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 v

CHAPTER 1 INTRODUCTION 1
1.1 Lattice Effective Field Theory 1
1.2 Quantum Circuits 3

CHAPTER 2 FERMION TO QUBIT MAPPINGS 7
2.1 Jordan-Wigner Transformation 7
2.2 Parity Basis 8
2.3 General Linear Transformation Mappings 9
2.4 Auxiliary Qubit Mappings 11
2.5 Majorana Fermion Methods 11
2.6 Equivalence of Fermion to Qubit Mappings 13

CHAPTER 3 THE $\eta\chi$ METHOD 14
3.1 The $\eta\chi$ Method 14
3.2 Exponentiation of $XX + YY$ 22
3.3 Single-particle creation 23
3.4 Using Perturbation Theory 24
3.5 Perturbative Term 28
3.6 Implementation of the $\eta\chi$ Method 30

CHAPTER 4 CONTRIBUTIONS TO VARIOUS PAPERS 31
4.1 Scale Invariance and Time Fractals 31
4.2 Projected Cooling 32
4.3 Rodeo Algorithm 34

CHAPTER 5 CONCLUSIONS AND OUTLOOK 38

BIBLIOGRAPHY 39

LIST OF ABBREVIATIONS

CNOT Controlled Not Gate

CZ Controlled Z Gate

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 Lattice Effective Field Theory

Nuclear lattice effective field theory is an established method for simulating nuclear matter [9, 10]. Lattice quantum chromodynamics 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

$$H_{\rm LO} = H_{\rm free} + V + V_{I^2} + V^{(3N)}. \tag{1.1}$$

The free part of the Hamiltonian is

$$H_{\rm free} = \frac{1}{2m}\sum_{\sigma,\tau}\int {\rm d}^3\mathbf{r}\,\nabla a^\dagger_{\sigma\tau}(\mathbf{r})\cdot\nabla a_{\sigma\tau}(\mathbf{r}), \tag{1.2}$$

which when discretized for a lattice simulation becomes

$$H_{\rm free} = \frac{3}{m}\sum_{\mathbf{n},\sigma,\tau} a^\dagger_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}) - \frac{1}{2m}\sum_{\mathbf{n},\sigma,\tau}\sum_{l=x,y,z}\left[a^\dagger_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}+\mathbf{e}_l) + a^\dagger_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}-\mathbf{e}_l)\right]. \tag{1.3}$$

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

$$H_{\rm free} = \frac{3}{m}\sum_{\mathbf{n},\sigma,\tau} a^\dagger_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}) - \frac{1}{2m}\sum_{\mathbf{n},\sigma,\tau}\sum_{l=x,y,z}\left[a^\dagger_{\sigma\tau}(\mathbf{n})a_{\sigma\tau}(\mathbf{n}+\mathbf{e}_l) + a^\dagger_{\sigma\tau}(\mathbf{n}+\mathbf{e}_l)a_{\sigma\tau}(\mathbf{n})\right], \tag{1.4}$$

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,

$$\rho(\mathbf{r}) = \sum_{\sigma,\tau} a^\dagger_{\sigma\tau}(\mathbf{r}) a_{\sigma\tau}(\mathbf{r}), \tag{1.5}$$

$$\rho_I(\mathbf{r}) = \sum_{\sigma,\tau,\tau'} a^\dagger_{\sigma\tau}(\mathbf{r}) [\tau_I]_{\tau\tau'} a_{\sigma\tau'}(\mathbf{r}), \tag{1.6}$$

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

$$V = \frac{C}{2} \int \mathrm{d}^3\mathbf{r} : [\rho(\mathbf{r})]^2 :, \tag{1.7}$$

$$V_{I^2} = \frac{C_{I^2}}{2} \int \mathrm{d}^3\mathbf{r} : [\rho_I(\mathbf{r})]^2 :, \tag{1.8}$$

$$V^{(3N)} = \frac{D}{6} \int \mathrm{d}^3\mathbf{r} : [\rho(\mathbf{r})]^3 : . \tag{1.9}$$

Recently, smeared creation and annihilation operators have been introduced to enforce conservation of orbital angular momentum [10]. 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

$$V_L = \frac{c_L}{2} \sum_{\mathbf{n}} : \rho_L(\mathbf{n}) \rho_L(\mathbf{n}) : + \frac{c_{S,L}}{2} \sum_{\mathbf{n},S} : \rho_{S,L}(\mathbf{n}) \rho_{S,L}(\mathbf{n}) : \tag{1.10}$$

$$+ \frac{c_{I,L}}{2} \sum_{\mathbf{n},I} : \rho_{I,L}(\mathbf{n}) \rho_{I,L}(\mathbf{n}) : + \frac{c_{S,I,L}}{2} \sum_{\mathbf{n},S,I} : \rho_{S,I,L}(\mathbf{n}) \rho_{S,I,L}(\mathbf{n}) :, \tag{1.11}$$

and the non-local portion is

$$V_{NL} = \frac{c_{NL}}{2} \sum_{\mathbf{n}} : \rho_{NL}(\mathbf{n}) \rho_{NL}(\mathbf{n}) : + \frac{c_{I,NL}}{2} \sum_{\mathbf{n},I} : \rho_{I,NL}(\mathbf{n}) \rho_{I,NL}(\mathbf{n}) :, \tag{1.12}$$

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

defined as follows:

$$\rho_{NL}(\mathbf{n}) = a^\dagger_{NL}(\mathbf{n}) a_{NL}(\mathbf{n}), \tag{1.13}$$

$$\rho_{I,NL}(\mathbf{n}) = a^\dagger_{NL}(\mathbf{n}) [\tau_I] a_{NL}(\mathbf{n}), \tag{1.14}$$

$$\rho_{L}(\mathbf{n}) = a^\dagger(\mathbf{n}) a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n}'\mathbf{n} \rangle} a^\dagger(\mathbf{n}') a(\mathbf{n}'), \tag{1.15}$$

$$\rho_{S,L}(\mathbf{n}) = a^\dagger(\mathbf{n}) [\sigma_S] a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n}'\mathbf{n} \rangle} a^\dagger(\mathbf{n}') [\sigma_S] a(\mathbf{n}'), \tag{1.16}$$

$$\rho_{I,L}(\mathbf{n}) = a^\dagger(\mathbf{n}) [\tau_I] a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n}'\mathbf{n} \rangle} a^\dagger(\mathbf{n}') [\tau_I] a(\mathbf{n}'), \tag{1.17}$$

$$\rho_{S,I,L}(\mathbf{n}) = a^\dagger(\mathbf{n}) [\sigma_S \otimes \tau_I] a(\mathbf{n}) + s_L \sum_{\langle \mathbf{n}'\mathbf{n} \rangle} a^\dagger(\mathbf{n}') [\sigma_S \otimes \tau_I] a(\mathbf{n}'), \tag{1.18}$$

where the sum over $\langle \mathbf{n}'\mathbf{n} \rangle$ 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

$$a_{NL}(\mathbf{n}) = a(\mathbf{n}) + s_{NL} \sum_{\langle \mathbf{n}'\mathbf{n} \rangle} a(\mathbf{n}'), \tag{1.19}$$

$$a^\dagger_{NL}(\mathbf{n}) = a^\dagger(\mathbf{n}) + s_{NL} \sum_{\langle \mathbf{n}'\mathbf{n} \rangle} a^\dagger(\mathbf{n}'). \tag{1.20}$$

1.2 Quantum Circuits

In 1982, Richard Feynman proposed that suitable devices could by created to simulate any quantum system: quantum computers [8]. I wish to determine how to use this rapidly-improving technology for simulations of nuclear lattice effective field theory. IBM has recently introduced a quantum computer with over 1000 qubits [4], but error rates continue to be one of the primary challenges in quantum computing.

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,

$$a|0\rangle + b|1\rangle, \tag{1.21}$$

where a and b are complex numbers with $|a|^2 + |b|^2 = 1$.

Analogously to logic gates on classical computers, there are one- and two-qubit quantum gates. The three Pauli matrices, X, Y, and Z, are used as single-qubit gates. The Hadamard gate creates a superposition of the $|0\rangle$ and $|1\rangle$ states,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.22}$$

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

$$P(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}. \tag{1.23}$$

An important special case is the S gate,

$$S = P(\pi/2) = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{1.24}$$

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

$$R_z(\theta) = e^{-iZ\theta/2} = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \tag{1.25}$$

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

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 [18]. A controlled R_z gate is equivalent to the circuit shown in Fig. 1.2, composed of R_z gates and CNOT gates.

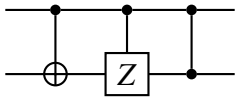


Figure 1.1 From left to right, a CNOT gate, CZ gate, and alternate method of displaying a CZ gate. The upper qubit is the control qubit.

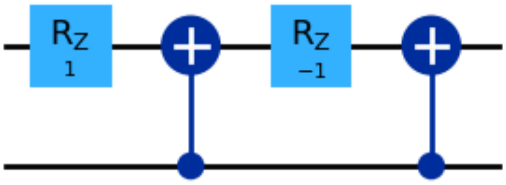


Figure 1.2 Decomposition of controlled $R_z(2)$ gate into R_z and CNOT gates. The lower qubit is the control qubit.

This circuit is an example of the utility of controlled reversal gates [1]. 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 [19]. 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.

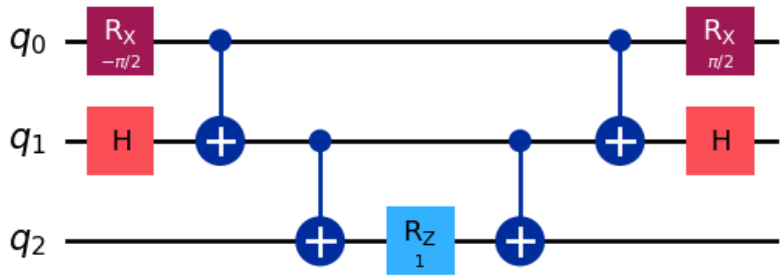


Figure 1.3 Circuit for $e^{-iY^{(0)}X^{(1)}Z^{(2)}/2}$. This circuit can be controlled by controlling the R_z gate.

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 [6],

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

along with the circuit from Fig. 1.3 to exponentiate the Pauli strings to create a quantum circuit to represent our Hamiltonian. I will primarily consider lattice effective theory in two dimensions.

2.1 Jordan-Wigner Transformation

The simplest method is the Jordan-Wigner transformation [25, 19]. 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 [2].

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 restore the parity terms, we must restore the anti-commutation relation of fermionic creation and annihilation operators,

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

We note that $\left\{\hat{Q}_j^{\pm}, Z_j\right\} = 0$, so

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

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

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.

For a hopping term on an L by L lattice, the horizontal hop will be efficient, $O(1)$, and the vertical hop will have a Pauli Z string of length $O(L)$.

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 [19].

This complicates the qubit creation and annihilation operators:

$$\hat{P}_j^{\pm} = |0\rangle\langle 0|_{j-1} \otimes \hat{Q}_j^{\pm} - |1\rangle\langle 1|_{j-1} \otimes \hat{Q}_j^{\mp} \tag{2.9}$$

$$= \frac{1}{2}(Z_{j-1} \otimes X_j \mp iY_j), \tag{2.10}$$

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. [21]. 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 [2, 19].

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

$$\boldsymbol{\omega} = A\,\boldsymbol{f}, \quad \boldsymbol{f} = A^{-1}\boldsymbol{\omega}, \tag{2.11}$$

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 such that

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

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 [2, 19, 22]. The transformation matrix A_{BK} has a binary tree structure. It can be recursively defined by

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

and

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

where $\leftarrow 1 \rightarrow$ represents a row of ones. Recursive formulae for the update, parity, and flip sets are given in Ref. [22]. Each set has Pauli weight $O(\log L)$, 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 [21].

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 [21]. It's not clear to me how much the need to update the auxiliary qubits offsets this advantage. Ref. [21] claims a Pauli string length of $O(1)$.

2.5 Majorana Fermion Methods

Several methods based on Majorana fermions also give constant scaling 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

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

2.5.1 Bravyi-Kitaev Superfast

Several methods put qubits on the edges of a graph. Bravyi-Kitaev Superfast is one of them [2, 21]. 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 [23]. 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 $ic_i d_j$ 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. [24]. 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.

2.6 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 [5]. 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. [13]. 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 [16]. 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.

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

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

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

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.

$$\Phi^{il} = -\Lambda^{ji}\Theta^{jk}\Lambda^{kl} \tag{3.8}$$

$$= -\Lambda^{ji} i\eta^j \eta^k i\eta^k \chi^l \tag{3.9}$$

$$= -i\Lambda^{ji} i\eta^j \chi^l \tag{3.10}$$

$$= -i\Lambda^{ji}\Lambda^{jl}. \tag{3.11}$$

We can rewrite our initial Majoranas (and hence 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. [15] 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.

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

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. [15], 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}$$

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

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,

$$f_{\uparrow,\mathbf{r}}^{p\dagger} f_{\uparrow,\mathbf{r+x}}^{n\dagger} = \frac{1}{2}(\gamma_{\mathbf{r}}^{1} + i\gamma_{\mathbf{r}}^{2})\frac{1}{2}(\gamma_{\mathbf{r+x}}^{5} + i\gamma_{\mathbf{r+x}}^{6}) \tag{3.25}$$

$$= \frac{1}{4}(-i\Lambda_{\mathbf{r}}^{14}\Lambda_{\mathbf{r+x}}^{53} + \Lambda_{\mathbf{r}}^{14}\Lambda_{\mathbf{r+x}}^{63} + \Lambda_{\mathbf{r}}^{24}\Lambda_{\mathbf{r+x}}^{53} + i\Lambda_{\mathbf{r}}^{24}\Lambda_{\mathbf{r+x}}^{63}) \tag{3.26}$$

$$= \frac{1}{4}(-i\Lambda_{\mathbf{r}}^{14}(\Lambda_{\mathbf{r+x}}^{53} + i\Lambda_{\mathbf{r+x}}^{63}) + \Lambda_{\mathbf{r}}^{24}(\Lambda_{\mathbf{r+x}}^{53} + i\Lambda_{\mathbf{r+x}}^{63})) \tag{3.27}$$

$$= \frac{1}{4}(-i\Lambda_{\mathbf{r}}^{14} + \Lambda_{\mathbf{r}}^{24})(\Lambda_{\mathbf{r+x}}^{53} + i\Lambda_{\mathbf{r+x}}^{63}) \tag{3.28}$$

$$= \frac{1}{4}(iX_{\mathbf{r}}^{(1)} + Y_{\mathbf{r}}^{(1)})(Y_{\mathbf{r+x}}^{(3)}Z_{\mathbf{r+x}}^{(4)}Z_{\mathbf{r+x}}^{(5)} + iX_{\mathbf{r+x}}^{(3)}Z_{\mathbf{r+x}}^{(4)}Z_{\mathbf{r+x}}^{(5)}) \tag{3.29}$$

$$= -Q_{\mathbf{r}}^{+(1)}Q_{\mathbf{r+x}}^{+(3)}Z_{\mathbf{r+x}}^{(4)}Z_{\mathbf{r+x}}^{(5)}. \tag{3.30}$$

To obtain the corresponding terms for other nucleon species, the first indices on the Λ operators are changed according to particle type, and 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

$$f_{\uparrow}^{p\dagger} f_{\uparrow}^{n\dagger} = \frac{1}{2}(\gamma^{1} + i\gamma^{2})\frac{1}{2}(\gamma^{5} + i\gamma^{6}) \tag{3.31}$$

$$= \frac{1}{4}(-i\Theta^{15} + \Theta^{16} + \Theta^{25} + i\Theta^{26}) \tag{3.32}$$

$$= \frac{1}{4}(-iY^{(1)}Z^{(2)}X^{(3)} - Y^{(1)}Z^{(2)}Y^{(3)} + X^{(1)}Z^{(2)}X^{(3)} - iX^{(1)}Z^{(2)}Y^{(3)}) \tag{3.33}$$

$$= \frac{1}{4}(-iY^{(1)}Z^{(2)}(X^{(3)} - iY^{(3)}) + X^{(1)}Z^{(2)}(X^{(3)} - iY^{(3)})) \tag{3.34}$$

$$= \frac{1}{4}(X^{(1)} - iY^{(1)})Z^{(2)}(X^{(3)} - iY^{(3)}) \tag{3.35}$$

$$= Q^{+(1)}Z^{(2)}Q^{+(3)}. \tag{3.36}$$

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.

$$f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}'} + f^{p\dagger}_{\uparrow,\mathbf{r}'} f^{p}_{\uparrow,\mathbf{r}} = \frac{1}{4}(\gamma^1_{\mathbf{r}} + i\gamma^2_{\mathbf{r}})(\gamma^1_{\mathbf{r}'} - i\gamma^2_{\mathbf{r}'}) + \frac{1}{4}(\gamma^1_{\mathbf{r}'} + i\gamma^2_{\mathbf{r}'})(\gamma^1_{\mathbf{r}} - i\gamma^2_{\mathbf{r}}) \tag{3.37}$$

$$= \frac{1}{2}(i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r}'} - i\gamma^1_{\mathbf{r}}\gamma^2_{\mathbf{r}'}) \tag{3.38}$$

$$= \frac{1}{2}(\Lambda^{2\mu}_{\mathbf{r}}\Lambda^{1\nu}_{\mathbf{r}'} - \Lambda^{1\mu}_{\mathbf{r}}\Lambda^{2\nu}_{\mathbf{r}'}) \tag{3.39}$$

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

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

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

A Hamiltonian with smearing (as in Ref. [10]) 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 also contain terms proportional to number operators:

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

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 [13]. The first type is plaquette constraints that arise from constructing an identity operator around a plaquette in the lattice:

$$
\Phi^{24}_{\mathbf{r}}\Phi^{32}_{\mathbf{r+x}}\Phi^{13}_{\mathbf{r+x+y}}\Phi^{41}_{\mathbf{r+y}} = -1
$$

$$
Z^{(1)}_{\mathbf{r}}Z^{(2)}_{\mathbf{r}}Z^{(3)}_{\mathbf{r}}Z^{(4)}_{\mathbf{r}}Y^{(5)}_{\mathbf{r}}X^{(5)}_{\mathbf{r+x}}Y^{(5)}_{\mathbf{r+x+y}}Z^{(1)}_{\mathbf{r+y}}Z^{(2)}_{\mathbf{r+y}}Z^{(3)}_{\mathbf{r+y}}Z^{(4)}_{\mathbf{r+y}}X^{(5)}_{\mathbf{r+y}} = 1
$$

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,

$$\Phi_{\mathbf{r}}^{12}\Phi_{\mathbf{r+y}}^{12}\ldots\Phi_{\mathbf{r-y}}^{12} = -1, \tag{3.41}$$

$$\Phi_{\mathbf{r}}^{34}\Phi_{\mathbf{r+x}}^{34}\ldots\Phi_{\mathbf{r-x}}^{34} = -1. \tag{3.42}$$

This constraint can be applied to the vacuum state by applying X gates to the auxiliary qubits along the diagonal of the lattice [14].

3.2 Exponentiation of $XX + YY$

As we just saw, the hopping terms are often of the form

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

with varying number of Z operators. We have 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 $XX + ZZ$ can be exponentiated using only two CNOT gates [20]. This can be converted to $XX + YY$ or $XY + YX$ by applying single-qubit gates. We then use our idea of "controlled reversal gates" [1]. 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.

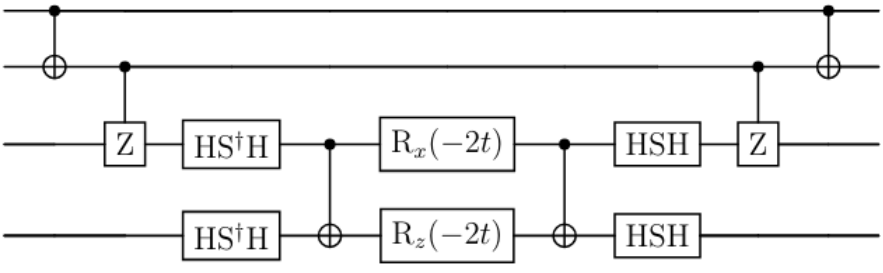


Figure 3.1 Exponentiation of $(XX + YY)ZZ$. The parity of the qubits that Z acts on is collected. Then the CZ gates flip the sign of the $XX + YY$ circuit if necessary.

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 [18], I was able to determine that the controlled version of that 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.

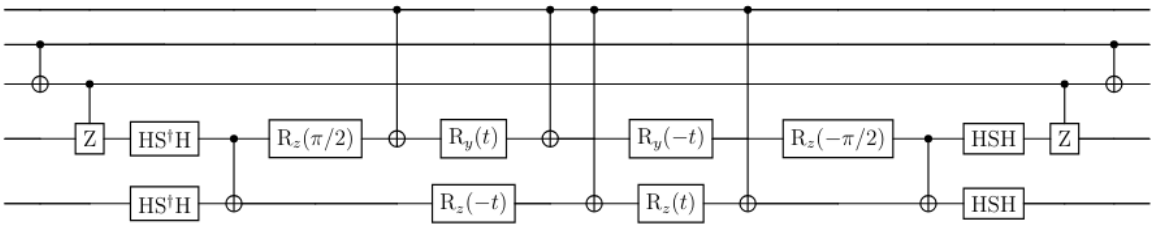


Figure 3.2 Controlled exponentiation of $(XX + YY)ZZ$. The top qubit is the control qubit.

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 $y^1_{\mathbf{0}} \to \Lambda^{11}_{\mathbf{0}}$ we can access the odd parity sector [16, 13]. 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 $i \neq 1$.

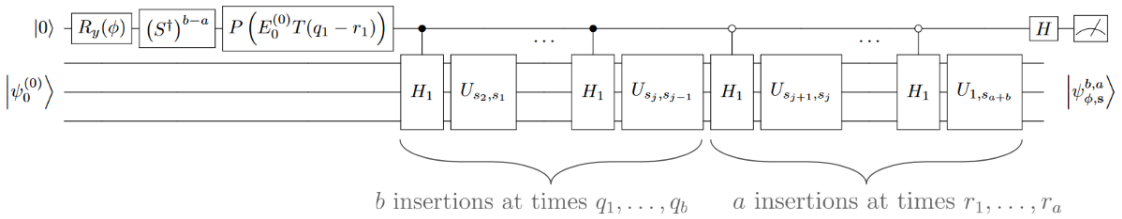
$$
\begin{aligned}
f_{\downarrow}^{p\dagger} &= \frac{1}{2}(\gamma^3 + i\gamma^4) \\
&= \frac{1}{2}\gamma^3(1+\Theta^{34}) \\
&= i\frac{1}{2}\gamma^1\Theta^{13}(1+\Theta^{34}) \\
&= -\frac{1}{2}Y^{(2)}Z^{(3)}Z^{(4)}X^{(5)}(1+Z^{(2)}) \\
f_{\uparrow}^{n\dagger} &= \frac{1}{2}(\gamma^5 + i\gamma^6) \\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

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 Using Perturbation Theory

We want to be able to do calculations using perturbation theory. My colleague Nicholas Cariello constructed a circuit that can calculate $\left\langle \psi_0^{(b)} \right| O \left| \psi_0^{(a)} \right\rangle$, where $\left| \psi_0^{(a)} \right\rangle$ is the ath-order correction to the ground state wave function [3]. The circuit 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



where

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

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

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. [24] (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 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 [9].

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

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 use the $\eta\chi$ method, we are using just one nucleon species per lattice site. This means that we will need two η Majoranas and four χ Majoranas. 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

$$f^{p\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}'} + f^{p\dagger}_{\uparrow,\mathbf{r}'} f^{p}_{\uparrow,\mathbf{r}} = \frac{1}{2}(\Lambda^{24}_{\mathbf{r}}\Lambda^{13}_{\mathbf{r}'} - \Lambda^{14}_{\mathbf{r}}\Lambda^{23}_{\mathbf{r}'}) \tag{3.47}$$

$$= \frac{1}{2}(X^{(1)}_{\mathbf{r}} X^{(1)}_{\mathbf{r}'} + Y^{(1)}_{\mathbf{r}} Y^{(1)}_{\mathbf{r}'}) Z^{(2)}_{\mathbf{r}'}. \tag{3.48}$$

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.

3.5 Perturbative Term

We consider as a perturbation the 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.) Applying the fermion to qubit mapping to the second term, using a 2x2 composite lattice, we find (for $\sigma = \uparrow$):

$$f^{p\dagger}_{\uparrow,\mathbf{r}} f^{n}_{\uparrow,\mathbf{r}} f^{n\dagger}_{\downarrow,\mathbf{r}} f^{p}_{\downarrow,\mathbf{r}} + f^{p\dagger}_{\downarrow,\mathbf{r}} f^{n}_{\downarrow,\mathbf{r}} f^{n\dagger}_{\uparrow,\mathbf{r}} f^{p}_{\uparrow,\mathbf{r}} = \frac{1}{16}(\gamma^1_{\mathbf{r}} + i\gamma^2_{\mathbf{r}})(\gamma^1_{\mathbf{r+2y}} - i\gamma^2_{\mathbf{r+2y}})(\gamma^1_{\mathbf{r+2y+2x}} + i\gamma^2_{\mathbf{r+2y+2x}})(\gamma^1_{\mathbf{r+2x}} - i\gamma^2_{\mathbf{r+2x}}) \tag{3.54}$$

$$+ \frac{1}{16}(\gamma^1_{\mathbf{r+2x}} + i\gamma^2_{\mathbf{r+2x}})(\gamma^1_{\mathbf{r+2y+2x}} - i\gamma^2_{\mathbf{r+2y+2x}})(\gamma^1_{\mathbf{r+2y}} + i\gamma^2_{\mathbf{r+2y}})(\gamma^1_{\mathbf{r}} - i\gamma^2_{\mathbf{r}}) \tag{3.55}$$

$$= \frac{1}{16}(\gamma^1_{\mathbf{r}}\gamma^1_{\mathbf{r+2y}} - i\gamma^1_{\mathbf{r}}\gamma^2_{\mathbf{r+2y}} + i\gamma^2_{\mathbf{r}}\gamma^1_{\mathbf{r+2y}} + \gamma^2_{\mathbf{r}}\gamma^2_{\mathbf{r+2y}})(\gamma^1_{\mathbf{r+2y+2x}}\gamma^1_{\mathbf{r+2x}} - i\gamma^1_{\mathbf{r+2y+2x}}\gamma^2_{\mathbf{r+2x}} + i\gamma^2_{\mathbf{r+2y+2x}}\gamma^1_{\mathbf{r+2x}} + \gamma^2_{\mathbf{r+2y+2x}}\gamma^2_{\mathbf{r+2x}}) \tag{3.56}$$

$$+ \frac{1}{16}(\gamma^1_{\mathbf{r+2x}}\gamma^1_{\mathbf{r+2y+2x}} - i\gamma^1_{\mathbf{r+2x}}\gamma^2_{\mathbf{r+2y+2x}} + i\gamma^2_{\mathbf{r+2x}}\gamma^1_{\mathbf{r+2y+2x}} + \gamma^2_{\mathbf{r+2x}}\gamma^2_{\mathbf{r+2y+2x}})(\gamma^1_{\mathbf{r+2y}}\gamma^1_{\mathbf{r}} - i\gamma^1_{\mathbf{r+2y}}\gamma^2_{\mathbf{r}} + i\gamma^2_{\mathbf{r+2y}}\gamma^1_{\mathbf{r}} + \gamma^2_{\mathbf{r+2y}}\gamma^2_{\mathbf{r}}) \tag{3.57}$$

$$= \frac{1}{16}(-i\Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}} - \Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}} + \Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}} - i\Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}}) \tag{3.58}$$

$$(i\Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}} + \Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}} - \Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}} + i\Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}}) \tag{3.59}$$

$$+ \frac{1}{16}(-i\Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}} - \Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{11}_{\mathbf{r+2y+2x}} + \Lambda^{12}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}} - i\Lambda^{22}_{\mathbf{r+2x}}\Phi^{12}_{\mathbf{r+y+2x}}\Lambda^{21}_{\mathbf{r+2y+2x}}) \tag{3.60}$$

$$(i\Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}} + \Lambda^{12}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}} - \Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{11}_{\mathbf{r+2y}} + i\Lambda^{22}_{\mathbf{r}}\Phi^{12}_{\mathbf{r+y}}\Lambda^{21}_{\mathbf{r+2y}}) \tag{3.61}$$

$$= \frac{1}{16}(-i\Lambda^{12}_{\mathbf{r}} + \Lambda^{22}_{\mathbf{r}})(\Phi^{12}_{\mathbf{r+y}})(\Lambda^{11}_{\mathbf{r+2y}} - i\Lambda^{21}_{\mathbf{r+2y}})(-i\Lambda^{12}_{\mathbf{r+2x}} + \Lambda^{22}_{\mathbf{r+2x}})(-\Phi^{12}_{\mathbf{r+y+2x}})(\Lambda^{11}_{\mathbf{r+2y+2x}} - i\Lambda^{21}_{\mathbf{r+2y+2x}}) \tag{3.62}$$

$$+ \frac{1}{16}(-i\Lambda^{12}_{\mathbf{r+2x}} + \Lambda^{22}_{\mathbf{r+2x}})(\Phi^{12}_{\mathbf{r+y+2x}})(\Lambda^{11}_{\mathbf{r+2y+2x}} - i\Lambda^{21}_{\mathbf{r+2y+2x}})(-i\Lambda^{12}_{\mathbf{r}} + \Lambda^{22}_{\mathbf{r}})(-\Phi^{12}_{\mathbf{r+y}})(\Lambda^{11}_{\mathbf{r+2y}} - i\Lambda^{21}_{\mathbf{r+2y}}) \tag{3.63}$$

$$= -\frac{1}{8}(iY^{(1)}_{\mathbf{r}}Y^{(2)}_{\mathbf{r}} - X^{(1)}_{\mathbf{r}}Y^{(2)}_{\mathbf{r}})(-Z^{(2)}_{\mathbf{r+y}})(Y^{(1)}_{\mathbf{r+2y}}X^{(2)}_{\mathbf{r+2y}} - iX^{(1)}_{\mathbf{r+2y}}X^{(2)}_{\mathbf{r+2y}}) \tag{3.64}$$

$$(iY^{(1)}_{\mathbf{r+2x}}Y^{(2)}_{\mathbf{r+2x}} - X^{(1)}_{\mathbf{r+2x}}Y^{(2)}_{\mathbf{r+2x}})(-Z^{(2)}_{\mathbf{r+y+2x}})(Y^{(1)}_{\mathbf{r+2y+2x}}X^{(2)}_{\mathbf{r+2y+2x}} - iX^{(1)}_{\mathbf{r+2y+2x}}X^{(2)}_{\mathbf{r+2y+2x}}) \tag{3.65}$$

$$= \frac{1}{8}(-iY^{(1)}_{\mathbf{r}} + X^{(1)}_{\mathbf{r}})(-Y^{(2)}_{\mathbf{r}}Z^{(2)}_{\mathbf{r+y}}X^{(2)}_{\mathbf{r+2y}})(-i)(iY^{(1)}_{\mathbf{r+2y}} + X^{(1)}_{\mathbf{r+2y}}) \tag{3.66}$$

$$(-iY^{(1)}_{\mathbf{r+2x}} + X^{(1)}_{\mathbf{r+2x}})(Y^{(2)}_{\mathbf{r+2x}}Z^{(2)}_{\mathbf{r+y+2x}}X^{(2)}_{\mathbf{r+2y+2x}})(-i)(iY^{(1)}_{\mathbf{r+2y+2x}} + X^{(1)}_{\mathbf{r+2y+2x}}) \tag{3.67}$$

$$= 2(Q^{+(1)}_{\mathbf{r}}Y^{(2)}_{\mathbf{r}}Z^{(2)}_{\mathbf{r+y}}X^{(2)}_{\mathbf{r+2y}}Q^{-(1)}_{\mathbf{r+2y}})(Q^{+(1)}_{\mathbf{r+2x}}Y^{(2)}_{\mathbf{r+2x}}Z^{(2)}_{\mathbf{r+y+2x}}X^{(2)}_{\mathbf{r+2y+2x}}Q^{-(1)}_{\mathbf{r+2y+2x}}). \tag{3.68}$$

3.6 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. [14].

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

$$Y^{(2)}_{\mathbf{r}} X^{(2)}_{\mathbf{r+x}} Y^{(2)}_{\mathbf{r+x+y}} X^{(2)}_{\mathbf{r+y}} \tag{3.69}$$

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. [20]. 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 [12].

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 $r \to \lambda_\pm r$, we obtain

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

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

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 $E\pm \rightarrow \lambda_\pm^{-1} E_\pm$.

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 $t \rightarrow \lambda_+ t$, 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,

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

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 [11].

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

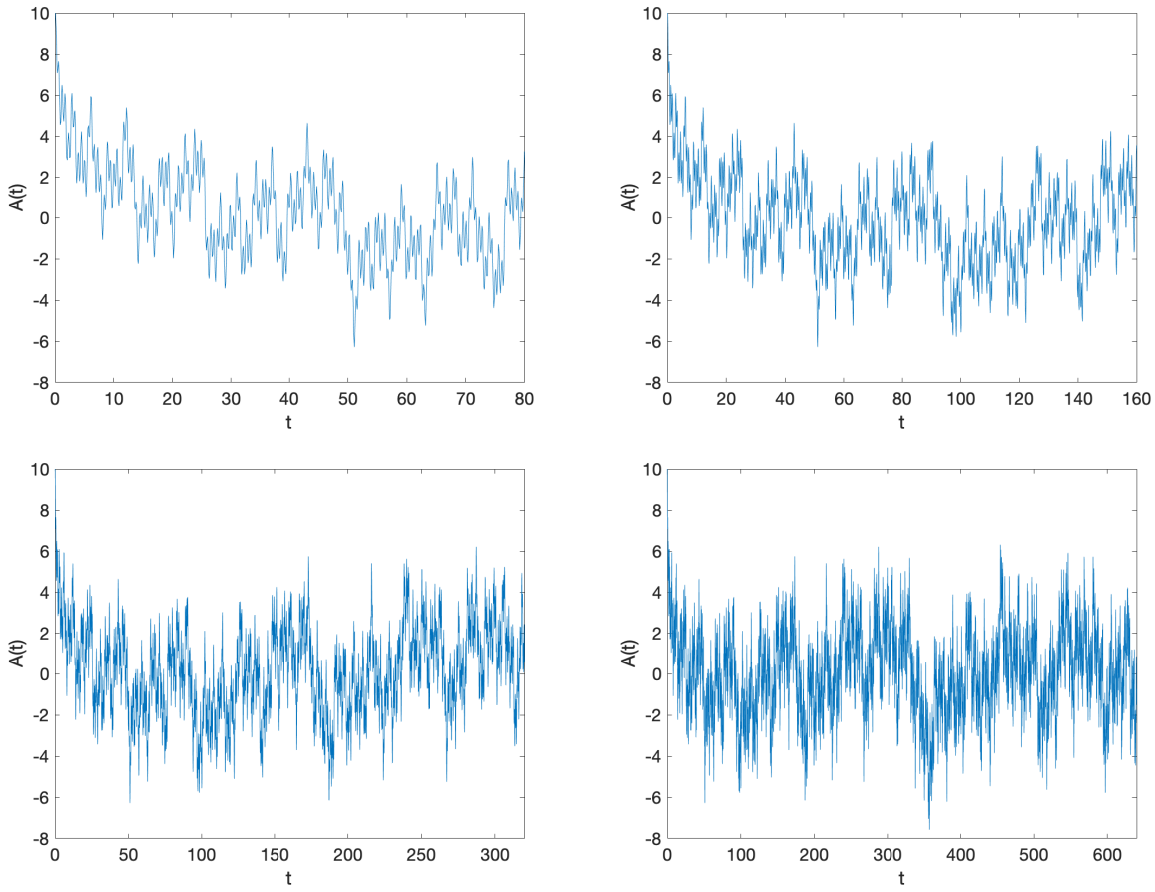


Figure 4.1 An example of time fractals. The amplitude $A(t)$ is displayed over the range from $t = 0$ to 80 in the upper left, $t = 0$ to 160 in the upper right, $t = 0$ to 320 in the lower left, and $t = 0$ to 640 in the lower right. When the time displayed doubles, the overall shape of the amplitude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

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

$$U(t) = e^{-iHt}. \tag{4.17}$$

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

$$f(t) = \frac{1}{\sigma(t)\sqrt{2\pi}} \int_{-R}^{R} e^{-\frac{1}{2}\left(\frac{x}{\sigma(t)}\right)^2} \mathrm{d}x. \tag{4.18}$$

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,

$$f(t) = \mathrm{erf}\left(\frac{R}{\sqrt{2\frac{a^2+(t/m)^2}{a}}}\right). \tag{4.19}$$

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

$$\psi(x) = \begin{cases} 0 & |x| > R \\ \frac{1}{\pi^{1/4}\sqrt{\sigma \mathrm{erf}(R/\sigma)}} e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2} & |x| \leq R \end{cases}. \tag{4.20}$$

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

$$\psi(x) = \begin{cases} 0 & |x| > R \\ \sqrt{\frac{1}{R}} \cos\left(\frac{\pi}{2R}x\right) & |x| \leq R \end{cases}. \tag{4.21}$$

The overlap is

$$O(t) = \frac{e^{-\frac{\pi^2\sigma^2}{8R^2}} \pi^{1/4} \sigma \left(\mathrm{erf}\left[\frac{R}{\sqrt{2}\sigma} - \frac{i\pi\sigma}{2\sqrt{2}R}\right] + \left[\mathrm{erf}\frac{R}{\sqrt{2}\sigma} + \frac{i\pi\sigma}{2\sqrt{2}R}\right]\right)}{\sqrt{2R\sigma \mathrm{erf}(R/\sigma)}}. \tag{4.22}$$

In Fig. 4.2, I compare this model to the projected cooling algorithm simulations in Ref. [11]. 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 [17, 7].

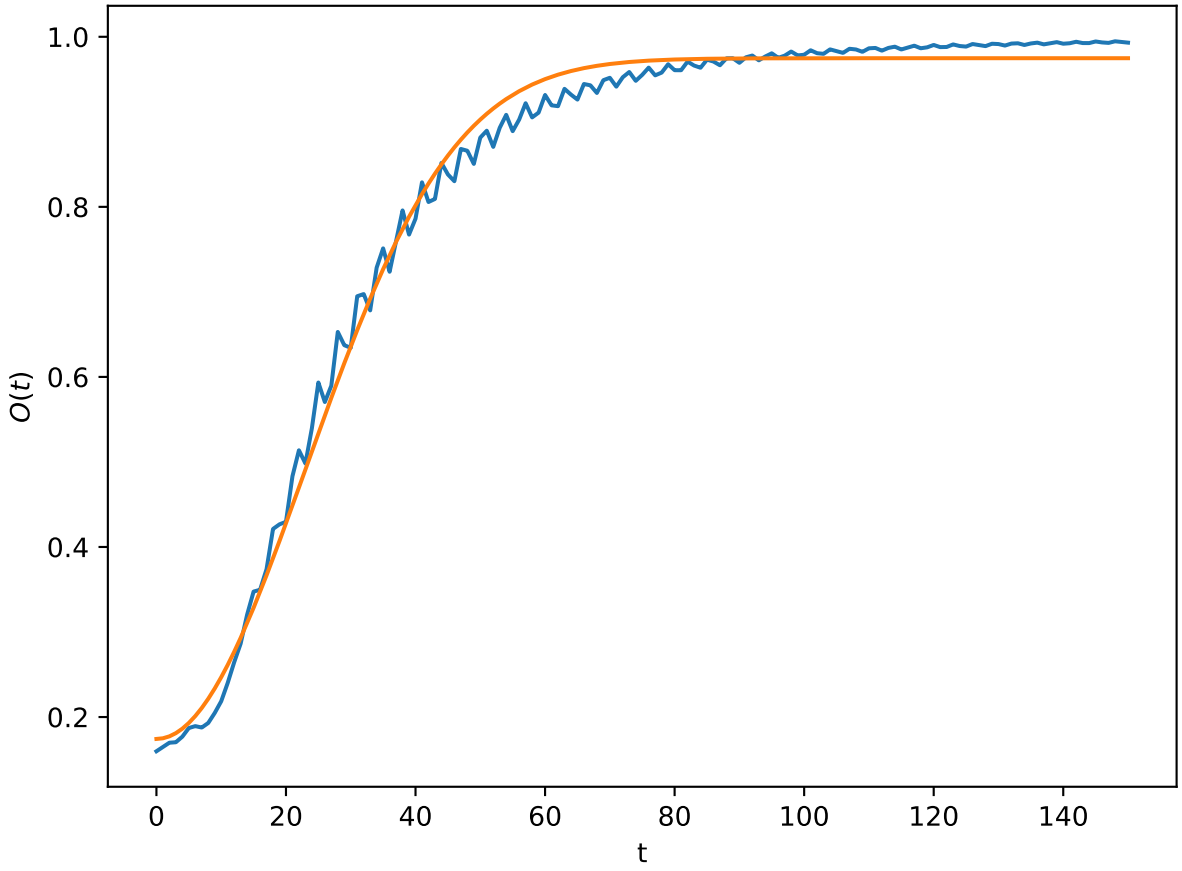


Figure 4.2 A comparison of my Gaussian wave packet model to projected cooling algorithm simulations. The ground state overlap computed in [11] is in blue, and the overlap between a truncated Gaussian wave packet and a particle-in-a-box wave function is in orange.

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

$$P_0(E, t_k) = \cos^2\left[\frac{t_k}{2}(E_{\text{obj}} - E)\right], \tag{4.23}$$

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

$$P_{0^N}(E) = \left[\frac{1 + e^{-(E_{\text{obj}}-E)^2\sigma^2/2}}{2}\right]^N. \tag{4.24}$$

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.

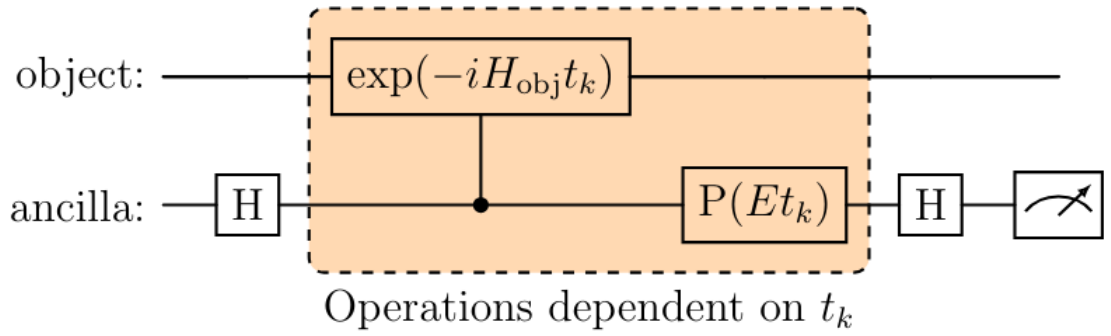


Figure 4.3 One cycle of the rodeo algorithm. In general the object Hamiltonian can have multiple qubits. The object qubit is initialized to some state $|\psi_0\rangle$, and the ancilla qubit is initialized to $|0\rangle$.

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.

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 [17]. 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%.

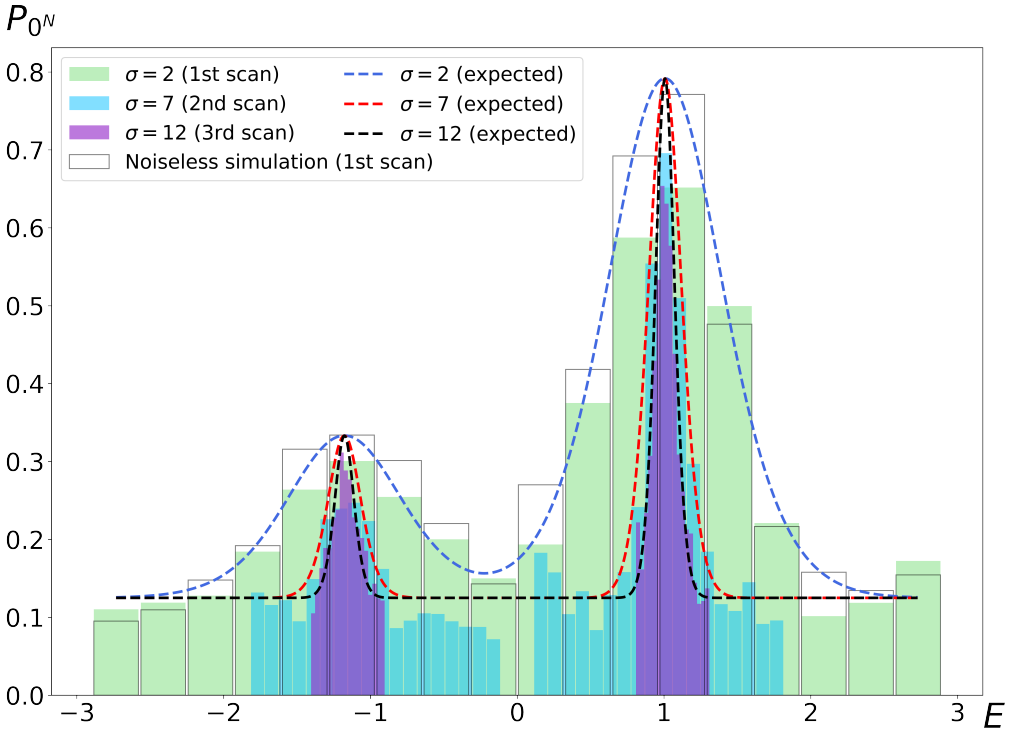


Figure 4.4 Results from running the Rodeo Algorithm on IBM Q Casablanca. The eigenenergies can be extracted from the centers of the peaks.

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. [20], 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] 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.

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

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

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

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

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

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

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

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

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

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

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

[13] 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.

[14] 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.

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

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

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

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

[19] 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).

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

[21] 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.

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

[23] 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.

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

[25] 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.