_i^\dagger, f_j^\dagger\} = 0, \tag{1.26}$$

$$\{f_i, f_j\} = 0, \tag{1.27}$$

$$\{f_i, f_j^\dagger\} = \delta_{ij}. \tag{1.28}$$

The fermionic operators act on many-particle states by

$$f_i^\dagger|n_1, \ldots, n_{i-1}, n_i, n_{i+1}, \ldots\rangle = (-1)^{\sum_{j=1}^{i-1} n_j}(1-n_i))|n_1, \ldots, n_{i-1}, n_i+1, n_{i+1}, \ldots\rangle \tag{1.29}$$

$$f_i|n_1 \ldots, n_{i-1}, n_i, n_{i+1}, \ldots\rangle = (-1)^{\sum_{j=1}^{i-1} n_j} n_i|n_1, \ldots, n_{i-1}, n_i-1, n_{i+1}, \ldots\rangle. \tag{1.30}$$

Note that unphysical application of the creation and annihilation operators, trying to create a second particle in a state in violation of the Pauli exclusion principle or annihilating a particle that doesn't exist, sends the state to 0.

Starting from the vacuum state $|0\rangle$ with no particles, any many body state can be constructed by repeated application of creation operators:

$$\prod_i (a_{p_i}^\dagger)^{n_i} = |n_1, n_2, \ldots\rangle\,. \tag{1.31}$$

One useful operator constructed from creation and annihilation operators is the number operator, $N_i = f_i^\dagger f_i$, which gives the number of particles in state i.

1.6 Lattice Effective Field Theory

Nuclear lattice effective field theory is an established method for simulating nuclear matter [24, 25]. Lattice quantum chromodynamics, our best theory explaining matter, is too computationally expensive for systems of more than a few nucleons. So effective field theories are introduced, which describe interactions between protons and neutrons instead of between quarks. The Hamiltonian for leading order pionless effective field theory is

$$H_{\mathrm{LO}} = H_{\mathrm{free}} + V + V_{I^2} + V^{(3N)}. \tag{1.32}$$

The free part of the Hamiltonian is

$$H_{\mathrm{free}} = \frac{1}{2m}\sum_{\sigma,\tau}\int \mathrm{d}^3\mathbf{r}\,\nabla a_{\sigma\tau}^\dagger(\mathbf{r})\cdot\nabla a_{\sigma\tau}(\mathbf{r}), \tag{1.33}$$

which when discretized for a lattice simulation becomes

$$H_{\text{free}} = \frac{3}{m}\sum_{\mathbf{n},\sigma,\tau} a^{\dagger}_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}) - \frac{1}{2m}\sum_{\mathbf{n},\sigma,\tau}\sum_{l=x,y,z}\left[a^{\dagger}_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}+\mathbf{e}_l) + a^{\dagger}_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}-\mathbf{e}_l)\right]. \quad (1.34)$$

Under periodic boundary conditions, we can rearrange the terms in the second sum to obtain

$$H_{\text{free}} = \frac{3}{m}\sum_{\mathbf{n},\sigma,\tau} a^{\dagger}_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}) - \frac{1}{2m}\sum_{\mathbf{n},\sigma,\tau}\sum_{l=x,y,z}\left[a^{\dagger}_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}+\mathbf{e}_l) + a^{\dagger}_{\sigma\tau}(\mathbf{n}+\mathbf{e}_l)a_{\sigma\tau}(\mathbf{n})\right], \quad (1.35)$$

which makes clear that the second term is a "hopping term" that moves particles between adjacent lattice sites. The potential terms are written in terms of nucleon density and local isospin density,

$$\rho(\mathbf{r}) = \sum_{\sigma,\tau} a^{\dagger}_{\sigma\tau}(\mathbf{r})a_{\sigma\tau}(\mathbf{r}), \quad (1.36)$$

$$\rho_I(\mathbf{r}) = \sum_{\sigma,\tau,\tau'} a^{\dagger}_{\sigma\tau}(\mathbf{r})[\tau_I]_{\tau\tau'}a_{\sigma\tau'}(\mathbf{r}), \quad (1.37)$$

where τ_I are Pauli matrices in isospin space. Using :: to indicate normal ordering, i.e. annihilation operators are to the right of creation operators, we can write

$$V = \frac{C}{2}\int d^3\mathbf{r} : [\rho(\mathbf{r})]^2 :, \quad (1.38)$$

$$V_{I^2} = \frac{C_{I^2}}{2}\int d^3\mathbf{r} : [\rho_I(\mathbf{r})]^2 :, \quad (1.39)$$

$$V^{(3N)} = \frac{D}{6}\int d^3\mathbf{r} : [\rho(\mathbf{r})]^3 : . \quad (1.40)$$

Recently, smeared creation and annihilation operators have been introduced to enforce conservation of orbital angular momentum [25]. The smeared operators also reduce lattice artifacts. The result is that short-range interactions now have a non-local term.

The local portion of the short-range interactions is

$$V_L = \frac{c_L}{2}\sum_{\mathbf{n}} : \rho_L(\mathbf{n})\rho_L(\mathbf{n}) : + \frac{c_{S,L}}{2}\sum_{\mathbf{n},S} : \rho_{S,L}(\mathbf{n})\rho_{S,L}(\mathbf{n}) : \quad (1.41)$$

$$+ \frac{c_{I,L}}{2}\sum_{\mathbf{n},I} : \rho_{I,L}(\mathbf{n})\rho_{I,L}(\mathbf{n}) : + \frac{c_{S,I,L}}{2}\sum_{\mathbf{n},S,I} : \rho_{S,I,L}(\mathbf{n})\rho_{S,I,L}(\mathbf{n}) :, \quad (1.42)$$

and the non-local portion is

$$V_{NL} = \frac{c_{NL}}{2}\sum_{\mathbf{n}} : \rho_{NL}(\mathbf{n})\rho_{NL}(\mathbf{n}) : + \frac{c_{I,NL}}{2}\sum_{\mathbf{n},I} : \rho_{I,NL}(\mathbf{n})\rho_{I,NL}(\mathbf{n}) :, \quad (1.43)$$

where : : indicates normal ordering. The interactions are written using smeared density operators defined as follows:

$$\rho_{NL}(\mathbf{n}) = a^{\dagger}_{NL}(\mathbf{n})a_{NL}(\mathbf{n}), \tag{1.44}$$

$$\rho_{I,NL}(\mathbf{n}) = a^{\dagger}_{NL}(\mathbf{n})[\tau_I]a_{NL}(\mathbf{n}), \tag{1.45}$$

$$\rho_{L}(\mathbf{n}) = a^{\dagger}(\mathbf{n})a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n'n} \rangle} a^{\dagger}(\mathbf{n'})a(\mathbf{n'}), \tag{1.46}$$

$$\rho_{S,L}(\mathbf{n}) = a^{\dagger}(\mathbf{n})[\sigma_S]a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n'n} \rangle} a^{\dagger}(\mathbf{n'})[\sigma_S]a(\mathbf{n'}), \tag{1.47}$$

$$\rho_{I,L}(\mathbf{n}) = a^{\dagger}(\mathbf{n})[\tau_I]a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n'n} \rangle} a^{\dagger}(\mathbf{n'})[\tau_I]a(\mathbf{n'}), \tag{1.48}$$

$$\rho_{S,I,L}(\mathbf{n}) = a^{\dagger}(\mathbf{n})[\sigma_S \otimes \tau_I]a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n'n} \rangle} a^{\dagger}(\mathbf{n'})[\sigma_S \otimes \tau_I]a(\mathbf{n'}), \tag{1.49}$$

where the sum over $\langle \mathbf{n'n} \rangle$ is over nearest neighbors on the lattice. We use σ_S to represent the spin Pauli matrices and τ_I to represent the isospin Pauli matrices. The smeared creation and annihilation operators are

$$a_{NL}(\mathbf{n}) = a(\mathbf{n}) + s_{NL} \sum_{\langle \mathbf{n'n} \rangle} a(\mathbf{n'}), \tag{1.50}$$

$$a^{\dagger}_{NL}(\mathbf{n}) = a^{\dagger}(\mathbf{n}) + s_{NL} \sum_{\langle \mathbf{n'n} \rangle} a^{\dagger}(\mathbf{n'}). \tag{1.51}$$

1.7 Quantum Circuits

In 1982, Richard Feynman proposed that suitable devices could by created to simulate any quantum system: quantum computers [19]. There are certain problems where it is believed that quantum computers can outperform classical computers. One famous example is the factorization of large numbers. RSA encryption relies on the inability of classical computers to find the prime factors of large numbers. However, in 1994, Shor proposed an algorithm that would allow a sufficiently large quantum computer to find the prime factorization of a large number [40]. Fortunately for the security of the internet, quantum computers are not yet capable of running Shor's algorithm on large numbers, as we are still in the Noisy Intermediate-Scale Quantum (NISQ) era [34]. Two-qubit gates typically have error rates of order 0.1%. There is also an error rate when taking a measurement of a quantum circuit, of order 1% for superconducting qubits. Intermediate-Scale refers to that most quantum computers have no more than a few hundred qubits. IBM has recently introduced a quantum computer with over 1000 qubits [8], but error rates continue to be one of the primary challenges in quantum computing. I wish to determine how to use this rapidly-improving technology for simulations of nuclear lattice effective field theory.

Qubits store data in a quantum two-level system, with states $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. The state of a qubit can be written as a superposition of those two states,

$$a|0\rangle + b|1\rangle, \tag{1.52}$$

where a and b are complex numbers with $|a|^2 + |b|^2 = 1$. Qubits can store much more information than classical bits: it takes 2^N classical bits to store the same amount of information as N qubits.

Analogously to logic gates on classical computers, there are one- and two-qubit quantum gates. The three Pauli matrices,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{1.53}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{1.54}$$

and

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1.55}$$

are used as single-qubit gates. The Hadamard gate creates a superposition of the $|0\rangle$ and $|1\rangle$ states,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.56}$$

The phase shift gate leaves the $|0\rangle$ state unchanged and applies a phase $e^{i\varphi}$ to the $|1\rangle$ state,

$$P(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}. \tag{1.57}$$

Two important special cases of the phase gate are the S gate,

$$S = P(\pi/2) = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \tag{1.58}$$

and the T gate,

$$T = P(\pi/4) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{1.59}$$

I will also use a rotation about the z-axis,

$$R_z(\theta) = e^{-iZ\theta/2} = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \tag{1.60}$$

Gates for rotations about the x and y axes are defined similarly,

$$R_x(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{1.61}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{1.62}$$

In classical computation, the NAND gate is universal, i.e. any circuit can be written using only NAND gates. For quantum computation, universality requires a set of gates. One such universal set is $\{H, T, CNOT\}$ [4].

An important class of two-qubit gates is controlled 1-qubit gates. If the control qubit (indicated by a dot) is in the $|1\rangle$ state, a single qubit gate is applied to the target qubit. Otherwise, the identity is applied. As the Pauli X gate acts as a not gate (flipping $|1\rangle$ to $|0\rangle$ and vice versa), the controlled X gate is also called a controlled not gate, CNOT. The controlled Z gate, CZ, has the special characteristic that it is symmetric in the control and target qubits. The circuit symbols for the CNOT and CZ gates are shown in Fig. 1.1.

It is often useful (or necessary) to decompose two qubit gates into circuits composed of single qubit gates and a certain subset of two-qubit gates, such as only CNOT gates. This can be done by the transpilation tools in Qiskit [36]. A controlled R_z gate is equivalent to the circuit shown in Fig. 1.2, composed of R_z gates and CNOT gates.

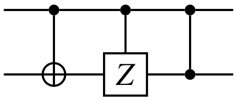


Figure 1.1 From left to right, a CNOT gate, CZ gate, and alternate method of displaying a CZ gate. The upper qubit is the control qubit.

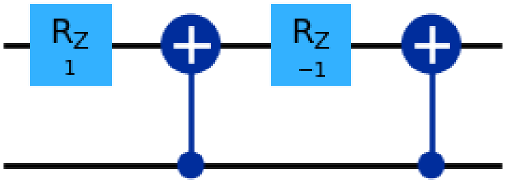


Figure 1.2 Decomposition of controlled $R_z(2)$ gate into R_z and CNOT gates. The lower qubit is the control qubit.

This circuit is an example of the utility of controlled reversal gates [3]. Let R be a product of single-qubit gates that anti-commutes with a Hamiltonian H. Then $RHR^\dagger = -H$ follows by right-multiplication by $R^\dagger$. Thus a controlled R gate, denoted C_R, flips the sign of time evolution (exponentiation) of H. In the circuit in Fig. 1.2, $R = X$ and $H = Z$. From its definition, the $R_z(\theta)$ gate is the time evolution of Z over time $\theta/2$. When the control qubit is on, both R_z gates contribute a rotation over time 1; when the control qubit is off, the two R_z gates cancel out, making the circuit into the identity. Thus the circuit creates the desired controlled $R_z(2)$.

One other useful circuit is to exponentiate a Pauli string [38]. Fig. 1.3 shows the exponentiation of YXZ. Single-qubit change of basis gates are applied to each qubit so that a single R_z can rotate all of the qubits at once. We then return each qubit to the computational basis. This circuit can be controlled by simply controlling the R_z gate: if the R_z gate is turned off, the left and right sides of the circuit cancel each other.

1.7.1 Quantum Algorithms

One old algorithm for using quantum computers to find eigenvalues is Quantum Phase Estimation [15]. Suppose that there is some unitary operator U with eigenvalue $e^{i2\pi\phi}$, and that we know how to implement controlled U^{2^k} for $k = 0, 1, 2, \ldots$. The jth auxiliary qubit is used to control

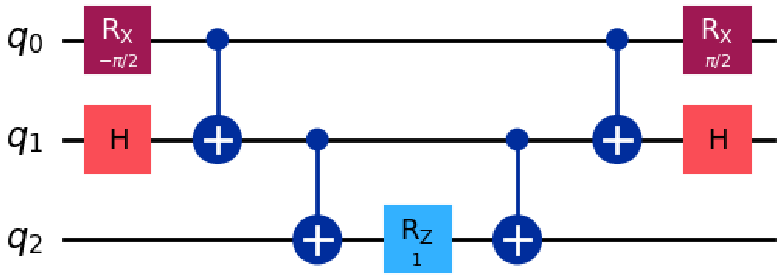


Figure 1.3 Circuit for $e^{-iY^{(0)}X^{(1)}Z^{(2)}/2}$. This circuit can be controlled by controlling the R_z gate.

U^{2^j}, so it ends up in the state

$$|0\rangle + e^{i2\pi(2^j\phi)}\,|1\rangle\,. \tag{1.63}$$

Then an inverse Quantum Fourier Transform can extract the best m-bit approximation of ϕ with a high probability, where m is the number of auxiliary qubits. This algorithm is not fault tolerant, so it is not practical to implement on NISQ-era devices.

In 2014, a new eigensolver was introduced based on the variational principle: the Variational Quantum Eigensolver [32]. This is a hybrid quantum-classical algorithm. First, the Hamiltonian is written as a sum of terms such that the expectation value of each term can be calculated efficiently on a quantum computer (e.g. by writing the Hamiltonian as a sum of Pauli strings). Then, by linearity, the expectation value of the Hamiltonian is the sum of expectation values for each term. An initial state with variational parameters is chosen, and the expectation value of the Hamiltonian is calculated. The ground state will minimize the expectation value. The classical computer is then used to perform the minimization and choose the next set of parameters for the initial state. Compared to Quantum Phase Estimation, the Variational Quantum Eigensolver requires a larger number of smaller circuits. This reduces the time it takes to run each circuit, making it more practical given the short decoherence times of NISQ era hardware. It has been proposed that, for large systems that can be decomposed into subsystems and interaction terms, it is possible to use variational quantum eigensolvers to construct ground states of each subsystem, and then a second variational quantum eigensolver to find the ground state of the entire system [20].

In 2012, a method was proposed to implement a linear combination of unitary operators on a quantum computer [12]. The group of unitary operators is not closed under addition, so a linear combination of unitaries cannot in general be implemented exactly on a quantum computer. However, it is possible to implement the linear combination $\kappa U_1 + U_2$ (up to normaliation) with an error rate at most $4\kappa/(\kappa + 1)^2$. Iterating this method allows any general linear combination of unitaries to be implemented.

The quantum approximation optimization algorithm is another method for finding eigenvalues [18, 23]. It involves two Hamiltonians: H_P, the Hamiltonian that we desire to find the eigenvalues of, and H_M that is easy to solve. The algorithm takes advantage of the adiabatic theorem to start with an eigenvector of H_M and end up with a good approximation of an eigenvector H_P. Time evolutions of the two Hamiltonians are applied alternately, according to a set of parameters that are optimized classically.

1.7.2 Quantum Hardware

The algorithms I work with are hardware-independent. However it is still useful to understand some of the technologies used to build quantum computers.

IBM is developing quantum computers based on superconducting qubits [43]. An *LC* circuit, formed of an inductor and a capacitor, creates a harmonic oscillator potential. This has evenly-spaced energy levels. An *LC* circuit therefore doesn't work for a qubit, as there is no way to distinguish transitions between the ground state and the first excited state from higher-level transitions. So an anharmonicity is added to the circuit in the form of a Josephson junction, a very thin (insulator width of order 2 nm) superconductor-insulator-superconductor circuit element. This makes the transition energies different, so that the two-state system of the ground state and first excited state can be isolated. Low-loss cavity resonators are used to mediate interactions between qubits. The Josephson junction circuits and cavity resonators can be manufactured in large quantities, allowing large quantum computers to be constructed.

One of the leading alternatives to superconducting quantum computers is trapped ion quantum computers, such as the Quantinuum H2 series [16]. The H2 computers use hyperfine states of ytterbium. The ions, paired with barium coolant ions, are stored in a race track-shaped surface-electrode trap. It is possible to rearrange the order of the ions on the trap, and two-qubit gates can be applied to adjacent pairs of ions. This allows two-qubit gates to be applied to any pair of qubits, unlike for superconducting qubits, where two-qubit gates can only be applied to qubits connected by a resonator.

CHAPTER 2

FERMION TO QUBIT MAPPINGS

In lattice effective field theory, we have a lattice of sites where the fermions can be located. We also have a Hamiltonian written in terms of fermionic creation and annihilation operators $f^\dagger$ and f. Our goal now is to implement time evolution of this Hamiltonian on a quantum computer. We first rewrite the Hamiltonian as a sum of Pauli strings acting on qubits, $H = \sum_j h_j$ where h_i are Pauli strings. Then we can use the first order Lie-Trotter formula [13],

$$e^{-iHt} \approx \prod_j e^{-ih_jt}, \tag{2.1}$$

along with the circuit from Fig. 1.3 to exponentiate the Pauli strings to create a quantum circuit to represent our Hamiltonian. The first order Lie-Trotter formula has error $O(t^2)$. Higher order Lie-Trotter formulas have errors that scale with higher powers of t. I will primarily consider lattice effective theory in two dimensions.

2.1 Jordan-Wigner Transformation

The simplest method to rewrite Hamiltonians in terms of Pauli strings is the Jordan-Wigner transformation [49, 38]. We use a lattice of the same size as the physical lattice, and use a one-to-one mapping between the fermion sites and qubits. We let the qubit states represent the occupation of the corresponding lattice sites: $|1\rangle_i$ means that there is a fermion on site i and $|0\rangle_i$ means that there is no fermion on site i. The fermionic creation operator satisfies

$$f_j^\dagger |f_1 \dots f_{j-1} 0 f_{j+1} \dots f_n\rangle = (-1)^{\sum_{i=0}^{j-1} f_i} |f_1 \dots f_{j-1} 1 f_{j+1} \dots f_n\rangle \tag{2.2}$$

and

$$f_j^\dagger |f_1 \dots f_{j-1} 1 f_{j+1} \dots f_n\rangle = 0, \tag{2.3}$$

where f_i is the number of fermions on lattice site i [5]. Note that Eq. 2.2 has a sign determined by parity; this is due to the anti-symmetry of fermion wave functions. Equation 2.3 enforces the Pauli exclusion principle by forbidding a second fermion from being added to a particular lattice site.

We want to find an equivalent operator written using Pauli matrices in order to construct a quantum circuit. A reasonable guess is

$$\hat{Q}_j^+ = |1\rangle\langle 0| = \frac{1}{2}(X_j - iY_j) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}_j , \tag{2.4}$$

$$\hat{Q}_j^- = |0\rangle\langle 1| = \frac{1}{2}(X_j + iY_j) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}_j . \tag{2.5}$$

However these operators do not reproduce the parity term of Eq. 2.2 required due to the anti-commutation of fermions. To make acceptable qubit creation and annihilation operators, we must restore the anti-commutation relation of fermionic creation and annihilation operators,

$$\left\{a_j, a_k\right\} = 0. \tag{2.6}$$

We note that $\left\{\hat{Q}_j^\pm, Z_j\right\} = 0$, so appending a string of Z operators will restore anti-commutation,

$$f_j^\dagger \rightarrow \bigotimes_{k=1}^{j-1} Z_k \otimes \hat{Q}_j^+, \tag{2.7}$$

$$f_j \rightarrow \bigotimes_{k=1}^{j-1} Z_k \otimes \hat{Q}_j^-. \tag{2.8}$$

The downside to the Jordan-Wigner transformation (in more than one dimension) is that the creation and annihilation operators are non-local, and are Pauli strings with length that scales with the system size.

In two (or more) dimensions, the lattice sites need to be numbered in some order for the operators to be well-defined. For a hopping term on an L by L lattice, with the qubits numbered left to right and top to bottom, the horizontal hop will be efficient, $O(1)$, as the two sites have consecutive indices. However the vertical hop will have a Pauli Z string of length $O(L)$ due to the gap in indices. In three dimensions, hops between layers of qubits would have Z strings of length $O(L^2)$.

2.2 Parity Basis

Since the computational cost of the Jordan-Wigner transformation is due to calculating parity, one might suspect that it would be more efficient to store parity, $p_j = \sum_{k=1}^{j} f_k$, rather than occupation in the qubits [38].

This complicates the qubit creation and annihilation operators,

$$\begin{aligned} \hat{P}_j^{\pm} &= |0\rangle\langle 0|_{j-1} \otimes \hat{Q}_j^{\pm} - |1\rangle\langle 1|_{j-1} \otimes \hat{Q}_j^{\mp} \qquad (2.9) \\ &= \frac{1}{2}(Z_{j-1} \otimes X_j \mp iY_j), \qquad (2.10) \end{aligned}$$

as we need to use $\hat{Q}_j^-$ to create a particle if the parity of site j is already 1. But we will also need to update the parity of all sites with indices greater than j with Pauli X matrices. So we have not solved the problem of the Pauli Z string, but merely moved it to a Pauli X string on different qubits.

For a hopping term on a L by L lattice, the horizontal hop will be efficient, $O(1)$, and the vertical hop will have a long Pauli Z string, $O(L)$, the same as for the Jordan-Wigner transformation.

2.3 General Linear Transformation Mappings

In two dimensions, the Jordan-Wigner transformation and parity basis methods are two examples of a general class of linear transformation-based methods, described in Ref. [44]. One particularly interesting case is a compromise between the occupation and parity bases that reduces the length of the Pauli strings: the Bravyi-Kitaev method gives strings of length $O(\log L)$ instead of $O(L)$, where L is the size of the lattice [5, 38].

To construct the general linear transformation mapping, let $n = L^2$ be the total number of fermion sites. Given an invertible $n \times n$ matrix A with elements in $\mathbb{F}_2$, we define a fermion to qubit transformation

$$\boldsymbol{\omega} = A\boldsymbol{f}, \quad \boldsymbol{f} = A^{-1}\boldsymbol{\omega}, \qquad (2.11)$$

where $\boldsymbol{f} = (f_1\ f_2\ \ldots\ f_n)$ is the vector of fermion mode occupation numbers, $\boldsymbol{\omega} \in \mathbb{Z}_2^{\otimes n}$ is the vector of qubit states, and all math is done in $\mathbb{F}_2$. For the Jordan-Wigner transformation, A is the identity matrix. For the parity transformation, A is an upper-triangular matrix of 1s,

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.12}$$

We will define several sets of qubits that will be convenient for writing the creation and annihilation operators. First are the update sets $U(j)$, which is the set of qubits that must be updated when the occupation of qubit j changes. $U(j)$ is the set of qubits with nonzero entries in the jth column of A. Notice that for the parity basis the update set $U(j)$ is all qubits with indices $\geq j$, and for the Jordan-Wigner transformation $U(j)$ contains only j. Similarly to the parity basis method, we will need to apply Pauli X operators to qubits in the update set.

To satisfy the anti-commutation of fermions, we need to calculate parity, $p_j = \sum_{i=0}^{j-1} f_i$. We define the parity set $P(j)$ as the set of qubits needed to calculate parity, which requires

$$\sum_{i \in P(J)} \omega_i = \sum_{i=0}^{j-1} f_i, \tag{2.13}$$

If we define R as an upper-triangular matrix of 1s with zeros on the diagonal, $P(j)$ contains the column indices of nonzero entries in row j of RA^{-1}. We will need to apply Pauli Z operators to qubits in the parity set. The third useful set is the flip set $F(j)$, which contains the set of qubits necessary to determine the occupation of qubit j. This allows us to determine whether Q^+ or Q^- is necessary. $F(j)$ contains the nonzero entries in row j of A^{-1}.

We can then use all three sets to find qubit representations of the fermionic creation and annihilation operators:

$$f_j^\dagger \rightarrow \frac{1}{2}\left(\bigotimes_{k \in U(j)} X_k\right)\left(\mathbb{I} + \bigotimes_{l \in F(j)} Z_l\right)\left(\bigotimes_{m \in P(j)} Z_m\right) \tag{2.14}$$

$$f_j \rightarrow \frac{1}{2}\left(\bigotimes_{k \in U(j)} X_k\right)\left(\mathbb{I} - \bigotimes_{l \in F(j)} Z_l\right)\left(\bigotimes_{m \in P(j)} Z_m\right). \tag{2.15}$$

2.3.1 Bravyi-Kitaev Transformation

The Bravyi-Kitaev transformation is a sort of compromise between the Jordan-Wigner and Parity bases [5, 38, 45]. The transformation matrix A_{BK} has a binary tree structure. It can be recursively defined by

$$A_{BK,1} = (1) \tag{2.16}$$

and

$$A_{BK,2^{x+1}} = \left(\begin{array}{c|c} A_{BK,2^x} & 0 \\ \hline 0 & \\ & A_{BK,2^x} \\ \leftarrow 1 \rightarrow & \end{array}\right), \tag{2.17}$$

where ← 1 → represents a row of ones. Recursive formulae for the update, parity, and flip sets are given in Ref. [45]. The parity and update sets have Pauli weight (string length) $O(\log L)$, and hence the hopping terms also have weight $O(\log L)$. However, for large systems, it is not possible to embed the Bravyi-Kitaev transformation into a square lattice of qubits [44].

2.4 Auxiliary Qubit Mappings

Auxiliary Qubit Mappings are an improvement on the Jordan-Wigner transformation that use auxiliary qubits to store the parity of sets of lattice qubits. These auxiliary qubits essentially provide shortcuts, avoiding the long Pauli strings of the Jordan-Wigner transformation [44]. It's not clear to me how much the need to update the auxiliary qubits offsets this advantage. Ref. [44] claims a Pauli string length of $O(1)$.

2.5 Majorana Fermion Methods

Several methods based on Majorana fermions also give constant scaling of Pauli string length with the lattice size. Mathematically, a Dirac fermion f can be split into two Majorana fermions γ that are their own antiparticles:

$$\gamma_{2j-1} = f_j + f_j^\dagger \tag{2.18}$$

$$\gamma_{2j} = -i\left(f_j - f_j^\dagger\right), \tag{2.19}$$

where

$$\gamma_j \gamma_j = 1 \text{ and } \gamma_j = \gamma_j^\dagger. \tag{2.20}$$

The Majorana fermions satisfy the anti-commutation relation

$$\left\{\gamma_j, \gamma_k\right\} = 2\delta_{jk}. \tag{2.21}$$

The $\eta\chi$ method that I use is a Majorana fermion method. First, I will describe a couple others.

2.5.1 Bravyi-Kitaev Superfast

Several methods put qubits on the edges of a graph. Bravyi-Kitaev Superfast is one of them [5, 44]. The vertices of the graph represent the fermionic modes, and the edges represent interaction terms in the Hamiltonian. For example, if the term $f_i^\dagger f_j$ appears in the Hamiltonian, then $(i, j) \in E$, the set of edges.

We use the Majorana fermions to define edge and vertex operators,

$$A_{jk} = -i\gamma_{2j-1}\gamma_{2k-1} \quad \text{(edge)} \tag{2.22}$$

$$B_j = -i\gamma_{2j-1}\gamma_{2j} \quad \text{(vertex)}. \tag{2.23}$$

These operators can generate any parity-preserving fermionic operator. For example, a hopping term can be written as

$$f_j^\dagger f_k + f_k^\dagger f_j = \frac{i}{2}(A_{jk}B_k + B_j A_{jk}). \tag{2.24}$$

We choose an (arbitrary) orientation $\epsilon_{jk} = \pm 1$ for every edge. For each vertex k, we order the incident edges (j, k), denoted by the symbol $\underset{k}{<}$. This allows us to define the edge and vertex operators:

$$B_k = \bigotimes_{a:(a,k)\in E} Z_{ak}, \tag{2.25}$$

$$A_{jk} = \epsilon_{jk} X_{jk} \left(\bigotimes_{b:(b,k)\underset{k}{<}(j,k)} Z_{bk} \right) \left(\bigotimes_{c:(j,c)\underset{j}{<}(j,k)} Z_{jc} \right). \tag{2.26}$$

We observe that the maximum Pauli weight of a hopping term is a constant, independent of number of particles or lattice size. To prevent the Majoranas from interacting with themselves, it is necessary to apply a loop constraint. For any closed length-l loop in the graph, $(j_1, \ldots, j_l, j_1)$, we have the constraint

$$C_{j_1,\ldots,j_l} = i^l A_{j_1 j_2} \ldots A_{j_{l-1} j_l} A_{j_l j_1}. \tag{2.27}$$

Downsides of this method include that the number of fermions is required to be even and that there is no direct correspondence between qubit occupation and fermionic mode occupation.

2.5.2 Verstraete-Cirac Encoding

Another Majorana fermion method is the Verstraete-Cirac encoding [47]. Auxiliary qubits are added for each lattice site, and set up in the ground state of the auxiliary Hamiltonian

$$\mathcal{H}_{\text{aux}} = \sum_{\langle i,j \rangle} i c_i d_j \tag{2.28}$$

for Majoranas c and d, and the sum over links in a graph. The operators $i c_i d_j$ leave the ground state of $\mathcal{H}_{\text{aux}}$ unchanged, so they can be appended to the physical Hamiltonian without changing the calculated energies. This conversion from quadratic to quartic operators is similar to what I will do with the $\eta\chi$ method, which I find easier to understand. Another presentation of the Verstraete-Cirac encoding is in Ref. [48]. They show that in 3 dimensions, the maximum Pauli weight of a hopping term is 12, independent of the lattice size. This is the same as I will find for the $\eta\chi$ method.

2.5.3 Other Majorana-Based Methods

The compact mapping of Ref. [17] constructs edge and vertex operators in a way similar to Bravyi-Kitaev superfast. It applies only to a square lattice, and adds auxiliary qubits to every other face of the lattice. This results in requiring fewer qubits than the Verstraete-Cirac encoding, as well as having a lower Pauli string length for hopping terms.

Bravyi-Kitaev superfast was generalized to the custom fermionic codes of Ref. [11]. It uses local Majoranas in a similar way to the $\eta\chi$ method. A discussion of parallelization of custom fermionic codes is in Ref. [6].

2.6 Equivalence of Fermion to Qubit Mappings

A recent paper has shown, using finite-depth generalized local unitary transformations, that two-dimensional fermion to qubit mappings are mathematically equivalent [10]. Starting with an exact bosonization (using Majorana fermions), they obtain the Jordan-Wigner transformation, Verstraete-Cirac encoding, Bravyi-Kitaev superfast, and other encodings. They find a "supercompact" mapping that has a qubit to fermion ratio of only 1.25 for spinless fermions. This is better than the ratio of 2 for Verstraete-Cirac, Bravyi-Kitaev superfast, and $\eta\chi$, but comes at the expense of a potentially higher (although still constant) Pauli string length for hopping terms.

I chose to work with the $\eta\chi$ method because I find it relatively easy to understand and implement, especially since it contains a direct correspondence between qubit occupation and fermionic mode occupation.

CHAPTER 3

THE $\eta\chi$ METHOD

3.1 The $\eta\chi$ Method

The fermion to qubit mapping that I choose to use is the $\eta\chi$ method of Ref. [28]. It requires more qubits than some of the most efficient mappings, but it has more consistently efficient hopping terms and in my opinion it is simpler to understand and implement. A nice presentation of the $\eta\chi$ method for spinless fermions is in Appendix B of [33]. This section will be a similar presentation, but for the more complicated case of nuclear matter (protons and neutrons with spin) on a two-dimensional rectangular lattice with periodic boundary conditions.

For each site, there can be spin-up and spin-down protons and neutrons, with at most one of each fermion species at a site due to the Pauli exclusion principle. This creates a 16-dimensional Hilbert space. We will use the frequently-used mathematical trick of rewriting the fermion creation and annihilation operators in terms of Majorana fermions, which satisfy $\{\gamma^i_{\mathbf{r}}, \gamma^j_{\mathbf{r}'}\} = 2\delta^{ij}\delta_{\mathbf{r},\mathbf{r}'}$:

$$f^p_{\uparrow\mathbf{r}} = \frac{1}{2}(\gamma^1_{\mathbf{r}} - i\gamma^2_{\mathbf{r}}) \tag{3.1}$$

$$f^p_{\downarrow\mathbf{r}} = \frac{1}{2}(\gamma^3_{\mathbf{r}} - i\gamma^4_{\mathbf{r}}) \tag{3.2}$$

$$f^n_{\uparrow\mathbf{r}} = \frac{1}{2}(\gamma^5_{\mathbf{r}} - i\gamma^6_{\mathbf{r}}) \tag{3.3}$$

$$f^n_{\downarrow\mathbf{r}} = \frac{1}{2}(\gamma^7_{\mathbf{r}} - i\gamma^8_{\mathbf{r}}). \tag{3.4}$$

We will next introduce two sets of auxiliary Majorana fermions on each lattice site. There will be 8 η^i, corresponding to the internal degrees of freedom (e.g. site occupation, spin, isospin), and 4 χ^i corresponding to links between adjacent lattice sites.

We define bosonic operators, which (since they are products of two Majoranas) commute at different sites. We call these operators bosonic due to this commutation.

$$\Theta^{ij}_{\mathbf{r}} = i\eta^i_{\mathbf{r}}\eta^j_{\mathbf{r}} \tag{3.5}$$

$$\Lambda^{ij}_{\mathbf{r}} = i\eta^i_{\mathbf{r}}\chi^j_{\mathbf{r}} \tag{3.6}$$

$$\Phi^{ij}_{\mathbf{r}} = -i\chi^i_{\mathbf{r}}\chi^j_{\mathbf{r}} \tag{3.7}$$

We can think of Θ as a single-site operator, describing interactions between nucleon species on the same site. Similarly, Λ describes interaction between nucleons on different lattice sites. Φ is a "pass-through" operator, and will be used to connect lattice sites that are not nearest neighbors. We can write some useful identities about the Φ operator.

$$\begin{aligned}
\Phi^{il} &= -\Lambda^{ji}\Theta^{jk}\Lambda^{kl} && (3.8)\\
&= -\Lambda^{ji}i\eta^j\eta^k i\eta^k\chi^l && (3.9)\\
&= -i\Lambda^{ji}i\eta^j\chi^l && (3.10)\\
&= -i\Lambda^{ji}\Lambda^{jl}. && (3.11)
\end{aligned}$$

We define an association between our initial Majoranas and these bosonic operators, so that we can rewrite our Hamiltonian in terms of these bosonic operators:

$$i\gamma^i_{\mathbf{r}}\gamma^j_{\mathbf{r}} = \Theta^{ij}_{\mathbf{r}} \tag{3.12}$$

$$i\gamma^i_{\mathbf{r}}\gamma^j_{\mathbf{r}'} = \Lambda^{i\mu}_{\mathbf{r}}\Lambda^{j\nu}_{\mathbf{r}'}, \tag{3.13}$$

where the indices μ, ν are determined by the orientation of the edge connecting the lattice sites. We follow the convention of Ref. [30] that 4 is the $+x$ direction, 3 is the $-x$ direction, 2 is the $+y$ direction, and 1 is the $-y$ direction. Note that $\gamma^i \neq \eta^i$, since multiplication by χ^μ is necessary for the terms involving multiple sites.

We now rewrite our $\eta\chi$ Majoranas as Dirac fermions. It may feel as though we are going in circles, but as I just showed, there are subtle differences between the types of Majoranas.

$$\begin{aligned}\eta^1 &= a + a^\dagger, \quad \eta^2 = i(a - a^\dagger)\\ \eta^3 &= b + b^\dagger, \quad \eta^4 = i(b - b^\dagger)\\ \eta^5 &= c + c^\dagger, \quad \eta^6 = i(c - c^\dagger)\\ \eta^7 &= d + d^\dagger, \quad \eta^8 = i(d - d^\dagger)\\ \chi^1 &= e + e^\dagger, \quad \chi^2 = i(e - e^\dagger)\\ \chi^3 &= g + g^\dagger, \quad \chi^4 = i(g - g^\dagger)\end{aligned}$$

Because there are 6 fermion modes, we will need 6 qubits for each site (which we will soon reduce to 5). We apply the Jordan-Wigner transformation to write each fermionic operator in terms of Pauli matrices applied to qubits:

$$\begin{aligned}a^\dagger &\to Q^{(1)+}\\ b^\dagger &\to Z^{(1)}Q^{(2)+}\\ c^\dagger &\to Z^{(1)}Z^{(2)}Q^{(3)+}\\ d^\dagger &\to Z^{(1)}Z^{(2)}Z^{(3)}Q^{(4)+}\\ e^\dagger &\to Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Q^{(5)+}\\ g^\dagger &\to Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Z^{(5)}Q^{(6)+}\end{aligned}$$

,

where the indices are qubit indices, an occupied qubit is represented by $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and

$$Q^+ = \frac{1}{2}(X - iY).$$

This gives us

$$\eta^1 \to X^{(1)}, \qquad \eta^2 \to -Y^{(1)} \tag{3.14}$$

$$\eta^3 \to Z^{(1)}X^{(2)}, \qquad \eta^4 \to -Z^{(1)}Y^{(2)} \tag{3.15}$$

$$\eta^5 \to Z^{(1)}Z^{(2)}X^{(3)}, \qquad \eta^6 \to -Z^{(1)}Z^{(2)}Y^{(3)} \tag{3.16}$$

$$\eta^7 \to Z^{(1)}Z^{(2)}Z^{(3)}X^{(4)}, \qquad \eta^8 \to -Z^{(1)}Z^{(2)}Z^{(3)}Y^{(4)} \tag{3.17}$$

$$\chi^1 \to Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}X^{(5)}, \qquad \chi^2 \to -Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Y^{(5)} \tag{3.18}$$

$$\chi^3 \to Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Z^{(5)}X^{(6)}, \qquad \chi^4 \to -Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Z^{(5)}Y^{(6)} \tag{3.19}$$

The η and χ operators that we have constructed span a 64-dimensional Hilbert space, 4 times as large as the physical Hilbert space we are modeling. Following Ref. [30], we can restrict the on-site parton parity,

$$(-i)^6\Pi_{i=1}^{8}\eta^i\Pi_{j=1}^{4}\chi^j \to -1$$

$$-Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Z^{(5)}Z^{(6)} \to 1.$$

This shows that qubit 6 is redundant, and that we can eliminate operators acting on it by applying these substitutions:

$$X^{(6)} \to 1 \tag{3.20}$$

$$Z^{(6)} \to -Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Z^{(5)} \tag{3.21}$$

$$Y^{(6)} \to -iZ^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Z^{(5)}. \tag{3.22}$$

This changes two of our Pauli string expressions for the Majoranas:

$$\chi^3 \to Z^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}Z^{(5)} \tag{3.23}$$

$$\chi^4 \to i \tag{3.24}$$

We now are able to write the bosonic operators in terms of Pauli operators.

$$\Theta_{\mathbf{r}}^{ij}=\begin{pmatrix} i & Z_{\mathbf{r}}^{(1)} & Y_{\mathbf{r}}^{(1)}X_{\mathbf{r}}^{(2)} & -Y_{\mathbf{r}}^{(1)}Y_{\mathbf{r}}^{(2)} & Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & -Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} \\ -Z_{\mathbf{r}}^{(1)} & i & X_{\mathbf{r}}^{(1)}X_{\mathbf{r}}^{(2)} & -X_{\mathbf{r}}^{(1)}Y_{\mathbf{r}}^{(2)} & X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & -X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} \\ -Y_{\mathbf{r}}^{(1)}X_{\mathbf{r}}^{(2)} & -X_{\mathbf{r}}^{(1)}X_{\mathbf{r}}^{(2)} & i & Z_{\mathbf{r}}^{(2)} & Y_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & -Y_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & Y_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -Y_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} \\ Y_{\mathbf{r}}^{(1)}Y_{\mathbf{r}}^{(2)} & X_{\mathbf{r}}^{(1)}Y_{\mathbf{r}}^{(2)} & -Z_{\mathbf{r}}^{(2)} & i & X_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & -X_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & X_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -X_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} \\ -Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & -X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & -Y_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & -X_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} & i & Z_{\mathbf{r}}^{(3)} & Y_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -Y_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} \\ Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & Y_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & X_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} & -Z_{\mathbf{r}}^{(3)} & i & X_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -X_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} \\ -Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -Y_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -X_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -Y_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & -X_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} & i & Z_{\mathbf{r}}^{(4)} \\ Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} & X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} & Y_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} & X_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} & Y_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} & X_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} & -Z_{\mathbf{r}}^{(4)} & i \end{pmatrix}_{ij}$$

$$\Lambda_{\mathbf{r}}^{ij}=\begin{pmatrix} Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & Y_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & -X_{\mathbf{r}}^{(1)} \\ X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & X_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & Y_{\mathbf{r}}^{(1)} \\ Y_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -Y_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & Y_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & -Z_{\mathbf{r}}^{(1)}X_{\mathbf{r}}^{(2)} \\ X_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -X_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & X_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & Z_{\mathbf{r}}^{(1)}Y_{\mathbf{r}}^{(2)} \\ Y_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -Y_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & Y_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & -Z_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}X_{\mathbf{r}}^{(3)} \\ X_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -X_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & X_{\mathbf{r}}^{(3)}Z_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & Z_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Y_{\mathbf{r}}^{(3)} \\ Y_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -Y_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & Y_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & -Z_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}X_{\mathbf{r}}^{(4)} \\ X_{\mathbf{r}}^{(4)}X_{\mathbf{r}}^{(5)} & -X_{\mathbf{r}}^{(4)}Y_{\mathbf{r}}^{(5)} & X_{\mathbf{r}}^{(4)}Z_{\mathbf{r}}^{(5)} & Z_{\mathbf{r}}^{(1)}Z_{\mathbf{r}}^{(2)}Z_{\mathbf{r}}^{(3)}Y_{\mathbf{r}}^{(4)} \end{pmatrix}_{ij}$$

$$\Phi^{ij}_{\mathbf{r}} = \begin{pmatrix} -i & -Z^{(5)}_{\mathbf{r}} & -Y^{(5)}_{\mathbf{r}} & Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} Z^{(3)}_{\mathbf{r}} Z^{(4)}_{\mathbf{r}} X^{(5)}_{\mathbf{r}} \\ Z^{(5)}_{\mathbf{r}} & -i & -X^{(5)}_{\mathbf{r}} & -Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} Z^{(3)}_{\mathbf{r}} Z^{(4)}_{\mathbf{r}} Y^{(5)}_{\mathbf{r}} \\ Y^{(5)}_{\mathbf{r}} & X^{(5)}_{\mathbf{r}} & -i & Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} Z^{(3)}_{\mathbf{r}} Z^{(4)}_{\mathbf{r}} Z^{(5)}_{\mathbf{r}} \\ Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} Z^{(3)}_{\mathbf{r}} Z^{(4)}_{\mathbf{r}} X^{(5)}_{\mathbf{r}} & -Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} Z^{(3)}_{\mathbf{r}} Z^{(4)}_{\mathbf{r}} Y^{(5)}_{\mathbf{r}} & Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} Z^{(3)}_{\mathbf{r}} Z^{(4)}_{\mathbf{r}} Z^{(5)}_{\mathbf{r}} & i \end{pmatrix}_{ij}$$

We will now consider how to set up an initial state by creating fermion pairs. We first consider a pair of particles adjacent in the x-direction. As an example,

$$\begin{aligned} f^{p\dagger}_{\uparrow,\mathbf{r}} f^{n\dagger}_{\uparrow,\mathbf{r+x}} &= \frac{1}{2}(\gamma^1_{\mathbf{r}} + i\gamma^2_{\mathbf{r}})\frac{1}{2}(\gamma^5_{\mathbf{r+x}} + i\gamma^6_{\mathbf{r+x}}) && (3.25) \\ &= \frac{1}{4}(-i\Lambda^{14}_{\mathbf{r}}\Lambda^{53}_{\mathbf{r+x}} + \Lambda^{14}_{\mathbf{r}}\Lambda^{63}_{\mathbf{r+x}} + \Lambda^{24}_{\mathbf{r}}\Lambda^{53}_{\mathbf{r+x}} + i\Lambda^{24}_{\mathbf{r}}\Lambda^{63}_{\mathbf{r+x}}) && (3.26) \\ &= \frac{1}{4}(-i\Lambda^{14}_{\mathbf{r}}(\Lambda^{53}_{\mathbf{r+x}} + i\Lambda^{63}_{\mathbf{r+x}}) + \Lambda^{24}_{\mathbf{r}}(\Lambda^{53}_{\mathbf{r+x}} + i\Lambda^{63}_{\mathbf{r+x}})) && (3.27) \\ &= \frac{1}{4}(-i\Lambda^{14}_{\mathbf{r}} + \Lambda^{24}_{\mathbf{r}})(\Lambda^{53}_{\mathbf{r+x}} + i\Lambda^{63}_{\mathbf{r+x}}) && (3.28) \\ &= \frac{1}{4}(iX^{(1)}_{\mathbf{r}} + Y^{(1)}_{\mathbf{r}})(Y^{(3)}_{\mathbf{r+x}} Z^{(4)}_{\mathbf{r+x}} Z^{(5)}_{\mathbf{r+x}} + iX^{(3)}_{\mathbf{r+x}} Z^{(4)}_{\mathbf{r+x}} Z^{(5)}_{\mathbf{r+x}}) && (3.29) \\ &= -Q^{+(1)}_{\mathbf{r}} Q^{+(3)}_{\mathbf{r+x}} Z^{(4)}_{\mathbf{r+x}} Z^{(5)}_{\mathbf{r+x}}. && (3.30) \end{aligned}$$

To obtain the corresponding terms for other nucleon species, the first indices on the Λ operators are changed according to particle type. The 2nd indices are changed from {4,3} to {2,1} for particles adjacent in the y-direction.

For two particles of different types on the same lattice site, we obtain

$$\begin{aligned} f^{p\dagger}_{\uparrow} f^{n\dagger}_{\uparrow} &= \frac{1}{2}(\gamma^1 + i\gamma^2)\frac{1}{2}(\gamma^5 + i\gamma^6) && (3.31) \\ &= \frac{1}{4}(-i\Theta^{15} + \Theta^{16} + \Theta^{25} + i\Theta^{26}) && (3.32) \\ &= \frac{1}{4}(-iY^{(1)}Z^{(2)}X^{(3)} - Y^{(1)}Z^{(2)}Y^{(3)} + X^{(1)}Z^{(2)}X^{(3)} - iX^{(1)}Z^{(2)}Y^{(3)}) && (3.33) \\ &= \frac{1}{4}(-iY^{(1)}Z^{(2)}(X^{(3)} - iY^{(3)}) + X^{(1)}Z^{(2)}(X^{(3)} - iY^{(3)})) && (3.34) \\ &= \frac{1}{4}(X^{(1)} - iY^{(1)})Z^{(2)}(X^{(3)} - iY^{(3)}) && (3.35) \\ &= Q^{+(1)}Z^{(2)}Q^{+(3)}. && (3.36) \end{aligned}$$

The parity of the site is unchanged, so no operation on the auxiliary qubit is needed.

Now that we have constructed the initial state, we will write our Hamiltonian in terms of Pauli operators. The kinetic portion of Hamiltonians realistic to nuclear physics consists primarily of hopping terms.

$$
\begin{aligned}
f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}'} + f^{p\dagger}_{\uparrow,\mathbf{r}'} f^{p}_{\uparrow,\mathbf{r}} &= \frac{1}{4}(\gamma^1_{\mathbf{r}} + i\gamma^2_{\mathbf{r}})(\gamma^1_{\mathbf{r}'} - i\gamma^2_{\mathbf{r}'}) + \frac{1}{4}(\gamma^1_{\mathbf{r}'} + i\gamma^2_{\mathbf{r}'})(\gamma^1_{\mathbf{r}} - i\gamma^2_{\mathbf{r}}) && (3.37)\\
&= \frac{1}{2}(i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r}'} - i\gamma^1_{\mathbf{r}}\gamma^2_{\mathbf{r}'}) && (3.38)\\
&= \frac{1}{2}(\Lambda^{2\mu}_{\mathbf{r}}\Lambda^{1\nu}_{\mathbf{r}'} - \Lambda^{1\mu}_{\mathbf{r}}\Lambda^{2\nu}_{\mathbf{r}'}) && (3.39)
\end{aligned}
$$

For a hop in the +x-direction, $\mu = 4$ and $\nu = 3$. This gives us

$$
f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}'} + f^{p\dagger}_{\uparrow,\mathbf{r}'} f^{p}_{\uparrow,\mathbf{r}} = \frac{1}{2}(X^{(1)}_{\mathbf{r}} X^{(1)}_{\mathbf{r}'} + Y^{(1)}_{\mathbf{r}} Y^{(1)}_{\mathbf{r}'}) Z^{(2)}_{\mathbf{r}'} Z^{(3)}_{\mathbf{r}'} Z^{(4)}_{\mathbf{r}'} Z^{(5)}_{\mathbf{r}'}. \qquad (3.40)
$$

For a hop in the +y-direction, $\mu = 2$ and $\nu = 1$.

A Hamiltonian with smearing (as in Ref. [25]) would also have next-to-nearest neighbor hopping. As an example, we take a hop in the x-direction, where $\mathbf{r}' = \mathbf{r} + \mathbf{2x}$.

$$
\begin{aligned}
f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}'} + f^{p\dagger}_{\uparrow,\mathbf{r}'} f^{p}_{\uparrow,\mathbf{r}} &= \frac{1}{4}(\gamma^1_{\mathbf{r}} + i\gamma^2_{\mathbf{r}})(\gamma^1_{\mathbf{r}'} - i\gamma^2_{\mathbf{r}'}) + \frac{1}{4}(\gamma^1_{\mathbf{r}'} + i\gamma^2_{\mathbf{r}'})(\gamma^1_{\mathbf{r}} - i\gamma^2_{\mathbf{r}})\\
&= \frac{1}{2}(i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r}'} - i\gamma^1_{\mathbf{r}}\gamma^2_{\mathbf{r}'})\\
&= -\frac{1}{2}(i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r+x}} i\gamma^1_{\mathbf{r+x}}\gamma^2_{\mathbf{r+x}} i\gamma^2_{\mathbf{r+x}}\gamma^1_{\mathbf{r+2x}}\\
&\quad - i\gamma^1_{\mathbf{r}}\gamma^1_{\mathbf{r+x}} i\gamma^1_{\mathbf{r+x}}\gamma^2_{\mathbf{r+x}} i\gamma^2_{\mathbf{r+x}}\gamma^2_{\mathbf{r+2x}})\\
&= -\frac{1}{2}(\Lambda^{24}_{\mathbf{r}}\Lambda^{13}_{\mathbf{r+x}}\Theta^{12}_{\mathbf{r+x}}\Lambda^{24}_{\mathbf{r+x}}\Lambda^{13}_{\mathbf{r+2x}}\\
&\quad - \Lambda^{14}_{\mathbf{r}}\Lambda^{13}_{\mathbf{r+x}}\Theta^{12}_{\mathbf{r+x}}\Lambda^{24}_{\mathbf{r+x}}\Lambda^{23}_{\mathbf{r+2x}})\\
&= \frac{1}{2}(\Lambda^{24}_{\mathbf{r}}\Phi^{34}_{\mathbf{r+x}}\Lambda^{13}_{\mathbf{r+2x}} - \Lambda^{14}_{\mathbf{r}}\Phi^{34}_{\mathbf{r+x}}\Lambda^{23}_{\mathbf{r+2x}})
\end{aligned}
$$

This shows the usefulness of the pass-through operators Φ.

We can similarly consider next-to-nearest neighbor hopping in the y-direction, with $\mathbf{r}' = \mathbf{r} + \mathbf{2y}$.

$$
\begin{aligned}
f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}'} + f^{p\dagger}_{\uparrow,\mathbf{r}'} f^{p}_{\uparrow,\mathbf{r}} &= \frac{1}{4}(\gamma^1_{\mathbf{r}} + i\gamma^2_{\mathbf{r}})(\gamma^1_{\mathbf{r}'} - i\gamma^2_{\mathbf{r}'}) + \frac{1}{4}(\gamma^1_{\mathbf{r}'} + i\gamma^2_{\mathbf{r}'})(\gamma^1_{\mathbf{r}} - i\gamma^2_{\mathbf{r}}) \\
&= \frac{1}{2}(i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r}'} - i\gamma^1_{\mathbf{r}}\gamma^2_{\mathbf{r}'}) \\
&= -\frac{1}{2}(i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r+y}} i\gamma^1_{\mathbf{r+y}}\gamma^2_{\mathbf{r+y}} i\gamma^2_{\mathbf{r+y}}\gamma^1_{\mathbf{r+2y}} \\
&\quad - i\gamma^1_{\mathbf{r}}\gamma^1_{\mathbf{r+y}} i\gamma^1_{\mathbf{r+y}}\gamma^2_{\mathbf{r+y}} i\gamma^2_{\mathbf{r+y}}\gamma^2_{\mathbf{r+2y}}) \\
&= -\frac{1}{2}(\Lambda^{22}_{\mathbf{r}}\Lambda^{11}_{\mathbf{r+y}}\Theta^{12}_{\mathbf{r+y}}\Lambda^{22}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}} \\
&\quad - \Lambda^{12}_{\mathbf{r}}\Lambda^{11}_{\mathbf{r+y}}\Theta^{12}_{\mathbf{r+y}}\Lambda^{22}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}}) \\
&= \frac{1}{2}(\Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}} - \Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}})
\end{aligned}
$$

Typical Hamiltonians in nuclear physics also contain terms proportional to number operators:

$$
\begin{aligned}
f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}} &= \frac{1}{2}(\gamma^1_{\mathbf{r}} + i\gamma^2_{\mathbf{r}})\frac{1}{2}(\gamma^1_{\mathbf{r}} - i\gamma^2_{\mathbf{r}}) \\
&= \frac{1}{4}(1 - i\gamma^1_{\mathbf{r}}\gamma^2_{\mathbf{r}} + i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r}} + 1) \\
&= \frac{1-\Theta^{12}_{\mathbf{r}}}{2} = \frac{1-Z^{(1)}_{\mathbf{r}}}{2}.
\end{aligned}
$$

So we have seen how to write the Hamiltonian as as sum of Pauli strings.

Even after eliminating a redundant qubit, the Hilbert space of the qubits at each lattice site is larger than the physical Hilbert space by a factor of 2. We will resolve this by introducing more constraints [28]. The first type is plaquette constraints that arise from constructing an identity operator around a plaquette in the lattice:

$$
\begin{aligned}
\Phi^{24}_{\mathbf{r}}\Phi^{32}_{\mathbf{r+x}}\Phi^{13}_{\mathbf{r+x+y}}\Phi^{41}_{\mathbf{r+y}} &= -1 \\
Z^{(1)}_{\mathbf{r}}Z^{(2)}_{\mathbf{r}}Z^{(3)}_{\mathbf{r}}Z^{(4)}_{\mathbf{r}}Y^{(5)}_{\mathbf{r}}X^{(5)}_{\mathbf{r+x}}Y^{(5)}_{\mathbf{r+x+y}}Z^{(1)}_{\mathbf{r+y}}Z^{(2)}_{\mathbf{r+y}}Z^{(3)}_{\mathbf{r+y}}Z^{(4)}_{\mathbf{r+y}}X^{(5)}_{\mathbf{r+y}} &= 1
\end{aligned}
$$

We need to apply this constraint before creating any particles, to initialize the vacuum state. When there are no particles in the lattice, we can impose this plaquette constraint by applying the following operator to each plaquette:

$$
C_{\mathbf{r}} = Y^{(5)}_{\mathbf{r}}X^{(5)}_{\mathbf{r+x}}Y^{(5)}_{\mathbf{r+x+y}}X^{(5)}_{\mathbf{r+y}}.
$$

There are not quite enough plaquette constraints to constrain the Hilbert space to its physical size due to the periodicity of the lattice, so we also have Wilson loop constraints. These arise from constructing identities along horizonal or vertical loops around the toroidal lattice,

$$\Phi^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\dots\Phi^{12}_{\mathbf{r-y}} = -1, \tag{3.41}$$

$$\Phi^{34}_{\mathbf{r}}\Phi^{34}_{\mathbf{r+x}}\dots\Phi^{34}_{\mathbf{r-x}} = -1. \tag{3.42}$$

This constraint can be applied to the vacuum state by applying X gates to the auxiliary qubits along the diagonal of the lattice [29].

3.2 Exponentiation of $XX + YY$

As we just saw, the hopping terms in the Hamiltonian, once rewritten as Pauli strings, are often of the form

$$(XX + YY)ZZ, \tag{3.43}$$

with varying number of Z operators. I figured out a method to exponentiate these terms using one circuit instead of two, as would be needed for the basic exponentiation method.

We start by noting that in general a two-qubit circuit can be implemented using only 3 CNOT gates [46]. Operators meeting certain conditions can be implemented even more efficiently: for example $XX + ZZ$ can be exponentiated using only two CNOT gates [42]. This can be converted to $XX + YY$ or $XY + YX$, that occur in hopping terms, by applying single-qubit gates. We then use our idea of "controlled reversal gates" [3]. We note that the eigenvalues of the Z operator are 1 and -1, depending on the occupation of the qubit. Hence the effect of the Z string is to possibly apply a negative sign to the overall exponential. We know that Z anti-commutes with $XX + YY$, so CZ will act as our controlled reversal gate. This gives the circuit shown in Fig. 3.1.

Similarly to the controlled version of the canonical Pauli string exponential circuit, if the rotation gates are removed from the circuit in Fig. 3.1, the rest of the circuit cancels out. Using Qiskit [36], I was able to determine that the controlled version of my circuit takes 4 CNOT gates. This is the same number of CNOT gates as would be needed for controlling two copies of the canonical circuit. My controlled circuit is shown in Fig. 3.2.

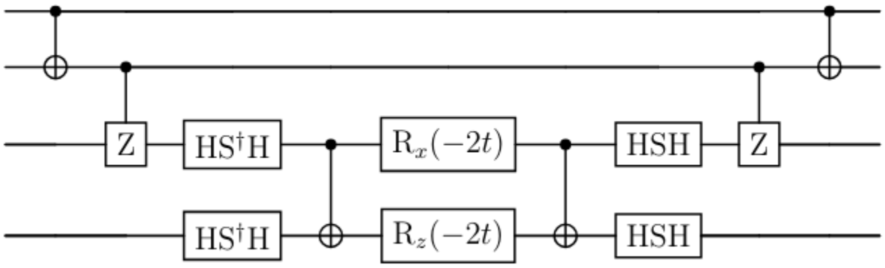


Figure 3.1 Exponentiation of $(XX + YY)ZZ$. The parity of the qubits that Z acts on is collected. Then the CZ gates flip the sign of the $XX + YY$ circuit if necessary. The H and S gates are needed to perform $XX + YY$ instead of $XX + ZZ$.

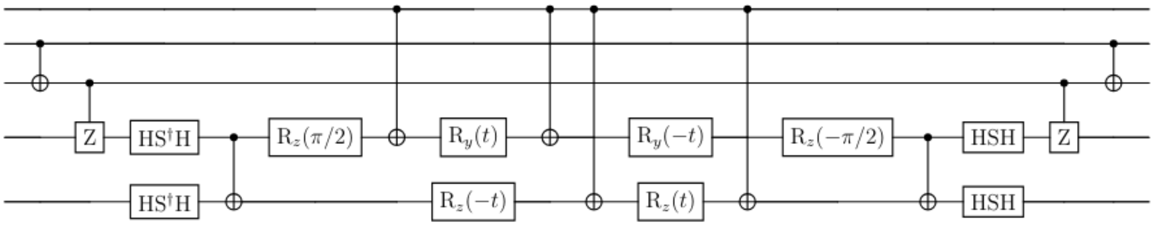


Figure 3.2 Controlled exponentiation of $(XX + YY)ZZ$. The top qubit is the control qubit.

3.3 Single-particle creation

So far, we have only considered the case where the initial state has an even number of particles. However by changing the direction of one link in the lattice and making the association $\gamma^1_{\mathbf{0}} \to \Lambda^{11}_{\mathbf{0}}$ we can access the odd parity sector [33, 28]. At the origin, where the lattice defect is,

$$f^{p\dagger}_{\uparrow,\mathbf{0}} = \frac{1}{2}(\gamma^1_{\mathbf{0}} + i\gamma^2_{\mathbf{0}}) \tag{3.44}$$

$$= \frac{1}{2}\Lambda^{11}_{\mathbf{0}}(1 + \Theta^{12}_{\mathbf{0}}) \tag{3.45}$$

$$= \frac{1}{2}Y^{(1)}Z^{(2)}Z^{(3)}Z^{(4)}X^{(5)}(1 + Z^{(1)}). \tag{3.46}$$

(For the rest of this section, all operators are implicitly at the origin.) The other cases are slightly more complicated, since we only have that $\gamma^1 \to \Lambda^{11}$. But we can insert the identity $\gamma^1\gamma^1$ to add additional Θ terms and eliminate single γ^i for $i \neq 1$.

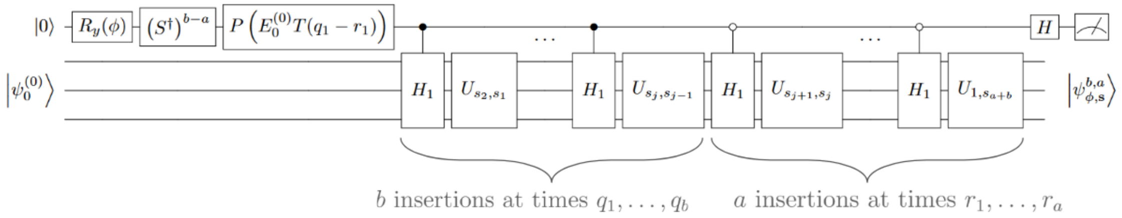
$$
\begin{aligned}
f_{\downarrow}^{p\dagger} &= \frac{1}{2}(\gamma^3 + i\gamma^4) \\
&= \frac{1}{2}\gamma^3(1 + \Theta^{34}) \\
&= i\frac{1}{2}\gamma^1\Theta^{13}(1 + \Theta^{34}) \\
&= -\frac{1}{2}Y^{(2)}Z^{(3)}Z^{(4)}X^{(5)}(1 + Z^{(2)}) \\
f_{\uparrow}^{n\dagger} &= \frac{1}{2}(\gamma^5 + i\gamma^6) \\
&= \frac{1}{2}\gamma^5(1 + \Theta^{56}) \\
&= i\frac{1}{2}\gamma^1\Theta^{15}(1 + \Theta^{56}) \\
&= -\frac{1}{2}Y^{(3)}Z^{(4)}X^{(5)}(1 + Z^{(3)}) \\
f_{\downarrow}^{n\dagger} &= \frac{1}{2}(\gamma^7 + i\gamma^8) \\
&= \frac{1}{2}\gamma^7(1 + \Theta^{78}) \\
&= i\frac{1}{2}\gamma^1\Theta^{17}(1 + \Theta^{78}) \\
&= -\frac{1}{2}Y^{(4)}X^{(5)}(1 + Z^{(4)})
\end{aligned}
$$

Single particles can be created away from the origin by constructing chains of operators to connect to the origin, but this is unnecessary due to translational invariance. Further work needs to be done to determine the impact on the constraints of an odd number of particles.

3.4 The $\eta\chi$ Method in 3 Dimensions

The $\eta\chi$ method can be extended to work on a three-dimensional lattice. The primary change is that there are now two more nearest neighbors to each lattice site. This requires adding two additional χ Majoranas for each lattice site. Accordingly, one more qubit is required per lattice site, and the lengths of the Pauli strings for hopping terms increase by up to 2.

Additionally, due to the increased degree of freedom, there are many more plaquette constraints, since they now occur in three planes. There are also additional Wilson loop constraints. This approximately triples the computational cost of setting up the vacuum state.


$|0\rangle$
$R_y(\phi)$
$(S^\dagger)^{b-a}$
$P\left(E_0^{(0)}T(q_1-r_1)\right)$
H
$\left|\psi_0^{(0)}\right\rangle$
H_1
U_{s_2,s_1}
$U_{s_j,s_{j-1}}$
U_{s_{j+1},s_j}
$U_{1,s_{a+b}}$
$\left|\psi_{\phi,s}^{b,a}\right\rangle$
b insertions at times $q_1,\ldots,q_b$
a insertions at times $r_1,\ldots,r_a$

where

$$U_{s_{j+1},s_j} = e^{-iTH_0(s_{j+1}-s_j)}$$

Figure 3.3 Circuit used to calculate $\left\langle\psi_0^{(b)}\middle|O\middle|\psi_0^{(a)}\right\rangle$. Credit: Nicholas Cariello

3.5 Using Perturbation Theory

A common method used in physics is perturbation theory, where a complicated, hard to solve, Hamiltonian is decomposed into a part that is easy to solve plus a part that is harder to solve. Then observables, such as energy, can be approximated using the wave functions for the simple Hamiltonian. We want to do perturbation theory on a quantum computer. My colleague Nicholas Cariello constructed a circuit that can calculate $\left\langle\psi_0^{(b)}\middle|O\middle|\psi_0^{(a)}\right\rangle$, where $\left|\psi_0^{(a)}\right\rangle$ is the ath-order correction to the ground state wave function [7]. The circuit for this is shown in Fig. 3.3. The circuit contains single-qubit gates, time evolution of the base Hamiltonian, and controlled application of the perturbation. Monte Carlo methods will be used to handle perturbations that are sums of unitary operators.

We consider a Hamiltonian of the form $H_0 + cH_1$, where H_0 has terms that preserve nucleon species and H_1 has more complicated terms that mix nucleon species. This way we can do a simpler time evolution with only the H_0 terms, and never have to exponentiate H_1. Instead of using the full $\eta\chi$ method with 5 qubits per lattice site and the associated long Pauli strings, we could use something like the stacked lattices from Ref. [48] (a separate lattice with local encoding for each nucleon species) to reduce the CNOT cost. However, I'm not sure how we would then handle the perturbative terms, due to the separate local encodings.

We can instead use a modification I propose: we can construct a composite lattice as shown in Fig. 3.4 and construct a local encoding using the $\eta\chi$ method on the entire composite lattice. Then we can easily define the perturbative terms.

$p \uparrow$	$p \downarrow$
$n \uparrow$	$n \downarrow$

Figure 3.4 The layout of the composite lattice I propose. Proton and neutron spin-up and -down each are located in their own quadrant.

We will use as the base Hamiltonian a Hamiltonian from leading order pionless effective field theory without a 3-body term [24].

$$H_0 = H_{\text{free}} + V,$$

where

$$V = \frac{C}{2} \int \mathrm{d}^3\vec{r} : [\rho(\vec{r})]^2 :,$$

normal ordering is represented by ::, and

$$\rho(\vec{r}) = \sum_{\sigma,\tau} f^\dagger_{\sigma\tau}(\vec{r}) f_{\sigma\tau}(\vec{r}).$$

$$H_{\text{free}} = \frac{2}{m} \sum_{\vec{n}} \sum_{\sigma,\tau} f^\dagger_{\sigma\tau}(\vec{n}) f_{\sigma\tau}(\vec{n}) - \frac{1}{2m} \sum_{\vec{n}} \sum_{\sigma\tau} \sum_{l=x,y} \left[f^\dagger_{\sigma\tau}(\vec{n}) f_{\sigma\tau}(\vec{n}+\vec{l}) + f_{\sigma\tau}(\vec{n}) f^\dagger_{\sigma\tau}(\vec{n}+\vec{l}) \right].$$

We note that the hopping terms preserve nucleon species, i.e. each hopping term is contained within one quadrant of the composite lattice.

When we apply the $\eta\chi$ method, we have only one nucleon species per lattice site, as the four nucleon species are spread across lattice sites in the four quadrants of the composite lattice. This means that we will need two η Majoranas and four χ Majoranas for each site. This initially requires three qubits. But, as in the case where we consider full nuclear matter at each lattice site, one qubit is redundant:

$$iZ^{(1)}Z^{(2)}Z^{(3)} \to 1.$$

We can replace the Pauli operators acting on the 3rd qubit as follows:

$$X^{(3)} \to 1$$

$$Z^{(3)} \to iZ^{(1)}Z^{(2)}$$

$$Y^{(3)} \to -Z^{(1)}Z^{(2)}$$

We need a total of eight qubits for each physical lattice site: two (including one auxiliary qubit) for each nucleon species. However the bosonic operators have Pauli strings of length at most 2 instead of 5:

$$\Theta^{ij}_{\mathbf{r}} = \begin{pmatrix} i & Z^{(1)}_{\mathbf{r}} \\ -Z^{(1)}_{\mathbf{r}} & i \end{pmatrix}_{ij}$$

$$\Lambda^{ij}_{\mathbf{r}} = \begin{pmatrix} Y^{(1)}_{\mathbf{r}} X^{(2)}_{\mathbf{r}} & -Y^{(1)}_{\mathbf{r}} Y^{(2)}_{\mathbf{r}} & Y^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} & X^{(1)}_{\mathbf{r}} \\ X^{(1)}_{\mathbf{r}} X^{(2)}_{\mathbf{r}} & -X^{(1)}_{\mathbf{r}} Y^{(2)}_{\mathbf{r}} & X^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} & -Y^{(1)}_{\mathbf{r}} \end{pmatrix}_{ij}$$

$$\Phi^{ij}_{\mathbf{r}} = \begin{pmatrix} -i & -Z^{(2)}_{\mathbf{r}} & -Y^{(2)}_{\mathbf{r}} & -Z^{(1)}_{\mathbf{r}} X^{(2)}_{\mathbf{r}} \\ Z^{(2)}_{\mathbf{r}} & -i & -X^{(2)}_{\mathbf{r}} & Z^{(1)}_{\mathbf{r}} Y^{(2)}_{\mathbf{r}} \\ Y^{(2)}_{\mathbf{r}} & X^{(2)}_{\mathbf{r}} & -i & -Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} \\ -Z^{(1)}_{\mathbf{r}} X^{(2)}_{\mathbf{r}} & Z^{(1)}_{\mathbf{r}} Y^{(2)}_{\mathbf{r}} & -Z^{(1)}_{\mathbf{r}} Z^{(2)}_{\mathbf{r}} & i \end{pmatrix}_{ij}$$

A hopping term in the $+x$ direction gives

$$\begin{aligned} f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}'} + f^{p\dagger}_{\uparrow,\mathbf{r}'} f^{p}_{\uparrow,\mathbf{r}} &= \frac{1}{2}(\Lambda^{24}_{\mathbf{r}}\Lambda^{13}_{\mathbf{r}'} - \Lambda^{14}_{\mathbf{r}}\Lambda^{23}_{\mathbf{r}'}) && (3.47) \\ &= \frac{1}{2}(X^{(1)}_{\mathbf{r}} X^{(1)}_{\mathbf{r}'} + Y^{(1)}_{\mathbf{r}} Y^{(1)}_{\mathbf{r}'}) Z^{(2)}_{\mathbf{r}'}. && (3.48) \end{aligned}$$

This is a shorter Pauli string than without using the composite lattice. At the cost of the extra auxiliary qubits, we now have a maximum Pauli string length for hopping terms of 4, while it was previously 10.

As we have seen, number operators can be written as

$$N^{p}_{\uparrow,\mathbf{r}} = f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}} = \frac{1 - \Theta^{12}_{\mathbf{r}}}{2} = \frac{1 - Z^{(1)}_{\mathbf{r}}}{2}.$$

So

$$N_{\mathbf{r}} N_{\mathbf{r}'} = \frac{1}{4}(1 - Z^{(1)}_{\mathbf{r}})(1 - Z^{(1)}_{\mathbf{r}'}).$$

Exponentiating $(C/2)N_{\mathbf{r}}N_{\mathbf{r}'}$ gives us, up to a constant prefactor,

$$e^{-itCZ^{(1)}_{\mathbf{r}}/8} e^{-itCZ^{(1)}_{\mathbf{r}'}/8} e^{itCZ^{(1)}_{\mathbf{r}} Z^{(1)}_{\mathbf{r}'}/8}.$$

We also should consider the special case where both number operators act on the same site:

$$N_{\mathbf{r}} N_{\mathbf{r}} = \frac{1}{4}(1 - Z^{(1)}_{\mathbf{r}})(1 - Z^{(1)}_{\mathbf{r}}) = \frac{1}{4}(2 - 2Z^{(1)}_{\mathbf{r}}) = N_{\mathbf{r}}.$$

This is the expected result, as both eigenvalues of the number operator (0 and 1) are idempotent under multiplication.

3.6 Perturbative Term

So far we have considered how to implement the base Hamiltonian, composed of hopping terms and number operator terms. We now consider as a perturbation an isospin term,

$$V_{I^2} = \frac{C_{I^2}}{2} \sum_{\mathbf{r}} : \left[(\sum_{\sigma} f^{p\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} + f^{n\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}})^2 + (\sum_{\sigma} -i f^{p\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} + i f^{n\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}})^2 + (\sum_{\sigma} f^{p\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} - f^{n\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}})^2 \right] : \tag{3.49}$$

$$= \frac{C_{I^2}}{2} \sum_{\mathbf{r}} \sum_{\sigma \neq \sigma'} (4 f^{p\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} - 2 f^{p\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} + 2 f^{p\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma',\mathbf{r}} f^{p}_{\sigma',\mathbf{r}} + 2 f^{n\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} f^{p\dagger}_{\sigma',\mathbf{r}} f^{n}_{\sigma',\mathbf{r}} \tag{3.50}$$

$$+ f^{p\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} f^{p\dagger}_{\sigma',\mathbf{r}} f^{p}_{\sigma',\mathbf{r}} - f^{p\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma',\mathbf{r}} f^{n}_{\sigma',\mathbf{r}} - f^{n\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} f^{p\dagger}_{\sigma',\mathbf{r}} f^{p}_{\sigma',\mathbf{r}} + f^{n\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma',\mathbf{r}} f^{n}_{\sigma',\mathbf{r}}) \tag{3.51}$$

$$= \frac{C_{I^2}}{2} \sum_{\mathbf{r}} \sum_{\sigma \neq \sigma'} (-6 f^{p\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} + 4 f^{p\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma',\mathbf{r}} f^{p}_{\sigma',\mathbf{r}} \tag{3.52}$$

$$+ f^{p\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} f^{p\dagger}_{\sigma',\mathbf{r}} f^{p}_{\sigma',\mathbf{r}} + f^{n\dagger}_{\sigma,\mathbf{r}} f^{n}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma',\mathbf{r}} f^{n}_{\sigma',\mathbf{r}} - 2 f^{p\dagger}_{\sigma,\mathbf{r}} f^{p}_{\sigma,\mathbf{r}} f^{n\dagger}_{\sigma',\mathbf{r}} f^{n}_{\sigma',\mathbf{r}}). \tag{3.53}$$

Except for the second term, these are products of two number operators. (We choose to not write the terms in normal ordering to make the number operators apparent.) So they are simple in principle to implement. Applying the $\eta\chi$ method to the second term, using a 2x2 composite lattice, we find (for $\sigma = \uparrow$):

$$
\begin{aligned}
&f^{p\dagger}_{\uparrow,\mathbf{r}}f^{n}_{\uparrow,\mathbf{r}}f^{n\dagger}_{\downarrow,\mathbf{r}}f^{p}_{\downarrow,\mathbf{r}}+f^{p\dagger}_{\downarrow,\mathbf{r}}f^{n}_{\downarrow,\mathbf{r}}f^{n\dagger}_{\uparrow,\mathbf{r}}f^{p}_{\uparrow,\mathbf{r}}=\frac{1}{16}(\gamma^1_{\mathbf{r}}+i\gamma^2_{\mathbf{r}})(\gamma^1_{\mathbf{r+2y}}-i\gamma^2_{\mathbf{r+2y}})(\gamma^1_{\mathbf{r+2y+2x}}+i\gamma^2_{\mathbf{r+2y+2x}})(\gamma^1_{\mathbf{r+2x}}-i\gamma^2_{\mathbf{r+2x}})\\
&\quad+\frac{1}{16}(\gamma^1_{\mathbf{r+2x}}+i\gamma^2_{\mathbf{r+2x}})(\gamma^1_{\mathbf{r+2y+2x}}-i\gamma^2_{\mathbf{r+2y+2x}})(\gamma^1_{\mathbf{r+2y}}+i\gamma^2_{\mathbf{r+2y}})(\gamma^1_{\mathbf{r}}-i\gamma^2_{\mathbf{r}})\\
&=\frac{1}{16}(\gamma^1_{\mathbf{r}}\gamma^1_{\mathbf{r+2y}}-i\gamma^1_{\mathbf{r}}\gamma^2_{\mathbf{r+2y}}+i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r+2y}}+\gamma^2_{\mathbf{r}}\gamma^2_{\mathbf{r+2y}})\\
&\quad(\gamma^1_{\mathbf{r+2y+2x}}\gamma^1_{\mathbf{r+2x}}-i\gamma^1_{\mathbf{r+2y+2x}}\gamma^2_{\mathbf{r+2x}}+i\gamma^2_{\mathbf{r+2y+2x}}\gamma^1_{\mathbf{r+2x}}+\gamma^2_{\mathbf{r+2y+2x}}\gamma^2_{\mathbf{r+2x}})\\
&\quad+\frac{1}{16}(\gamma^1_{\mathbf{r+2x}}\gamma^1_{\mathbf{r+2y+2x}}-i\gamma^1_{\mathbf{r+2x}}\gamma^2_{\mathbf{r+2y+2x}}+i\gamma^2_{\mathbf{r+2x}}\gamma^1_{\mathbf{r+2y+2x}}+\gamma^2_{\mathbf{r+2x}}\gamma^2_{\mathbf{r+2y+2x}})\\
&\quad(\gamma^1_{\mathbf{r+2y}}\gamma^1_{\mathbf{r}}-i\gamma^1_{\mathbf{r+2y}}\gamma^2_{\mathbf{r}}+i\gamma^2_{\mathbf{r+2y}}\gamma^1_{\mathbf{r}}+\gamma^2_{\mathbf{r+2y}}\gamma^2_{\mathbf{r}})\\
&=\frac{1}{16}(-i\Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}}-\Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}}+\Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}}-i\Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}})\\
&\quad(i\Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}}+\Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}}-\Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}}+i\Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}})\\
&\quad+\frac{1}{16}(-i\Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}}-\Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}}+\Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}}-i\Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}})\\
&\quad(i\Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}}+\Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}}-\Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}}+i\Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}})\\
&=\frac{1}{16}(-i\Lambda^{12}_{\mathbf{r}}+\Lambda^{22}_{\mathbf{r}})(\Phi^{12}_{\mathbf{r+y}})(\Lambda^{11}_{\mathbf{r+2y}}-i\Lambda^{21}_{\mathbf{r+2y}})(-i\Lambda^{12}_{\mathbf{r+2x}}+\Lambda^{22}_{\mathbf{r+2x}})(-\Phi^{12}_{\mathbf{r+y+2x}})(\Lambda^{11}_{\mathbf{r+2y+2x}}-i\Lambda^{21}_{\mathbf{r+2y+2x}})\\
&\quad+\frac{1}{16}(-i\Lambda^{12}_{\mathbf{r+2x}}+\Lambda^{22}_{\mathbf{r+2x}})(\Phi^{12}_{\mathbf{r+y+2x}})(\Lambda^{11}_{\mathbf{r+2y+2x}}-i\Lambda^{21}_{\mathbf{r+2y+2x}})(-i\Lambda^{12}_{\mathbf{r}}+\Lambda^{22}_{\mathbf{r}})(-\Phi^{12}_{\mathbf{r+y}})(\Lambda^{11}_{\mathbf{r+2y}}-i\Lambda^{21}_{\mathbf{r+2y}})\\
&=-\frac{1}{8}(iY^{(1)}_{\mathbf{r}}Y^{(2)}_{\mathbf{r}}-X^{(1)}_{\mathbf{r}}Y^{(2)}_{\mathbf{r}})(-Z^{(2)}_{\mathbf{r+y}})(Y^{(1)}_{\mathbf{r+2y}}X^{(2)}_{\mathbf{r+2y}}-iX^{(1)}_{\mathbf{r+2y}}X^{(2)}_{\mathbf{r+2y}})\\
&\quad(iY^{(1)}_{\mathbf{r+2x}}Y^{(2)}_{\mathbf{r+2x}}-X^{(1)}_{\mathbf{r+2x}}Y^{(2)}_{\mathbf{r+2x}})(-Z^{(2)}_{\mathbf{r+y+2x}})(Y^{(1)}_{\mathbf{r+2y+2x}}X^{(2)}_{\mathbf{r+2y+2x}}-iX^{(1)}_{\mathbf{r+2y+2x}}X^{(2)}_{\mathbf{r+2y+2x}})\\
&=\frac{1}{8}(-iY^{(1)}_{\mathbf{r}}+X^{(1)}_{\mathbf{r}})(-Y^{(2)}_{\mathbf{r}}Z^{(2)}_{\mathbf{r+y}}X^{(2)}_{\mathbf{r+2y}})(-i)(iY^{(1)}_{\mathbf{r+2y}}+X^{(1)}_{\mathbf{r+2y}})\\
&\quad(-iY^{(1)}_{\mathbf{r+2x}}+X^{(1)}_{\mathbf{r+2x}})(Y^{(2)}_{\mathbf{r+2x}}Z^{(2)}_{\mathbf{r+y+2x}}X^{(2)}_{\mathbf{r+2y+2x}})(-i)(iY^{(1)}_{\mathbf{r+2y+2x}}+X^{(1)}_{\mathbf{r+2y+2x}})\\
&=2(Q^{+(1)}_{\mathbf{r}}Y^{(2)}_{\mathbf{r}}Z^{(2)}_{\mathbf{r+y}}X^{(2)}_{\mathbf{r+2y}}Q^{-(1)}_{\mathbf{r+2y}})(Q^{+(1)}_{\mathbf{r+2x}}Y^{(2)}_{\mathbf{r+2x}}Z^{(2)}_{\mathbf{r+y+2x}}X^{(2)}_{\mathbf{r+2y+2x}}Q^{-(1)}_{\mathbf{r+2y+2x}}).
\end{aligned}
$$

3.7 Implementation of the $\eta\chi$ Method

I prepared code to implement the (controlled and non-controlled) time evolution of nuclear lattice effective field theory Hamiltonians using the $\eta\chi$ method. I am initially using a 2 by 2 composite lattice, but writing the code so that it also works for larger lattice sizes. The general structure of the code is based on the code from Ref. [29].

I start by constructing the grids of physical and auxiliary qubits, and helper functions for finding the qubit indices corresponding to specific lattice sites. The physical qubits are numbered 0 to 15, and the auxiliary qubits are numbered 16 to 31. (For example, the 2 by 2 lattice for proton spin-up uses physical qubits 0, 1, 4, and 5, and auxiliary qubits 16, 17, 20, and 21.) Qiskit initializes every qubit in the $|0\rangle$ state, which is not the vacuum state for the auxiliary qubits. I initialize the auxiliary qubits into their vacuum state by applying the plaquette and Wilson loop constraints. The Wilson loop constraints are implemented with X gates along the diagonal of the auxiliary lattice, in this case qubits 16, 21, 26, and 31. The plaquette constraints are implemented by applying the operator

$$Y_{\mathbf{r}}^{(2)} X_{\mathbf{r+x}}^{(2)} Y_{\mathbf{r+x+y}}^{(2)} X_{\mathbf{r+y}}^{(2)} \tag{3.54}$$

to each plaquette. Qubit 2 is the auxiliary qubit for each lattice site.

Now that the system is initialized in the vacuum state, I can create pairs of particles for the initial state. The two particle creation operators are products of qubit creation operators on each physical qubit and Pauli operators on the auxiliary qubits. Since we know that we are starting from the vacuum state, we note that the action of Q^+ on the vacuum is the same as the action of X on the vacuum.

I can finally construct the Hamiltonian. For each lattice site, I apply the (controlled if necessary) hopping terms in the $+x$ and $+y$ directions, using the circuits from Fig. 3.1 and 3.2. I can then apply the number operator terms using (controlled) R_z gates and the circuit from Ref. [42]. To assist with constructing the hopping terms, I wrote code to calculate the Pauli strings for each bosonic operator. This code also allowed me to easily construct the bosonic operator matrices earlier in the chapter.

CHAPTER 4

CONTRIBUTIONS TO VARIOUS PAPERS

4.1 Scale Invariance and Time Fractals

We consider bosons on a one-dimensional lattice, and show that with the correct choice of parameters there is discrete scale invariance and fractal-like time dependence [27].

4.1.1 Discrete Scale Invariance

If we introduce a single-site deep trapping potential to immobilize one boson at the origin, we obtain that for low energies, the Hamiltonian for the second boson is

$$H(p,r) = 2J_0 \sin(\alpha\pi/2)\Gamma(1-\alpha)|p|^{\alpha-1} + \frac{V_0}{|r|^{\alpha-1}}, \tag{4.1}$$

where α is a parameter which we will set to 2. In the zero energy limit, we obtain wave functions

$$\psi_+(r) = \frac{1}{2}\left(|r|^{i\delta_+} + |r|^{-i\bar{\delta}_+}\right), \tag{4.2}$$

$$\psi_-(r) = \frac{1}{2}\mathrm{sgn}(r)\left(|r|^{i\delta_-} + |r|^{-i\bar{\delta}_-}\right), \tag{4.3}$$

where $\mathrm{sgn}(r)$ is the sign function and $\delta_\pm$ satisfy the constraints

$$\delta_+ = \frac{V_0}{J_0\pi}\coth(\delta_+\pi/2), \tag{4.4}$$

$$\delta_- = \frac{V_0}{J_0\pi}\tanh(\delta_-\pi/2). \tag{4.5}$$

The values of $\delta_\pm$ are real when V_0/J_0 is positive. When $V_0/J_0 \gg \pi$,

$$\delta_+ \approx \delta_- \approx \frac{V_0}{J_0\pi}. \tag{4.6}$$

In the case where $\delta_\pm$ are real, we can rewrite the wave functions as

$$\psi_+(r) = \cos[\delta_+ \ln(|r|)], \tag{4.7}$$

$$\psi_-(r) = \mathrm{sgn}(r)\cos[\delta_- \ln(|r|)]. \tag{4.8}$$

If we apply discrete scale transformations $r \longrightarrow \lambda_\pm r$, we obtain

$$\psi_+(r) \longrightarrow \cos[\delta_+ \ln(|r|) + \delta_+ \ln(\lambda_+)], \tag{4.9}$$

$$\psi_-(r) \longrightarrow \mathrm{sgn}(r)\cos[\delta_- \ln(|r|) + \delta_- \ln(\lambda_-)]. \tag{4.10}$$

We obtain discrete scale invariance (up to a minus sign) if

$$\lambda_+ = \exp(\pi/\delta_+), \quad \lambda_- = \exp(\pi/\delta_-). \tag{4.11}$$

The energies scale as $E\pm \longrightarrow \lambda_\pm^{-1} E_\pm$.

4.1.2 Time Fractals

We consider a state that is a linear combination of the first N states from Eq. 4.2,

$$|S\rangle = \sum_{n=0}^{N-1} |\psi_+^{(n)}\rangle. \tag{4.12}$$

We calculate the amplitude

$$A(t) = \mathrm{Re}[\langle S|\exp(-iHt)|S\rangle] \tag{4.13}$$

$$A(t) = \sum_{n=0}^{N-1} \cos(E_+^{(n)} t) \tag{4.14}$$

$$A(t) = \sum_{n=0}^{N-1} \cos(\epsilon_+ \lambda_+^{-n} t). \tag{4.15}$$

We observe that $A(t)$ is invariant under the rescaling $t \longrightarrow \lambda_+ t$, thus it is a time fractal. We plot $A(t)$ in Fig. 4.1 for $J_0 = -1$ and $V_0 = -14.2388293$ so that $\lambda_+ = 2$. We also observe that $A(t)$ are closely related to the famous continous everywhere but differentiable nowhere Weierstrass function,

$$w(x) = \sum_{n=0}^{\infty} a^n \cos(2\pi b^n x). \tag{4.16}$$

Fractal dimension is a way of quantifying the scaling of a system. Similarly to how a two-dimensional square has area proportional to the 2nd power of its side length and a 3-dimensional cube has volume proportional to the 3rd power of its side length, an object with fractal dimension D has some size proportional to the Dth dimension of some parameter. The Weierstrass function has fractal dimension

$$D = 2 + \frac{\log a}{\log b}, \tag{4.17}$$

for several different definitions of fractal dimension [39]. Our amplitude $A(t)$, in the limit that $N \longrightarrow \infty$, becomes the Weierstrass function where $a = 1$ and $b = \lambda_+$. This gives our amplitude a fractal dimension of 2. I was able to numerically estimate the fractal dimension to be about 1.9.

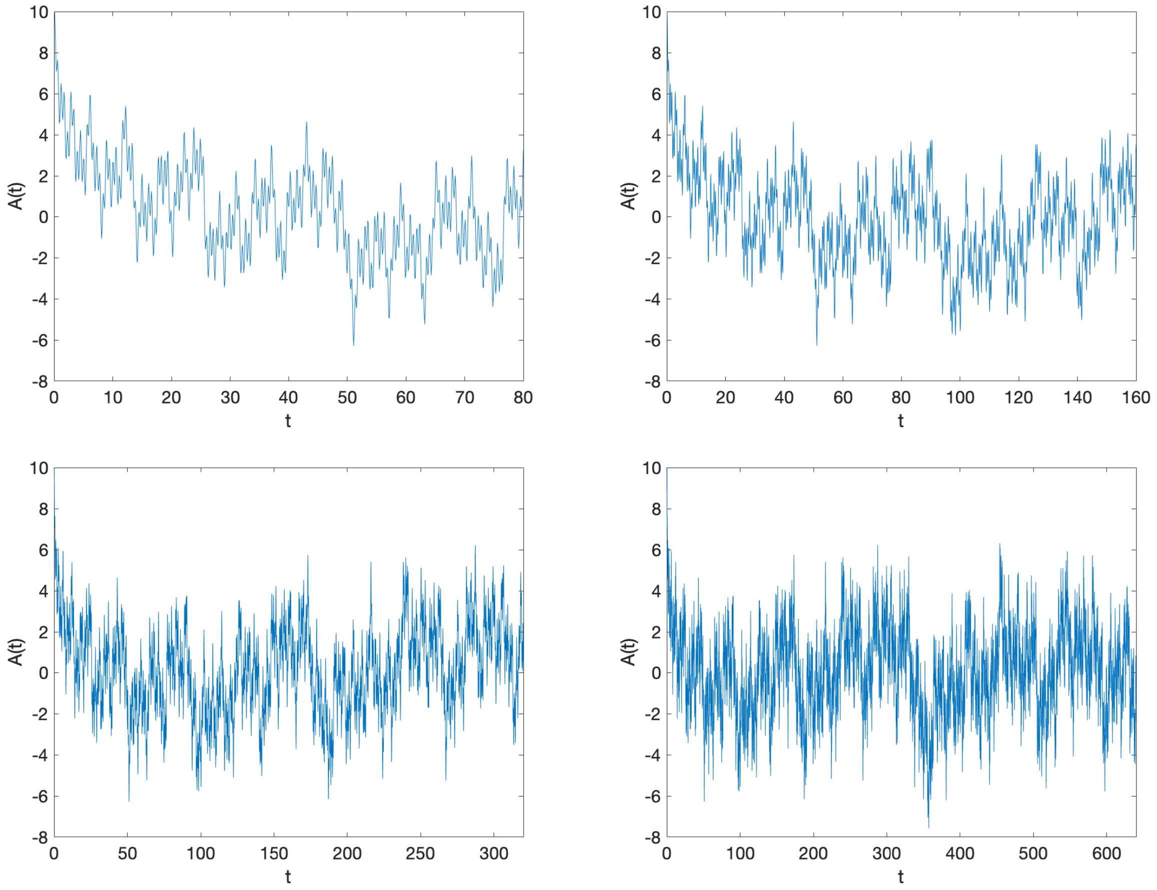


Figure 4.1 An example of time fractals. The amplitude $A(t)$ is displayed over the range from $t = 0$ to 80 in the upper left, $t = 0$ to 160 in the upper right, $t = 0$ to 320 in the lower left, and $t = 0$ to 640 in the lower right. When the time displayed doubles, the overall shape of the amplitude remains the same.

4.2 Projected Cooling

The projected cooling algorithm, based on the principles of evaporative cooling, can be used on a quantum computer to prepare the localized ground state of any Hamiltonian with a translationally-invariant kinetic energy and interactions that vanish at long distances [26].

Suppose we have a one-dimensional chain of $2L + 1$ qubits and a particle number-preserving Hamiltonian defined on it. We use the state $|1\rangle$ to represent that a qubit is occupied by a particle. We define a compact region ρ to be $2R + 1$ qubits where $R \ll L$. We define a projection operator P as the product of $|0\rangle\langle 0|$ over all qubits outside ρ (and the identity inside ρ). P projects onto the subspace where all particles are within ρ.

Suppose that our Hamiltonian's ground state $|\psi_0\rangle$ is a localized bound state, and the only bound state of H. Let $|\psi_I\rangle$ be some initial state contained in ρ that has nonzero overlap with $|\psi_0\rangle$. The time evolution oeprator is

$$U(t) = e^{-iHt}. \tag{4.18}$$

In the limit $L \to \infty$, $PU(t)P$ has a stable fixed point proportional to $P\,|\psi_0\rangle$. This means that over time, the excited states leave the region ρ and are killed off by the projection operator. So we are left with the ground state.

The principle behind projected cooling can be illustrated by considering a Gaussian wave packet. Suppose that at time 0, we have a Gaussian wave packet centered on the origin with width $\sigma = \sqrt{a}$. The standard formula for wave packet spreading gives that $\sigma(t) = \sqrt{\frac{a^2+(t/m)^2}{a}}$. At time t, the fraction of the wave packet contained in our region ρ is given by

$$f(t) = \frac{1}{\sigma(t)\sqrt{2\pi}} \int_{-R}^{R} e^{-\frac{1}{2}\left(\frac{x}{\sigma(t)}\right)^2} \mathrm{d}x. \tag{4.19}$$

Performing a u-substitution with $u = \frac{1}{\sqrt{2}}\left(\frac{x}{\sigma(t)}\right)$, we obtain a result in terms of the error function,

$$f(t) = \mathrm{erf}\left(\frac{R}{\sqrt{2\frac{a^2+(t/m)^2}{a}}}\right). \tag{4.20}$$

This allows us to normalize the Gaussian wave function truncated to within the region ρ:

$$\psi(x) = \begin{cases} 0 & |x| > R \\ \frac{1}{\pi^{1/4}\sqrt{\sigma \mathrm{erf}(R/\sigma)}} e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2} & |x| \le R \end{cases}. \tag{4.21}$$

We compute the overlap of this wave function with the ground state wave function of a particle in an infinite square well,

$$\psi(x) = \begin{cases} 0 & |x| > R \\ \sqrt{\frac{1}{R}} \cos\left(\frac{\pi}{2R}x\right) & |x| \le R \end{cases}. \tag{4.22}$$

The overlap is

$$O(t) = \frac{e^{-\frac{\pi^2\sigma^2}{8R^2}} \pi^{1/4} \sigma \left(\mathrm{erf}\left[\frac{R}{\sqrt{2}\sigma} - \frac{i\pi\sigma}{2\sqrt{2}R}\right] + \left[\mathrm{erf}\frac{R}{\sqrt{2}\sigma} + \frac{i\pi\sigma}{2\sqrt{2}R}\right]\right)}{\sqrt{2R\sigma \mathrm{erf}(R/\sigma)}}. \tag{4.23}$$

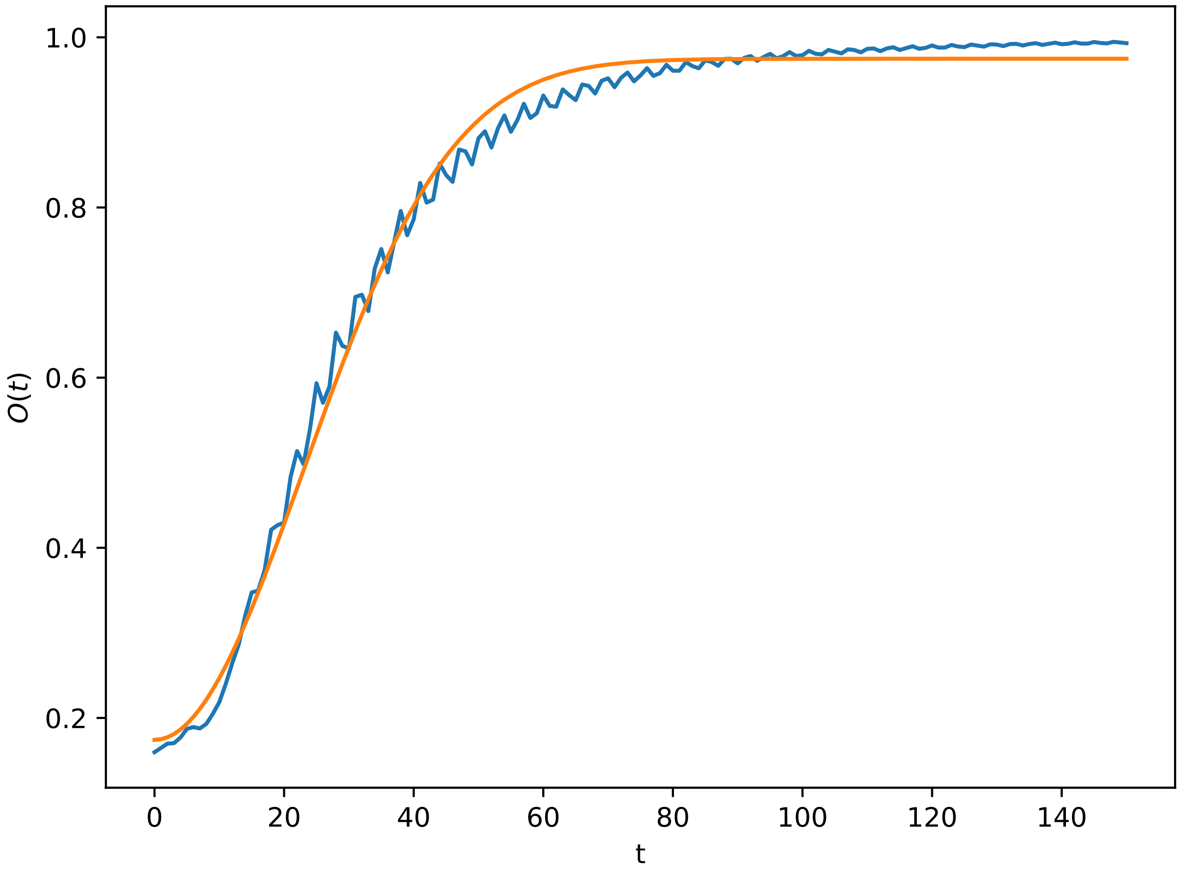


Figure 4.2 A comparison of my Gaussian wave packet model to projected cooling algorithm simulations. The ground state overlap computed in [26] is in blue, and the overlap between a truncated Gaussian wave packet and a particle-in-a-box wave function is in orange.

In Fig. 4.2, I compare this model to the projected cooling algorithm simulations in Ref. [26]. I used a series expansion to eliminate the imaginary terms and fit my model to the simulated data. The shapes of the functions are in general agreement.

4.3 Rodeo Algorithm

The rodeo algorithm is a method of eigenstate preparation and eigenvalue estimation on a quantum computer. It is exponentially faster than phase estimation and adiabatic evolution [35, 14]. Our initial work was with one-qubit Hamiltonians, but the Rodeo algorithm generalizes easily to larger Hamiltonians.

One cycle of the (single-qubit) rodeo algorithm is illustrated in Fig. 4.3 . The object qubit starts in some state $|\psi_0\rangle$ with nonzero overlap with the ground state. Then N cycles of the Rodeo algorithm are run, with t_k chosen from a Gaussian distribution centered at 0 with a specified width σ. Within each cycle, there is controlled time evolution of the Hamiltonian for time t_k and a phase rotation applied to the ancilla qubit with a parameter E that we choose. The probability that the ancilla qubit is still in the $|0\rangle$ state after one cycle of the Rodeo Algorithm is

$$P_0(E, t_k) = \cos^2\left[\frac{t_k}{2}(E_{\text{obj}} - E)\right], \tag{4.24}$$

where E_{obj} is the ground state energy of the object Hamiltonian. Performing N cycles of the Rodeo Algorithm (each with a different t_k but the same E) and marginalizing over all possible values of t_k gives that the probability that every measurement of the ancilla qubit results in $|0\rangle$ is

$$P_{0^N}(E) = \left[\frac{1 + e^{-(E_{\text{obj}}-E)^2\sigma^2/2}}{2}\right]^N. \tag{4.25}$$

We see that the probability goes exponentially to zero for $E \neq E_{\text{obj}}$, and is one for $E = E_{\text{obj}}$. Intuitively, this is similar to interference in interferometry: when $E \neq E_{\text{obj}}$ there is destructive interference between the controlled time evolution and phase shift.

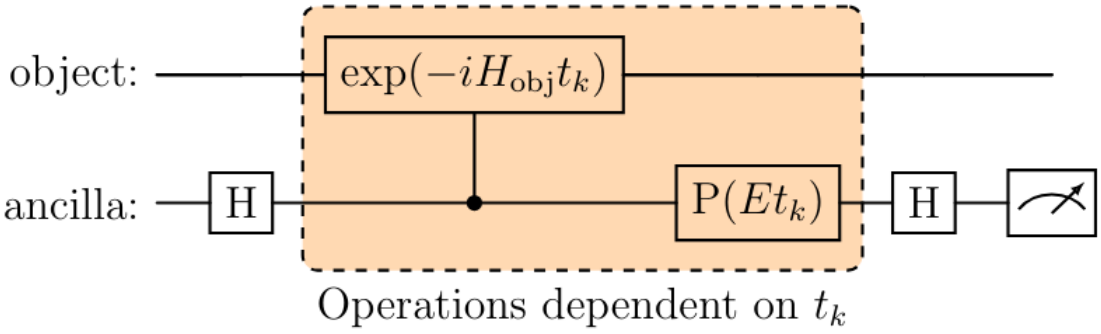


Figure 4.3 One cycle of the rodeo algorithm. In general the object Hamiltonian can have multiple qubits. The object qubit is initialized to some state $|\psi_0\rangle$, and the ancilla qubit is initialized to $|0\rangle$.

Since the probability of measuring the ancilla in the $|0\rangle$ state decays exponentially, if we make a histogram of successful measurements for a variety of guesses E, we expect to find peaks at the eigenvalues of H_{obj}. These peaks are robust to quantum computing errors: peaks will be lowered and the counts away from peaks will be increased, but the locations of the peaks will be unchanged.

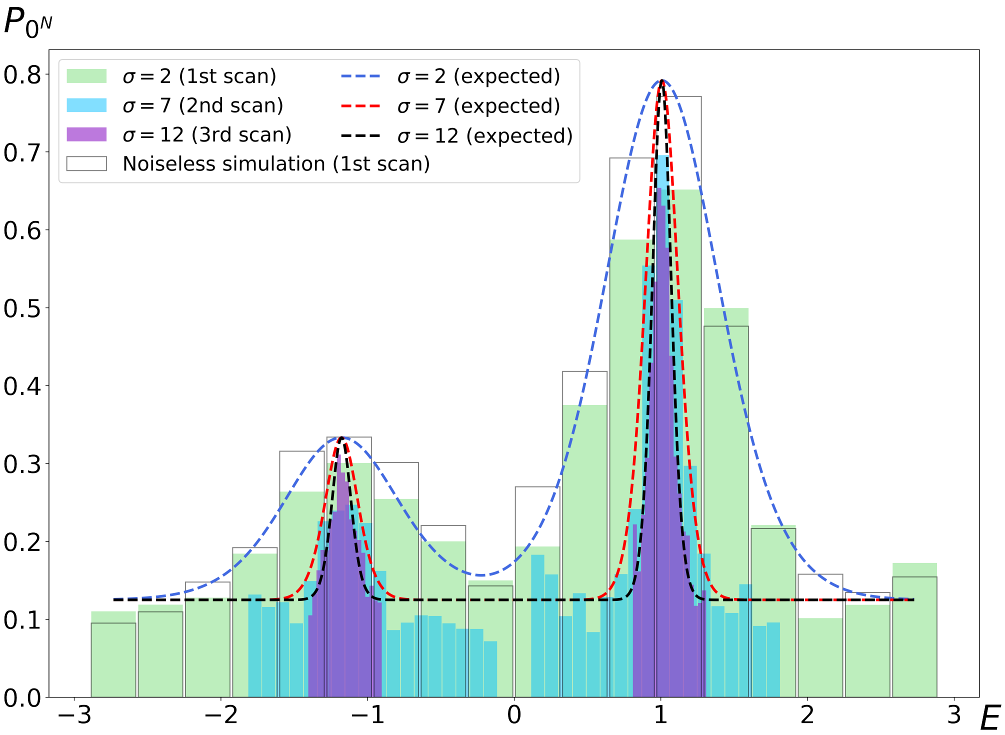


Figure 4.4 Results from running the Rodeo Algorithm on IBM Q Casablanca. The eigenenergies can be extracted from the centers of the peaks.

I determined that IBM's at the time recent introduction of mid-circuit measurements allowed the Rodeo Algorithm to be run on actual quantum hardware without requiring an additional ancilla qubit for each cycle. We prepared the single-qubit Rodeo Algorithm and ran it on IBM Q Casablanca [35]. The results are shown in Fig. 4.4. We performed three scans over E, with increasing resolution as we located the peaks. The peaks are somewhat lower than predicted by theory and noiseless simulations, due to errors inherent in current quantum hardware. We were able to determine the energy eigenstates within a relative error of 0.08%.

4.4 Quantum Monte Carlo and Superfluidity

I have also been involved with studies of superfluidity in neutron matter. I attempted to use quantum Monte Carlo methods to look for pairing effects indicative of simultaneous p-wave and s-wave superfluidity. I ran into some difficulties, as the particles appeared to enter bound states. Other people are continuing research into coexistence of superfluid states using auxiliary-field lattice simulations.

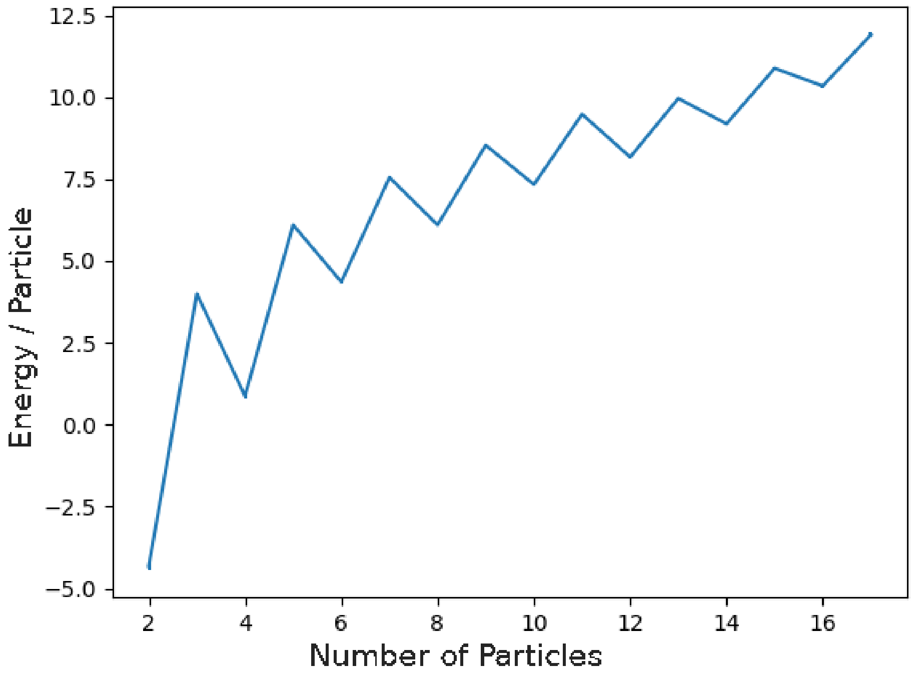


Figure 4.5 Results of Diffusion Monte Carlo simulation of fermions with an s-wave Pöschl-Teller interaction and a BCS wave function. There are pairs of up- and down-spin particles, with an additional up-spin for odd numbers of particles. We observe that completing a pair lowers the energy per particle.

Superfluid systems exhibit pairing similar to Cooper pairs in superconductors, and quantum Monte Carlo simulations are able to observe this pairing [9]. Superfluidity in neutron matter is important because it occurs inside neutron stars. It is believed that there is s-wave pairing superfluidity in the low-density inner crust, and p-wave pairing in higher-density regions deeper inside neutron stars [31, 22, 50]. Neutron pairing is also relevant in neutron-rich nuclei [21].

We are interested in whether s-wave pairing and p-wave pairing can coexist, perhaps in an intermediate-density region inside neutron stars. We are able to observe s-wave pairing using a Pöschl-Teller interaction, $V(r) = 1/\cosh^2(r)$, in quantum Monte Carlo simulations, see Fig. 4.5. If we have only spin-up neutrons, we are able to observe p-wave pairing, shown in Fig. 4.6.

To perform these calculations, we first run Variational Monte Carlo [2]. This uses the variational principle to tune a trial wave function and obtain an upper bound for the ground state energy. If we

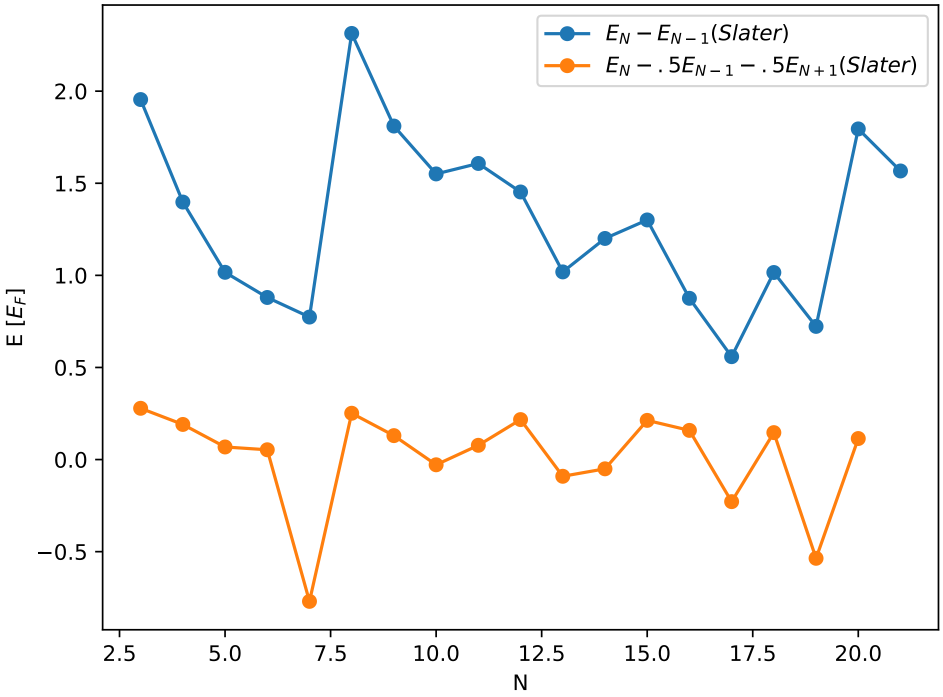


Figure 4.6 Results of quantum Monte Carlo simulation of fermions with a p-wave Pöschl-Teller interaction and a Slater determinant wave function. All particles are spin-up. The plots show the differences between energies of adjacent numbers of particles. There appears to be pairing at large numbers of particles. The minima are at odd N because the zero-momentum state is unpaired. We also observe shell closures at $N = 7$ and 19.

know the Hamiltonian and ground state wave function Ψ_0, we can calculate the energy:

$$E_0 = \frac{\int d\mathbf{R}\Psi_0^*(\mathbf{R})\hat{H}\Psi_0(\mathbf{R})}{\int d\mathbf{R}|\Psi_0(\mathbf{R})|^2}. \tag{4.26}$$

Calculating this integral analytically or even numerically is impractical, as for N particles there are $3N$ spatial coordinates, so we use Monte Carlo integration. We interpret the wave function as a probability distribution,

$$P(\mathbf{R}) = \frac{\Psi^*(\mathbf{R})\Psi(\mathbf{R})}{\int d\mathbf{R}|\Psi(\mathbf{R})|^2}, \tag{4.27}$$

and define a local energy

$$E_L(\mathbf{R}) = \Psi^{-1}(\mathbf{R})\hat{H}\Psi(\mathbf{R}). \tag{4.28}$$

So we can rewrite the energy as

$$E = \int d\mathbf{R}E_L(\mathbf{R})P(\mathbf{R}). \tag{4.29}$$

If we have a set $\{\mathbf{R}_i\}$ distributed according to $P(\mathbf{R})$, then by the central limit theorem

$$E \approx \frac{1}{M} \sum_{i=1}^{M} E_L(\mathbf{R}_i). \tag{4.30}$$

We sample configurations $\mathbf{R}$ according to the probability distribution $P(\mathbf{R})$ using the Metropolis algorithm. We construct a Markov chain with transition probabilities $T(\mathbf{R}' \to \mathbf{R})$ that satisfy detailed balance,

$$P(\mathbf{R})T(\mathbf{R}' \to \mathbf{R}) = P(\mathbf{R}')T(\mathbf{R} \to \mathbf{R}'), \tag{4.31}$$

ensuring that the equilibrium of the Markov chain is $P(\mathbf{R})$. If the probability of selecting a particular configuration update is $\Lambda(\mathbf{R}' \to \mathbf{R})$, then the acceptance rate (the probability that we accept the update) is

$$A(\mathbf{R}' \to \mathbf{R}) = \min\left[1, \frac{P(\mathbf{R})\Lambda(\mathbf{R} \to \mathbf{R}')}{P(\mathbf{R}')\Lambda(\mathbf{R}' \to \mathbf{R})}\right]. \tag{4.32}$$

We can choose $\Lambda(\mathbf{R} \to \mathbf{R}')$ to be symmetrical, such as by choosing a uniform or Gaussian distribution, causing the acceptance rate to be calculated solely from the wave function. We don't know the exact ground state wave function, so we choose a wave function with parameters we can vary. Then by the variational principle, we obtain upper bounds of the ground state energy,

$$E_V = \frac{\int d\mathbf{R}\Psi_T^*(\mathbf{R})\hat{H}\Psi_T(\mathbf{R})}{\int d\mathbf{R}|\Psi_T(\mathbf{R})|^2} \geq E_0. \tag{4.33}$$

As a result of performing Variational Monte Carlo, we also obtain a set of particle configurations that samples the trial wave function. We use this as the initial state for Diffusion Monte Carlo (with importance sampling), a numerical method to solve the imaginary time Schrödinger equation which is a combination of a diffusion process and a branching process [37]. The main idea behind Diffusion Monte Carlo is the imaginary time evolution,

$$\begin{aligned} \Psi(\tau) &= e^{-(\hat{H}-E_T)\tau}\Psi_T & (4.34) \\ &= \sum_i c_i e^{-(E_i-E_T)\tau}\psi_i, & (4.35) \end{aligned}$$

where we expand in eigenstates of the Hamiltonian. If the trial energy $E_T \approx E_0$, the ground state is projected out at large τ, as the other eigenstates vanish.

For each iteration of Diffusion Monte Carlo, the particle positions in each configuration are updated, and then the configurations are branched: configurations with low energies are duplicated, while configurations with high energies are removed. The position update includes a random term and a drift velocity term that moves the particles to regions with a large trial wave function. The moves are accepted or rejected using the Metropolis algorithm to satisfy detailed balance. The branching process duplicates and removes configurations according to their energies. Then the ground state energy is updated and the process is repeated.

Diffusion Monte Carlo can in principle find the exact ground state energy. However, since fermion wave functions have both positive and negative regions, we avoid the fermion sign problem with the fixed node approximation which prohibits configurations from moving to a region with opposite sign. This means that the quality of the result depends on the choice of nodes of the trial wave function, and we can only obtain an upper bound for the ground state energy.

CHAPTER 5

CONCLUSIONS AND OUTLOOK

There is a trade-off between number of qubits required and number of quantum gates required for fermion to qubit mappings. As we saw, for two-dimensional simulations, the $\eta\chi$ method requires 1.25 qubits per fermionic mode (lattice site), and up to 10 Pauli operators for each hopping term in the Hamiltonian. My composite lattice modification increases the number of qubits to 2 per mode, but decreases the maximum number of Pauli operators per hopping term to 4.

The canonical circuit for exponentiation of a Pauli string of length n requires $2(n-1)$ two-qubit gates. I showed, using results from Ref. [42], that terms of the form $(XX + YY)ZZ$, which are pairs of Pauli strings of length n, can be exponentiated using $2(n - 1)$ 2-qubit gates. This is half the number required for the canonical circuit.

My motivation for obtaining these improvements is the limitations of current quantum computers. Gate error rates and decoherence time are two of the current limitations to quantum computation, and two-qubit gates have a larger computational cost for both of these than single-qubit gates. Thus I have sought to reduce the number of two-qubit gates required to simulate nuclear physics.

Current nuclear lattice effective field theory simulations are limited by the amount of memory available on classical computers. I hope that my work will help enable larger systems to be studied using quantum computers once the hardware is sufficiently developed.

BIBLIOGRAPHY

[1] M Bajdich et al. “Pfaffian pairing wave functions in electronic-structure quantum Monte Carlo simulations”. In: *Physical review letters* 96.13 (2006), p. 130201.

[2] Michal Bajdich. *Generalized Pairing Wave Functions and Nodal Properties for Electronic Structure Quantum Monte Carlo*. 2007. arXiv: `0712.3066 [cond-mat.other]`. URL: `https://arxiv.org/abs/0712.3066`.

[3] Max Bee-Lindgren et al. *Controlled Gate Networks Applied to Eigenvalue Estimation*. 2024. arXiv: `2208.13557 [quant-ph]`. URL: `https://arxiv.org/abs/2208.13557`.

[4] P.Oscar Boykin et al. “A new universal and fault-tolerant quantum basis”. In: *Information Processing Letters* 75.3 (2000), pp. 101–107. ISSN: 0020-0190. DOI: `https://doi.org/10.1016/S0020-0190(00)00084-3`. URL: `https://www.sciencedirect.com/science/article/pii/S0020019000000843`.

[5] Sergey B Bravyi and Alexei Yu Kitaev. “Fermionic quantum computation”. In: *Annals of Physics* 298.1 (2002), pp. 210–226.

[6] Jacob Bringewatt and Zohreh Davoudi. “Parallelization techniques for quantum simulation of fermionic systems”. In: *Quantum* 7 (2023), p. 975.

[7] Nicholas Cariello. personal communication. Oct. 2024.

[8] Davide Castelvecchi. “IBM releases first-ever 1,000-qubit quantum chip”. In: *Nature* 624.7991 (2023), pp. 238–238.

[9] Soon-Yong Chang et al. “Quantum Monte Carlo studies of superfluid Fermi gases”. In: *Physical Review A—Atomic, Molecular, and Optical Physics* 70.4 (2004), p. 043602.

[10] Yu-An Chen and Yijia Xu. “Equivalence between fermion-to-qubit mappings in two spatial dimensions”. In: *PRX Quantum* 4.1 (2023), p. 010326.

[11] Riley W Chien and James D Whitfield. “Custom fermionic codes for quantum simulation”. In: *arXiv preprint arXiv:2009.11860* (2020).

[12] Andrew M Childs and Nathan Wiebe. “Hamiltonian simulation using linear combinations of unitary operations”. In: *arXiv preprint arXiv:1202.5822* (2012).

[13] Andrew M Childs et al. “Theory of trotter error with commutator scaling”. In: *Physical Review X* 11.1 (2021), p. 011020.

[14] Kenneth Choi et al. “Rodeo algorithm for quantum computing”. In: *Physical Review Letters* 127.4 (2021), p. 040505.

[15] Richard Cleve et al. “Quantum algorithms revisited”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (1998), pp. 339–354.

[16] Matthew DeCross et al. “The computational power of random quantum circuits in arbitrary geometries”. In: *arXiv preprint arXiv:2406.02501* (2024).

[17] Charles Derby et al. “Compact fermion to qubit mappings”. In: *Physical Review B* 104.3 (July 2021). ISSN: 2469-9969. DOI: 10.1103/physrevb.104.035118. URL: http://dx.doi.org/10.1103/PhysRevB.104.035118.

[18] Edward Farhi et al. *Quantum Computation by Adiabatic Evolution*. 2000. arXiv: quant-ph/0001106 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0001106.

[19] Richard P Feynman. “Simulating Physics with Computers”. In: *International Journal of Theoretical Physics* 21.6/7 (1982).

[20] Keisuke Fujii et al. “Deep variational quantum eigensolver: a divide-and-conquer method for solving a larger problem with smaller size quantum computers”. In: *PRX Quantum* 3.1 (2022), p. 010346.

[21] Stefano Gandolfi, Alexandros Gezerlis, and Joe Carlson. “Neutron matter from low to high density”. In: *Annual Review of Nuclear and Particle Science* 65.1 (2015), pp. 303–328.

[22] A. Gezerlis, C. J. Pethick, and A. Schwenk. *Pairing and superfluidity of nucleons in neutron stars*. 2015. arXiv: 1406.6109 [nucl-th]. URL: https://arxiv.org/abs/1406.6109.

[23] Stuart Hadfield et al. “From the quantum approximate optimization algorithm to a quantum alternating operator ansatz”. In: *Algorithms* 12.2 (2019), p. 34.

[24] Dean Lee. “Lattice simulations for few-and many-body systems”. In: *Progress in Particle and Nuclear Physics* 63.1 (2009), pp. 117–154.

[25] Dean Lee. “Recent progress in nuclear lattice simulations”. In: *Frontiers in Physics* 8 (2020), p. 174.

[26] Dean Lee et al. “Projected cooling algorithm for quantum computation”. In: *Physics Letters B* 807 (2020), p. 135536.

[27] Dean Lee et al. “Time fractals and discrete scale invariance with trapped ions”. In: *Physical Review A* 100.1 (2019), p. 011403.

[28] Kangle Li and Hoi Chun Po. “Higher-dimensional Jordan-Wigner transformation and auxiliary Majorana fermions”. In: *Phys. Rev. B* 106 (11 Sept. 2022), p. 115109.

DOI: 10.1103/PhysRevB.106.115109. URL: https://link.aps.org/doi/10.1103/PhysRevB.106.115109.

[29] Jannes Nys and Giuseppe Carleo. “Quantum circuits for solving local fermion-to-qubit mappings”. In: *Quantum* 7 (Feb. 2023), p. 930. ISSN: 2521-327X. DOI: 10.22331/q-2023-02-21-930. URL: https://doi.org/10.22331/q-2023-02-21-930.

[30] Jannes Nys and Giuseppe Carleo. “Variational solutions to fermion-to-qubit mappings in two spatial dimensions”. In: *Quantum* 6 (2022), p. 833.

[31] Georgios Palkanoglou and Alexandros Gezerlis. “Superfluid neutron matter with a twist”. In: *Universe* 7.2 (2021), p. 24.

[32] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature communications* 5.1 (2014), p. 4213.

[33] Hoi Chun Po. *Symmetric Jordan-Wigner transformation in higher dimensions*. 2021. arXiv: 2107.10842 [cond-mat.str-el].

[34] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79. URL: https://doi.org/10.22331/q-2018-08-06-79.

[35] Zhengrong Qian et al. “Demonstration of the rodeo algorithm on a quantum computer”. In: *arXiv preprint arXiv:2110.07747* (2021).

[36] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.

[37] Peter J Reynolds et al. “Fixed-node quantum Monte Carlo for moleculesa) b)”. In: *The Journal of Chemical Physics* 77.11 (1982), pp. 5593–5603.

[38] Jacob T Seeley, Martin J Richard, and Peter J Love. “The Bravyi-Kitaev transformation for quantum computation of electronic structure”. In: *The Journal of Chemical Physics* 137.22 (2012).

[39] Weixiao Shen. “Hausdorff dimension of the graphs of the classical Weierstrass functions”. In: *Mathematische Zeitschrift* 289 (2018), pp. 223–266.

[40] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172.

[41] John C Slater. “The theory of complex spectra”. In: *Physical review* 34.10 (1929), p. 1293.

[42] Adam Smith et al. “Simulating quantum many-body dynamics on a current digital quantum computer”. In: *npj Quantum Information* 5.1 (2019), p. 106.

[43] Matthias Steffen et al. “Quantum computing: An IBM perspective”. In: *IBM Journal of Research and Development* 55.5 (2011), pp. 13–1.

[44] Mark Steudtner and Stephanie Wehner. “Quantum codes for quantum simulation of fermions on a square lattice of qubits”. In: *Physical Review A* 99.2 (2019), p. 022308.

[45] Andrew Tranter et al. “The Bravyi–Kitaev transformation: Properties and applications”. In: *International Journal of Quantum Chemistry* 115.19 (2015), pp. 1431–1441.

[46] Farrokh Vatan and Colin Williams. “Optimal quantum circuits for general two-qubit gates”. In: *Physical Review A—Atomic, Molecular, and Optical Physics* 69.3 (2004), p. 032315.

[47] F Verstraete and J I Cirac. “Mapping local Hamiltonians of fermions to local Hamiltonians of spins”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.09 (Sept. 2005), P09012. DOI: 10.1088/1742-5468/2005/09/P09012. URL: https://dx.doi.org/10.1088/1742-5468/2005/09/P09012.

[48] James D Watson et al. “Quantum Algorithms for Simulating Nuclear Effective Field Theories”. In: *arXiv preprint arXiv:2312.05344* (2023).

[49] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. “Simulation of electronic structure Hamiltonians using quantum computers”. In: *Molecular Physics* 109.5 (2011), pp. 735–750. DOI: 10.1080/00268976.2011.552441.

[50] Shigehiro Yasui, Chandrasekhar Chatterjee, and Muneto Nitta. “Phase structure of neutron P 2 3 superfluids in strong magnetic fields in neutron stars”. In: *Physical Review C* 99.3 (2019), p. 035213.