Time-Evolution of
SU(2) Lattice Gauge Theory
on a Quantum Computer

Sarmed A Rahman

A Thesis Submitted to
The Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science

Graduate Program in Physics and Astronomy
York University
Toronto, Ontario

December 2023

Abstract

Lattice Gauge Theory is a mathematical tool used to study the forces of nature, like Quantum Electrodynamics and Quantum Chromodynamics. Quantum computers offer an alternative to classical computers in studying these forces. In my thesis, a gate-based quantum computer was used to perform calculations of the propagation of an excitation in real-time. A new error mitigation method was developed to greatly extend the range of comprehensible data over time by using the physics circuits to estimate the accumulated error [1]. I also developed the theoretical foundation for higher energy systems, as well as higher dimensional geometry.

Acknowledgements

It is traditional to include a section at the beginning of a thesis that contains recognition and gratitude to those who have helped the writer along his graduate school journey.

I want to dedicate this thesis to my mother, my father, and my little brother, without whom I would have quit after one year of starting.

My father never accepted defeat and always stood in the face of hardship. My mother always taught me to strive for self-betterment. Thank you for always pushing me to move forward.

My brother Marwan always called me on my BS and he set me straight with his surgical dissection of my silly excuses. Thank you for not letting me fall behind.

The rest of my family doesn't care much for theoretical physics, lattice gauge theory, or quantum computing. Thank you for reminding me that there is life outside my office. FIFA 2022 was a blast.

To my best friend, Diaa, thank you for the hilarious memes and the intellectual conversations. I could never predict whether our chats were going to be the epitome of stupidity or sophisticated and intellectual, but it was lots of fun either way.

I want to express my gratitude to Professor Randy Lewis. You took a huge risk accepting an undergraduate student into a project you have just begun. Your passion for physics was contagious and it was an absolute pleasure to be a part of your classes and research team. Solving the mathematics and physics problems felt like a treasure hunt, and it was an amazing adventure.

To my colleagues Emanuele and Sarah, I was always in awe of what you achieved. Your theoretical and computational abilities are exceptional and I see greatness in your futures.

Contents

List of Figures

بسم الله الرحمن الرحيم

1 Introduction

The laws describing the electromagnetic force stipulate that like charges repel, and opposite charges attract. The discovery that atomic nuclei are composed of protons and neutrons was paramount to the world of particle physics because it posed the following question: what prevents the positively charged protons that are held together inside the nucleus from repelling? The answer to this is the strong force. The strong force is one of the four fundamental forces [2]. The theory describing strong interactions and the strong force is called quantum chromodynamics (QCD), which is similar to the theory describing electromagnetism at the quantum scale, called quantum electrodynamics (QED). Both forces are carried by massless, spin-1 gauge bosons: the electrically neutral photon for QED and the color-charged gluons for QCD [3]. The mathematics of such theories exhibit special symmetries that make them gauge invariant, and hence are called gauge theories. The gauge theory describing QED is called $U(1)$ gauge theory, which is the same mathematics that describes complex numbers. On the other hand, the group that describes QCD is called $SU(3)$. While QED has only one electric charge as a degree of freedom, QCD has three color charges. Because of the presence of additional charges and the larger symmetry group, color charges exhibit self interactions. We use lattice gauge theory to study quantum field theories. Lattice gauge theory discretizes space-time, resulting in a grid of links connecting lattice sites. Matter fields are defined on the lattice sites, while gauge fields are defined on the lattice links.

The conventional method for studying lattice gauge theory has been Markov Chain Monte Carlo (MCMC) calculations on classical hardware. The well known issue with that approach is the notorious sign problem, which prevents real-time dynamics of quantum systems due to the appearance of complex values that can not be interpreted as probabilities[4, 5]. Scaling is also a problematic aspect of using classical computing due to the need for exponentially more MCMC sampling points for larger systems [6, 7].

Quantum computing allows us to use quantum properties of matter to forego the restrictions of classical computers. Quantum bits (qubits), unlike classical bits which can take binary values 1 or 0, can be in a superposition of states since each qubit is described by a superposition of 1 and 0. This allows us to accommodate an exponentially growing number of states (2^n) by using n qubits, where n rises linearly. We can also use quantum entanglement on two or more qubits. Gate-based quantum computers provide a set of quantum gates that allow one to encode their problem as a quantum circuit, which is just an ordered set of gate applications [6]. In the case of lattice gauge theory, one can write out the Hamiltonian in terms of the quantum gates. At the end of a quantum circuit, the qubits are measured and the final state is recorded. Many runs are performed, and the expectation value of the Hamiltonian can be constructed from the mean of the measurements.

The purpose of this research project is to explore the use of quantum computing to study lattice gauge theories. Despite the fact that strong interactions are described by $SU(3)$, it was not what we decided to study. While the simplest gauge theory is $U(1)$, which has been studied using quantum computing [8, 9, 10], it is an Abelian gauge theory. That means that the algebra is not comparable to the non-Abelian $SU(3)$. The smallest non-Abelian gauge theory is $SU(2)$, which is the object of this research. $SU(2)$ lattice gauge theory describes physics similar to that of the strong force, but with one less color charge. The mathematics of $SU(2)$ is the same as angular momentum algebra. Thus, studying $SU(2)$ lattice gauge theory will serve to lay the computational groundwork for when quantum computers are powerful enough to deal with the number of qubits needed for large and reliable $SU(3)$ calculations. A further simplification is achieved via the absence of charged particles, although $SU(2)$ calculations have been done with matter on the lattice [11, 12]. The system can be described by a Hamiltonian that is comprised of two parts. The diagonal elements of the Hamiltonian matrix are the chromo-electric components, and the off diagonal elements are the chromo-magnetic components, in analogy with QED and QCD [13]. The main way to deal with the large size of the Hamiltonian is to truncate the matrix. The most common way is to restrict the maximum j angular momentum value, which allows us to study all the possible states with the links having a maximum chromo-electric energy defined by j [14].

In the past, we were the first to use Hamiltonian matrix in conjunction with the time evolution operator to study time evolution on DWave's quantum annealer [15]. However, this time we used IBM's gate-based quantum computer to perform time evolution calculations. The systems we have chosen are linear chains of plaquettes (squares) of lengths two and five. However, it was quickly apparent that the noise accumulation hinders any effort to collect data at large time values. We used the "physics circuit" to time-evolve the initial state of the system by a full time-step t. We then used the same circuit to time evolve the system forward by a half time-step $t/2$, then evolve it backwards by a half time-step $-t/2$. This accumulates the same amount of noise as the physics calculation, but with the upside of knowing that the final state should be the initial state. That allows us to estimate the noise amount, and using that estimate to create a scaling factor for each point, then to adjust the measured physics calculations by said scale factor to account for the noise. This allowed us to expand the range of coherent data points [1]. Now that we have a powerful error mitigation method, we can combine it with the methods from the previous section to reduce the noise even further.

One of the next steps in studying $SU(2)$ lattice gauge theory is to go beyond the $j_{max} = 1/2$ cutoff. While it is the general starting point for the truncation of the Hamiltonian, the next step is to consider $j_{max} = 1$. All the possible states from the $j = 1/2$ case are included, but now we have new states that arise from allowing for the gauge links to have a j value of 1. The total number of states becomes 11, which requires 4 qubits to describe. This poses a challenge, but using the translational symmetry, we can create a new 11×11 matrix, with an 8×8 submatrix that contains the ground state and a superposition of the states that have the same energy. This allows us to use 3 qubits only.

Another way to go beyond what is familiar is to study two-dimensional lattices. Instead of the chain of plaquettes in one dimension, we considered a set of four plaquettes that meet at a vertex, forming a large square. When one studies one-dimensional lattices, there are two types of sites. There are sites with two links, and there are sites with three. This is relevant because it is crucial for constructing the Hamiltonian. To create a state that describes the lattice, one must use the product of all the sites in the lattice. In one-dimensional lattices, the most complex site is the three-linked site. Thus all sites are described by using a general three-linked site state. The sites that have only two links will have a zero for the third. Upon using the procedure for angular momentum addition to derive the three-linked site state, it becomes apparent that it is not important which two links are added together, then added to the third. The order of links does not matter, and thus the process is straightforward. Once the state of the lattice is created, it can be used to create the plaquette operator, which is crucial for calculating the chromo-magnetic part of the Hamiltonian.

The 2×2 clover-leaf lattice in consideration introduces us to the four-linked site. While a three-linked site allowed for an arbitrary ordering of the links, we are not awarded that privilege in the case of the four-linked site. It appears that one must impose a convention for which links to add together at first. The convention chosen was that angular momentum of the links in the positive x and y directions are added together, and the angular momentum of the links in the negative x and y directions are added together. The resultants are added together to get the final state of the four-linked site. Once the four-linked state was derived, many tests were conducted to ensure that the state is equal to the two-linked and three-linked states. That was done with the help of the Wigner 3j symbol identities. The next step was to create the gauge link operator, and use that to calculate the elements of the plaquette operator.

The last contribution was a program that takes in a matrix (e.g. Hamiltonian) and returns the gates that represent the matrix. A tedious part of lattice gauge theory is expressing a system's Hamiltonian matrix in terms of quantum gates, specifically the identity gate (I) and the Pauli gates (X, Y, Z). The process involved trial and error in long periods of calculating matrix products. The need arose for a program that can eliminate that part of our work. The program is based on observed patterns and symmetries that occur when expressing symmetric matrices in terms of Pauli gates. Pauli decomposition algorithms existed before, specially for 2×2 Hermitian matrices. This is merely one possible way of generalizing it to matrices of size $2^n \times 2^n$, which require n qubits to encode. The algorithm takes as an input a $2^n \times 2^n$ matrix M. The algorithm segments the M into 2^n parts using n products of the identity gate (I) and the flip gate (X). These products perform targeted rotations to elements in the matrix such that certain elements are placed on the diagonal, from which they are then extracted into an array. The second part is using Kronecker products of n Hadamard gates to create a "mixing coefficients" matrix. That matrix is used with the array of extracted elements to mix them in just the right way to generate the coefficients of the Pauli gates needed to represent the input matrix M. The program has shown success with Hermitian and non-Hermitian matrices. The part where this algorithm shines is that the method can be used when doing "pen and paper" calculations to find the answers just by looking at the matrix and performing 2^n sums. That is due to the use of symmetry-based patterns, which are the basis of the program.

This thesis is structured in a linear manner. Section 2 is a review of Abelian and non-Abelian gauge theories. Then, the lattice is established via the discretization of space. Lastly, the Hamiltonian form is introduced, which is crucial for studying the physics in terms of the energy of the system. Section 3 puts the non-Abelian $SU(2)$ gauge theory on the lattice and shows the chromoelectric and chromomagnetic fields. Section 4 discusses our past work on calculating $SU(2)$ on a quantum annealer, as well as my calculations on gate-based quantum computers. Both of these works act as motivation for the need for error mitigation techniques. Section 5 discusses our work that introduced a new error mitigation method called self mitigation, which uses the time evolution circuits in order to estimate the errors acquired by the circuit. Section 6 contains my work on creating a Pauli decomposition algorithm that is not only executed with computational ease, but can also be implemented in pen-and-paper calculations due to its reliance on visual symmetries on the square matrix. Finally, section 7 lays the theoretical foundation for extensions to our works in terms of higher energy systems and a higher-dimensional geometry.

2 Background

2.1 Gauge Theory

2.1.1 Abelian Gauge Fields

The field theories used to describe physical systems in nature are indifferent to some transformations to our description. This indifference to gauge transformations is called **gauge invariance**. A **gauge transformation** is simply a change of phase that has no physical significance. Quantum electrodynamics (QED) is described by the U(1) gauge theory, which is the group of unitary matrices of size 1×1, equivalent to complex numbers. We begin with the action of the free Dirac field, which is invariant under global $U(1)$ transformations

$$S_F^{(0)} = \int d^4x \bar{\psi}(x)(i\gamma^\mu \partial_\mu - M)\psi(x) \tag{2.1}$$

where μ is the space-time index. Einstein's summation notation is used, where there is an implied summation over the contracted indices. However, the action is not invariant under a local $U(1)$ transformation

$$\psi(x) \to e^{i\alpha(x)}\psi(x) \tag{2.2}$$

$$\partial_\mu \psi(x) \to e^{i\alpha(x)}[\partial_\mu + i\partial_\mu \alpha(x)]\psi(x) \tag{2.3}$$

where $\alpha(x)$ is a function and not a constant, and therefore the transformation acts differently at different spacetime points. The derivative of the wave function is the source of the issue. We introduce a covariant field $A_\mu(x)$ that can cancel the problematic piece. Thus, we have the covariant derivative:

$$D_\mu = \partial_\mu - igA_\mu(x) \tag{2.4}$$

where g is known as the **coupling strength**. The fermion field couples to the gauge field through the covariant derivative. To understand how the gauge field must transform, we look at the covariant derivative of the fermion field

$$\begin{aligned} D_\mu\psi \to (\partial_\mu - igA'_\mu)e^{i\alpha}\psi &= \partial_\mu(e^{i\alpha}\psi) - igA'_\mu e^{i\alpha}\psi \\ &= i(\partial_\mu\alpha)e^{i\alpha}\psi + e^{i\alpha}\partial_\mu\psi - ige^{i\alpha}A'_\mu\psi \\ e^{i\alpha}\underbrace{\{\partial_\mu - igA_\mu\}}_{=D_\mu}\psi &- ige^{i\alpha}\underbrace{\left\{A'_\mu - A_\mu - \frac{1}{g}\partial_\mu\alpha\right\}}_{=0}\psi \end{aligned}$$

Therefore, by enforcing that the covariant derivative be invariant to local gauge transformations, we find the necessary way that the gauge field transforms

$$A_\mu(x) \to A_\mu(x) + \frac{1}{g}\partial_\mu\alpha(x) \tag{2.5}$$

Now that we have derived the transformation of the gauge field, we can begin to write a gauge invariant action. Using the minimal coupling prescription, we can just replace the ordinary derivatives with the covariant derivative in our fermionic action to get

$$S_F = \int d^4x\bar{\psi}(x)(i\gamma^\mu D_\mu - M)\psi(x) \tag{2.6}$$

In the presence of matter, expanding the covariant derivative in the fermionic action gives

$$S_F = S_F^{(0)} + S_F^{(int)} = \int d^4x\left[(i\bar{\psi}\gamma^\mu\partial_\mu\psi - M\bar{\psi}\psi) + g\bar{\psi}\gamma^\mu\psi A_\mu\right] \tag{2.7}$$

where we retrieve the original fermionic lagrangian with the addition of an interaction term. Looking again at electromagnetism, we see that this is analogous to the four-current coupling to the four-potential A_μ. The interaction here is in terms of the Noether current $j^\mu = \bar{\psi}\gamma^\mu\psi$, and is mediated by the coupling strength.

We must now address one of the consequences of the covariant derivative. In regular calculus, for well behaved functions, the derivatives commute like $(\partial_\mu \partial_\nu - \partial_\nu \partial_\mu)\psi = 0$. However, the covariant derivative commutator does not vanish

$$
\begin{aligned}
[D_\mu, D_\nu]\, \psi &= (\partial_\mu - igA_\mu)(\partial_\nu \psi - igA_\nu \psi) - (\partial_\nu - igA_\nu)(\partial_\mu \psi - igA_\mu \psi) \\
&= -ig(\partial_\mu A_\nu - \partial_\nu A_\mu - ig\, [A_\mu, A_\nu])\psi
\end{aligned}
$$

Since the gauge field is an element of the $U(1)$ group, its commutator vanishes. Thus, we recognize the electromagnetic field strength tensor

$$F_{\mu\nu} = \frac{i}{g}[D_\mu, D_\nu] = \partial_\mu A_\nu - \partial_\nu A_\mu \tag{2.8}$$

We need to add a kinetic lagrangian for the gauge field dynamics. Intuitively, it must use derivatives of the gauge field $\partial_\mu A_\nu$, and must be quadratic with respect to the gauge field, $\sim A^2$ [16]. Thus, we use the square of 2.8 and we get

$$S_G = -\frac{1}{4} \int d^4x F^{\mu\nu}(x) F_{\mu\nu}(x) \tag{2.9}$$

Thus the QED action is the sum of the fermionic action and the gauge action.

2.1.2 Non-abelian Gauge Fields

This section will be based on the Yang-Mills theory and will be mirroring the treatment from the previous section. The 1954 paper by C.N. Yang and R. Mills[17] looked at the procedure of enforcing local gauge invariance by introducing a gauge field that cancels out the problematic parts of the theory and resulting in the description for QED. They applied that treatment to higher order groups. Their interest was non-abelian gauge theories. Specially, the group of unitary $N \times N$ matrices with determinant unity, or $SU(N)$. A non-abelian group contains elements that do not commute. Thus they have a non-zero commutator, and that gives rise to new physics, as we will see in this section.

The gauge transformation on a fermion field ψ is

$$\psi = \begin{pmatrix} \psi^1 \\ \vdots \\ \psi^N \end{pmatrix}, \qquad \psi \rightarrow \Omega(x)\psi = e^{iT^a\alpha^a(x)}\psi \tag{2.10}$$

where T is a vector made of the $N^2 - 1$ infinitesimal generators of the $SU(N)$ group [18]. For SU(2), the components are the Pauli matrices [19]

$$T_a = \frac{\sigma_a}{2}, \quad a = 1, 2, 3 \tag{2.11}$$

and for SU(3), the components are the Gell-Mann matrices [20]

$$T_a = \frac{\lambda_a}{2}, \quad a = 1, \ldots, 8 \tag{2.12}$$

These are $N \times N$ matrices that do not commute. Thus, the commutator and the trace of the $SU(N)$ generators are

$$[T_a, T_b] = if_{abc}T_c \tag{2.13}$$

$$Tr(T_a) = 0, \quad Tr(T_aT_b) = \frac{1}{2}\delta_{ab} \tag{2.14}$$

where f^{abc} are the totally anti-symmetric structure constants of the group [16]. In the case of $SU(2)$, the structure constants are described by the famous Levi-Cevita symbol, ϵ_{ijk}. Following the same procedure as above, the covariant derivative for a non-abelian gauge theory is defined by introducing a $N^2 - 1$ component gauge field that is coupled with the generators of $SU(N)$

$$D_\mu = \partial_\mu - igA^a_\mu(x)T^a \tag{2.15}$$

$$A_\mu(x) \to \Omega(x)A_\mu\Omega^\dagger(x) + \frac{i}{g}\left(\partial_\mu\Omega(x)\right)\Omega^\dagger(x) \tag{2.16}$$

We can now calculate the general form for a non-abelian gauge field transformation by considering an infinitesimal gauge transformation that disregards $O(\alpha^2)$ [21]. Combining equations 2.13, 2.16, and 2.8, we arrive at the gauge field transformation and the field strength tensor defined in terms of the commutator of the generators [22]

$$A^a_\mu \to A^a_\mu + \frac{1}{g}\partial_\mu\alpha^a + f^{abc}A^b_\mu\alpha^c \tag{2.17}$$

$$F^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + gf^{abc}A^b_\mu A^c_\nu \tag{2.18}$$

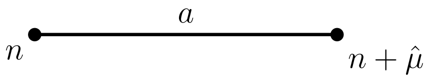
This highlights the emerging difference between abelian and non-abelian gauge theories. In the abelian case, the gauge field is affected by $\partial_\mu\alpha^a$, which is the change in the phase of the local gauge transformation. In the non-abelian case, another term, $A^b_\mu\alpha^c$, describes how α^b causes A^a_μ to be affected by A^c_μ. In the field strength tensor, the new term containing $A^b_\mu A^c_\mu$ describes self interactions of the gauge field components. Due to the new factor in the field strength tensor, the kinetic term in the action will now contain cubic and quartic powers of A^a_μ that describe non-trivial interactions. The gauge action is now the Yang-Mills action

$$S_G = -\frac{1}{2}\int d^4x\ tr\left(F^{\mu\nu}F_{\mu\nu}\right) = -\frac{1}{2}\int d^4xF^{a\mu\nu}F^a_{\mu\nu}\ Tr\left(T^aT^b\right) = -\frac{1}{4}\int d^4xF^{a\mu\nu}F^a_{\mu\nu} \tag{2.19}$$

2.2 The Lattice

2.2.1 Space-time Discretization

The 1974 paper by K. G. Wilson[23] discretized Euclidean spacetime in order to study QCD. The lattice is defined as the set of sites n where the fermion fields are defined. The sites are separated by lattice spacing a, and are connected by links where the gluon fields are defined.



The lattice in $4D$ spacetime is formally defined as

$$\Lambda = \{n = (n_1, n_2, n_3, n_4) | n_1, n_2, n_3 = 0, \ldots, N-1; n_4 = 0, \ldots, N_T - 1\} \tag{2.20}$$

Our theories are described in terms of fields and their derivatives. The fermion fields on the lattice site are now defined as

$$\psi(n) \to \Omega(n)\psi(n) \tag{2.21}$$

$$\bar{\psi}(n) \to \bar{\psi}(n)\Omega^{\dagger}(n) \tag{2.22}$$

The integrals are replaced with sums and the derivatives are replaced with differences

$$\partial_\mu \psi(n) \to \frac{\psi(n+\hat{\mu}) - \psi(n-\hat{\mu})}{2a} \tag{2.23}$$

$$\int d^4x \, \mathcal{L}(x) \to a^4 \sum_{n \in \Lambda} \mathcal{L}(n) \tag{2.24}$$

Thus, the discretized free fermion action can be written as

$$S_F^0 = a^4 \sum_{n \in \Lambda} \bar{\psi}(n) \left(\frac{\psi(n+\hat{\mu}) - \psi(n-\hat{\mu})}{2a} + M\psi(n) \right) \tag{2.25}$$

2.2.2 The Link

Now that the fermion action is discretized, a problem arises. The problem here is that we are dealing with separate spacetime points in equation 2.25 where $\bar{\psi}(n)$, $\psi(n+\hat{\mu})$, and $\psi(n-\hat{\mu})$

have different gauge transformations and are therefore non-invariant

$$\bar{\psi}(n)\psi(n+\hat{\mu}) \rightarrow \bar{\psi}(n)\Omega^{\dagger}(n)\Omega(n+\hat{\mu})\psi(n+\hat{\mu})$$
$$\bar{\psi}(n)\psi(n-\hat{\mu}) \rightarrow \bar{\psi}(n)\Omega^{\dagger}(n)\Omega(n-\hat{\mu})\psi(n-\hat{\mu})$$

We seek an object W that has different right and left transformations such that

$$W(y,x)\psi(x) \rightarrow \Omega(y)W(y,x)\psi(x) \tag{2.26}$$

Such an object is called the parallel transporter. In the continuum, we wanted to make sure that $\partial_{\mu}\psi(x)$ transformed like $\psi(x)$. We modified the derivative and enforced gauge invariance via the introduction of the gauge field $A_{\mu}(x)$. In mathematics, the field $A_{\mu}(x)$ is called the connection. While the connection dealt with infinitesimal gauge invariance, gauge invariance over finite distances must now be considered. Because our symmetry is in terms of local phases, we expect the transporter to be a local phase itself. Since we are moving matter fields through the potential $A_{\mu}(x)$, we expect the argument of the transporter to contain A. Thus, if we wish to perform parallel transport from x_i to x_f along a path Γ, we use the Wilson line [24]

$$W(x_f,x_i) = \mathcal{P}exp\left(ig\int_{\Gamma} dx^{\mu}A_{\mu}(x)\right) \tag{2.27}$$
$$W(x_f,x_i) \rightarrow \Omega(x_f)W(x_f,x_i)\Omega^{\dagger}(x_i) \tag{2.28}$$

where $\mathcal{P}$ is the path ordering operator. We can discretize the Wilson line by evaluating it at point x to point $x+a$ and approximate the integral, then use the Taylor expansion of $A_{\mu}(x+a)$ with respect to a [25]. The result is

$$W(x+a,x) = exp\left\{iagA_{\mu}(n) + O(a^2)\right\} \tag{2.29}$$

Thus, we define link variable $U_\mu(n)$, which is the discretized gauge transporter that exists between sites n and $n+\hat{\mu}$.

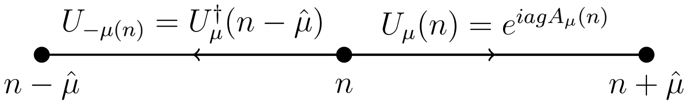

$U_{-\mu(n)} = U_\mu^\dagger(n-\hat{\mu})$
$U_\mu(n) = e^{iagA_\mu(n)}$
$n-\hat{\mu}$
n
$n+\hat{\mu}$

Figure 1: Visual representation of a link variable and its inverse between two sites.

$$U_\mu(n) = exp\left(iagA_\mu(n)\right) = 1 + iagA_\mu(n) + O(a^2) \tag{2.30}$$

$$U_{-\mu}(n) = U_\mu^\dagger(n-\hat{\mu}) \tag{2.31}$$

$$U_\mu(n) \to \Omega(n)U_\mu(n)\Omega^\dagger(n+\hat{\mu}) \tag{2.32}$$

This enables us to retrieve gauge invariance on the lattice. Thus, just as with the continuum case, we redefine the derivative while incorporating a factor that enables us to have a gauge invariant action, but this time we have the link variable

$$\frac{\psi(n+\hat{\mu})-\psi(n-\hat{\mu})}{2a} \to \frac{U_\mu(n)\psi(n+\hat{\mu}) - U_{-\mu}(n)\psi(n-\hat{\mu})}{2a} \tag{2.33}$$

As we take $a \to 0$, the covariant derivative reverts to the continuum version

$$\begin{aligned}
&\frac{U_\mu(n)\psi(n+\hat{\mu}) - U_{-\mu}(n)\psi(n-\hat{\mu})}{2a} \\
&= \frac{1}{2a}\Big[(1+igaA_\mu(n)+O(a^2))(\psi(n)+a\partial_\mu\psi(n)+O(a^2)) - \\
&(1-igaA_\mu(n-\hat{\mu})+O(a^2))(\psi(n)-a\partial_\mu\psi(n)+O(a^2))\Big] \\
&= \frac{\psi(n+\hat{\mu})-\psi(n-\hat{\mu})}{2a} + igA_\mu(n)\psi(n) + O(a^2) \\
&\overset{a\to 0}{=} \partial_\mu\psi(x) + igA_\mu(x)\psi(x)
\end{aligned}$$

This result agrees with what we derived before. Thus, the gauge invariant fermionic action is

$$S_F = a^4 \sum_{n\in\Lambda} \bar{\psi}(n) \left(\frac{U_\mu(n)\psi(n+\hat{\mu}) - U_{-\mu}(n)\psi(n-\hat{\mu})}{2a} + M\psi(n) \right) \tag{2.34}$$

2.2.3 The Plaquette

The next step is to construct the gauge action. To understand the next step, we will briefly restate the motivation for the gauge action in the continuum. The gauge action is the kinetic part of the action that describes the gauge field dynamics. In the continuum, we describe the dynamics in terms of the $F_{\mu\nu}(x)$, which is called “the curvature” in mathematics. This is evident by the fact that the strength tensor has the form of a curl operator applied on the potential. Discretizing the Wilson line gave us the link variable, which is now our gauge field. We now wish to describe the dynamics of the link variable $U_\mu(n)$. The curvature on the lattice is achieved by going in a loop. We can describe a loop in the continuum by setting the endpoint as the start point in the Wilson line equation, which gives us the Wilson Loop. The integral can be rewritten using Stokes' theorem and the result is in terms of the field strength tensor over the area Σ enclosed by the closed path Γ

$$W(y,y) = \mathcal{P}exp \left(ig \oint_\Gamma dx^\mu A_\mu(x) \right) = \mathcal{P}exp \left(ig \iint_\Sigma dx^\mu dx^\nu F_{\mu\nu}(x) \right) \tag{2.35}$$

A loop on the lattice can be described by the plaquette variable defined as the ordered product of link variables representing a loop

$$U_{\mu\nu}(n) = U_\mu(n)U_\nu(n+\hat{\mu})U_{-\mu}(n+\hat{\mu}+\hat{\nu})U_{-\nu}(n) \tag{2.36}$$

Using the Baker-Campbell-Hausdorff formula for the product of exponential matrices:

$$exp\,(A)\,exp\,(B) = exp\left(A + B + \frac{1}{2}[A,B] + \dots \right) \tag{2.37}$$

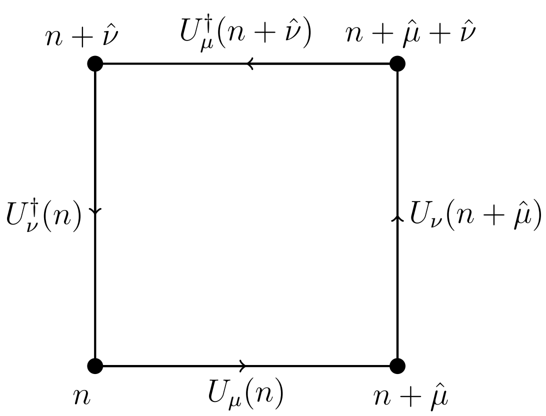


Figure 2: A plaquette with it's path ordered link variables

along with equation 2.30, then Taylor expanding terms such as $A_\mu(n+\hat{\mu}) = A_\mu(n) + a\partial_\mu A_\nu(n) + O(a^2)$, we obtain

$$\begin{aligned} U_{\mu\nu}(n) &= exp\left(iga^2F_{\mu\nu}(n) + O(a^3)\right) \\ &= 1 + iga^2F_{\mu\nu}(n) - \frac{g^2a^4}{2}F^{\mu\nu}(n)F_{\mu\nu}(n) + O(a^6) \end{aligned} \tag{2.38}$$

Therefore the plaquette variable is the lattice version of the Wilson loop. Now that we have the plaquette, we first isolate the real part of the plaquette variable, then the trace. We can construct the action by requiring that the taking $a \to 0$ retrieves $-\frac{1}{4}\int d^4xF^{\mu\nu}F_{\mu\nu}$. Using equation 2.36, we derive the Wilson gauge action

$$S_G = \frac{1}{2g^2}\sum_{n\in\Lambda}\sum_{\mu<\nu}\left[4 - Tr(U_{\mu\nu}(n) + U^\dagger_{\mu\nu}(n))\right] = \frac{a^4}{2}\sum_{n\in\Lambda}\sum_{\mu,\nu}Tr(F^{\mu\nu}F_{\mu\nu}) + O(a^8) \tag{2.39}$$

2.3 Hamiltonian Formulation

The Hamiltonian formulation of the electromagnetic field can be found in many quantum field theory texts [21, 26]. For our consideration we will consider pure gauge theory, and we will neglect the matter fields. The Lagrangian density of the abelian gauge theory from equation 2.9 is used to find the canonical momentum density of the theory

$$\Pi^{\mu\nu}(x) = \frac{\partial \mathcal{L}}{\partial(\partial_\mu A_\nu)} = \frac{\partial \mathcal{L}}{\partial F_{\alpha\beta}} \frac{\partial F_{\alpha\beta}}{\partial(\partial_\mu A_\nu)} = -F^{\mu\nu} \tag{2.40}$$

Since we have a freedom to choose the gauge field, we choose the temporal gauge $A_0 = 0$.

$$\Pi^{0j} = -F^{0j} = -(\partial^0 A^j) = E^j \tag{2.41}$$

$$\mathcal{L} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} = \frac{1}{2}(E^2 - B^2) \tag{2.42}$$

The Hamiltonian density is constructed using the time-like component of the momentum density of the gauge field.

$$\mathcal{H} = \Pi^{0\mu} \partial_0 A_\mu - \mathcal{L} = E^2 - \frac{E^2 - B^2}{2} = \frac{1}{2}(E^2 + B^2) \tag{2.43}$$

is the familiar Hamiltonian density for QED.

The same process will allow us to calculate the Hamiltonian density for a non-abelian gauge theory

$$\mathcal{H} = \frac{1}{2} E^{ai} E^a_i + \frac{1}{4} F^{aij} F^a_{ij} \tag{2.44}$$

The first part of the Hamiltonian is called the chromoelectric part, and the second part is called the chromomagnetic part. Following the steps of Kogut and Susskind in their 1974 paper [27], we will discretize the Hamiltonian. We need to integrate over the space we are working in, and for the purposes of this thesis, it will be a two dimensional space. Using equation 2.39, the Hamiltonian is thus

$$
\begin{aligned}
H = \int d^2x \frac{1}{2} E^{aj} E^a_j + \frac{1}{4} F^{ajk} F^a_{jk} &\rightarrow a^2 \sum_{n\in\Lambda} \left(\frac{1}{2} E^{aj} E^a_j + \frac{1}{4} F^{ajk}(n) F^a_{jk}(n) \right) \\
&= a^2 \sum_{n\in\Lambda} \left(\frac{1}{2} \sum_a (E^a(n))^2 + \frac{1}{a^4 g^2} (4 - Tr(U_{jk}(n) + U^\dagger_{jk}(n))) \right)
\end{aligned}
$$

Since the plaquette variable is an element of $SU(2)$, we can say that $U^\dagger_{\mu\nu} = U_{\mu\nu}$ [28]. We define the plaquette operator as the trace of the plaquette variable $\hat{\Box} = Tr(U_{jk})$. We define the sum over the squared components of the chromoelectric field as the SU(2) Casimir operator $\hat{E}^2(n) = \sum_a (E^a(n))^2$ [12]. The chromoelectric field is the canonical conjugate of the gauge field A^a and has left and right end operators where $\hat{E}^2 = \hat{E}^2_L = \hat{E}^2_R$. This is because links connect two sites, and at each site, there is an angular momentum state that is not necessarily the same. A discussion about the chromoelectric and chromomagnetic energies can be found in [29].

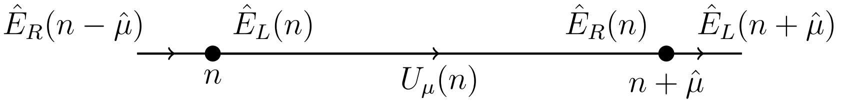


Figure 3: Visual representation of a link variable and its inverse between two sites.

Since the color components of the chromoelectric field are elements of $SU(2)$, they have the following commutation relations [12]

$$[\hat{E}^a_L, \hat{E}^b_L] = -i\epsilon^{abc}\hat{E}^c_L \tag{2.45}$$

$$[\hat{E}^a_R, \hat{E}^b_R] = i\epsilon^{abc}\hat{E}^c_R \tag{2.46}$$

$$[\hat{E}^a_L, \hat{E}^b_R] = 0 \tag{2.47}$$

$$[\hat{E}^a_L, \hat{U}] = T^a\hat{U} \tag{2.48}$$

$$[\hat{E}^a_L, \hat{U}] = \hat{U}T^a \tag{2.49}$$

We discard the constant from the chromomagnetic term and we rescale factors, and define

$x = 2/g^4$ to achieve the SU(2) Kogut-Susskind Hamiltonian as expressed in [13]

$$H_{KS}^{SU(2)} = \sum_{j \in links} \hat{E}_j^2 - 2x \sum_{k \in plaquettes} \hat{\Box}_k \tag{2.50}$$

3 SU(2) Lattice Gauge Theory

To write down the Hamiltonian as a matrix, a basis must be chosen. In $SU(2)$ lattice gauge theory, the color state of each gauge link is described using the product of angular momentum basis $|j,m,m'\rangle = |j,m\rangle \otimes |j,m'\rangle$, where j is the angular momentum quantum number, and m and m' are the angular momentum projections of the left and right ends of a link in the lattice, with $m, m' \in \{-j,+j\}_{\Delta=1}$. Thus the state of the lattice is described by the product of all the sites

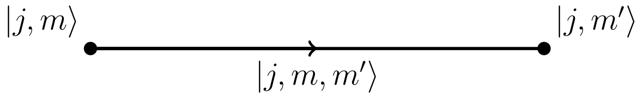


Figure 4: Angular momentum basis description of a link

$$|\Lambda\rangle = \prod_{i \in \text{sites}} |j_i, m_i, n_i\rangle \tag{3.1}$$

Conservation of color is crucial for our theory. Sites on the lattice must form an SU(2) singlet state to adhere to Gauss' law. Addition of angular momenta at the sites of the lattice depends on the number of links converging towards a lattice site.

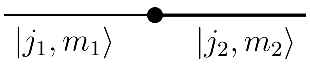


Figure 5: Angular momentum addition of two link ends at a site

For a site with two links connected to it, we use a coupled state to describe the new total angular momentum and the angular momentum projections of the site:

$$|J,M\rangle = \sum_{m_1=-j_1}^{+j_1} \sum_{m_2=-j_2}^{+j_2} \langle j_1, m_1; j_2, m_2 | J, M\rangle \; |j_1, m_1\rangle |j_2, m_2\rangle \tag{3.2}$$

where $J \in \{|j_1 - j_2|, j_1 + j_2\}_{\Delta=1}$ and $M \in \{-J, +J\}_{\Delta=1}$. The Clebsch-Gordan coefficient, $\langle j_1, m_1; j_2, m_2 | J, M\rangle$, describes the allowed angular momentum additions that can occur.

We can convert a Clebsch-Gordan coefficient into an expression in terms of the Wigner-3j symbols using the following identity:

$$\langle j_1, m_1; j_2, m_2 | J, M\rangle = (-1)^{j_2 - j_1 - M}\sqrt{2J+1}\begin{pmatrix} j_1 & j_2 & J \\ m_1 & m_2 & -M \end{pmatrix} \tag{3.3}$$

Appendix A.2 contains a review of 3j symbols and their properties. Thus, the new 3j symbol substitution allows us to write

$$|J, M\rangle = \sum_{m_1=-j_1}^{+j_1}\sum_{m_2=-j_2}^{+j_2}(-1)^{j_2-j_1-M}\sqrt{2J+1}\begin{pmatrix} j_1 & j_2 & J \\ m_1 & m_2 & -M \end{pmatrix}|j_1, m_1\rangle|j_2, m_2\rangle$$

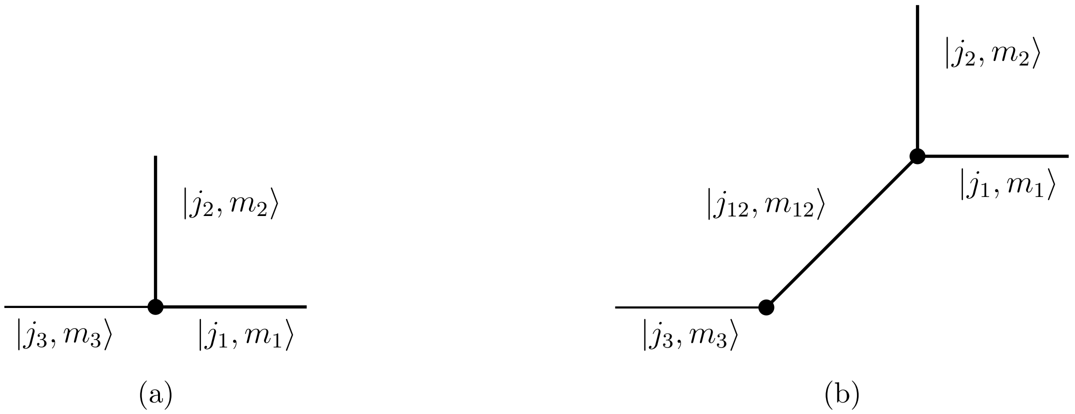
The selection rules for 3j symbols are as follows: $m_1 + m_2 - M = 0$ and $j_1 + j_2 + J =$ integer. Both must be satisfied for the 3j symbol to be non-zero. We are working in the absence of matter fields, and Gauss' law must be respected at each site of the lattice, which means that color must be conserved such that each lattice site has no color. Therefore, we have the following

$$|J = 0, M = 0\rangle = \sum_{m=-j}^{+j}\begin{pmatrix} j & j & 0 \\ m & -m & 0 \end{pmatrix}|j, m\rangle|j, -m\rangle = \sum_m \frac{(-1)^{j-m}}{\sqrt{2j+1}}|j, m\rangle|j, -m\rangle \tag{3.4}$$

For a site with three gauge link connections, we use the procedure above to add the contribution of any of the two links first, then we add the third to that.

Firstly, let's focus on links 1 and 2, and add them together:

$$|j_{12}, m_{12}\rangle = \sum_{m_1=-j_1}^{j_1}\sum_{m_2=-j_2}^{j_2}|j_1, m_1; j_2, m_2\rangle\langle j_1, m_1; j_2, m_2|j_1, j_2, j_{12}, m_{12}\rangle \tag{3.5}$$



(a) (b)

Figure 6: Angular momentum addition of three links at a site

Now to create a state that describes the site, we'll add j_{12} and j_3:

$$
\begin{aligned}
|J, M\rangle &= \sum_{m_{12}=-j_{12}}^{j_{12}} \sum_{m_3=-j_3}^{j_3} \langle j_{12}, m_{12}; j_3, m_3 | j_{12}, j_3, J, M\rangle |j_{12}, m_{12}\rangle |j_3, m_3\rangle \\
&= \sum_{m_1=-j_1}^{j_1} \sum_{m_2=-j_2}^{j_2} \sum_{m_3=-j_C}^{j_3} \sum_{m_{12}=-j_{12}}^{j_{12}} |j_1, m_1\rangle |j_2, m_2\rangle |j_3, m_3\rangle \\
&\langle j_1, m_1; j_2, m_2 | j_1, j_2, j_{12}, m_{12}\rangle \langle j_{12}, m_{12}; j_3, m_3 | j_{12}, j_3, J, M\rangle \\
&= \sum_{m_1=-j_1}^{j_1} \sum_{m_2=-j_2}^{j_2} \sum_{m_3=-j_3}^{j_3} \sum_{m_{12}=-j_{12}}^{j_{12}} |j_1, m_1\rangle |j_2, m_2\rangle |j_3, m_3\rangle \\
&(-1)^{j_2-j_1-m_{12}} (-1)^{j_3-j_{12}-M} \sqrt{2j_{12}+1}\sqrt{2J+1} \begin{pmatrix} j_1 & j_2 & j_{12} \\ m_1 & m_2 & -m_{12} \end{pmatrix} \begin{pmatrix} j_{12} & j_3 & J \\ m_{12} & m_3 & -M \end{pmatrix}
\end{aligned}
$$

In the absence of fermions, the state of the site with three connections becomes:

$$
\begin{aligned}
|J=0, M=0\rangle &= \\
(-1)^{j_3+j_2-j_1} \sum_{m_1=-j_1}^{j_1} \sum_{m_2=-j_2}^{j_2} \sum_{m_3=-j_3}^{j_3} &\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} |j_1, m_1\rangle |j_2, m_2\rangle |j_3, m_3\rangle
\end{aligned} \tag{3.6}
$$

The state of a lattice, $|\Lambda\rangle$, is therefore equivalent to the product of all its site states. For our purposes, the lattice will be a one dimensional array of plaquettes with solid boundary conditions. Thus, the lattice contains two-linked and three-linked sites only.

The next step is to use the angular momentum basis to express the chromoelectric and chromomagnetic parts of the Hamiltonian with the angular momentum basis. We can construct any state by applying the link operator onto an unexcited link state

$$|j,m,m'\rangle = \sqrt{2j+1}U^{j}_{m,m'}|0,0\rangle \tag{3.7}$$

where j is the irreducible representation of $SU(2)$, which corresponds to total angular momentum. A truncation value will be chosen, in this case $j_{max} = 1/2$, so that we limit the size of our Hilbert space. This means that no link can have a j value larger than $j_{\max}$. The gauge link operator will contain two indices for each end of the link. The action of the gauge link operator is expected to be that of angular momentum addition

$$U_{s,s'}|j,m,m'\rangle = \sum_{J=|j-1/2|}^{j+1/2} \sqrt{\frac{2j+1}{2J+1}} \sum_{M.M'} \langle J,M|j,m;\frac{1}{2},s'\rangle\langle J,M'|j,m';\frac{1}{2},s'\rangle|J,M,M'\rangle \tag{3.8}$$

$$= \sum_{J,M,M'} \sqrt{2j+1}\sqrt{2J+1}(-1)^{1-2j+M+M'} \begin{pmatrix} j & \frac{1}{2} & J \\ m & s & -M \end{pmatrix} \begin{pmatrix} j & \frac{1}{2} & J \\ m' & s' & -M' \end{pmatrix} |J,M,M'\rangle$$

where $M = m + n$. The application of the Casimir operator results in the eigenvalue of the link state

$$\hat{E}^2|j,m,m'\rangle = j(j+1)|j,m,m'\rangle \tag{3.9}$$

Thus, each link contributes a factor of $j(j+1)$ to the chromoelectric energy.

We can now begin to construct the plaquette operator. The focus of this thesis is one dimensional lattices. A test plaquette with the relevant links labeled is shown in figure 7. Any links that do not touch the plaquette will not be affected by the plaquette operator, and will adhere to the orthogonality relation

$$\langle l, m_L, m_R | j, n_L, n_R \rangle = \delta_{lj}\delta_{m_L n_L}\delta_{m_R n_R} \tag{3.10}$$

The adjacent links have been included because they are involved in Gauss' law. The one dimensional lattice ensures that the top and bottom adjacent links are the same. The full derivation for the plaquette operator expectation value can be found in appendix A.3. The action of the plaquette operator is

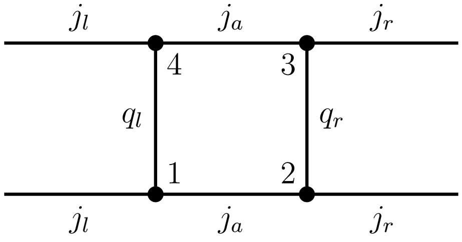


Figure 7: An arbitrary plaquette in a 1D lattice with the relevant links labeled

$$\Box = \sum_{s_1,s_2,s_3,s_4} (-1)^{s_1+s_2+s_3+s_4} U^{\rightarrow}_{-s_1,s_2} U^{\uparrow}_{-s_2,s_3} U^{\leftarrow}_{s_4,-s_3} U^{\downarrow}_{s1,-s_4} \tag{3.11}$$

The result is therefore [14]

$$\begin{gathered} \langle j_l, q_{lf}, j_{af}, q_{rf}, j_r | \hat{\Box} | j_l, q_{li}, j_{ai}, q_{ri}, j_r \rangle = \\ dim(j_{ai})dim(j_{af})\sqrt{dim(q_{li})dim(q_{lf})dim(q_{ri})dim(q_{rf})}(-1)^{2(2j_{af}+j_l+j_r-q_{li}-q_{ri})} \\ \begin{Bmatrix} j_l & j_{ai} & q_{li} \\ \frac{1}{2} & q_{lf} & j_{af} \end{Bmatrix}^2 \begin{Bmatrix} j_r & j_{ai} & q_{ri} \\ \frac{1}{2} & q_{rf} & j_{af} \end{Bmatrix}^2 \end{gathered} \tag{3.12}$$

where $dim(j) = 2j + 1$. Equations 3.9 and 3.12 show that the only information needed in calculating the Hamiltonian is the total angular momentum of each link and not the projections. Thus, the states will only show the j values of the links.

The Hamiltonian matrix element can be written as

$$\langle \Lambda_{\text{final}} | \hat{H} | \Lambda_{\text{initial}} \rangle = \sum_{l \in \text{links}} j_l(j_l + 1)\delta_{\text{initial,final}} - 2x \sum_{p \in \text{plaquettes}} \langle \Lambda_{\text{final}} | \hat{\square}(p) | \Lambda_{\text{initial}} \rangle \tag{3.13}$$

If the final state matches the initial state, the chromoelectric terms will be computed and they lie on the diagonal of the matrix. The chromomagnetic terms are akin to the transition elements and they lie in the off-diagonal region. The plaquette terms act as transition elements between two states. They encode information about how one state becomes another under the plaquette operator.

4 Quantum Computing

Quantum computing replaces classical computing's bits with quantum bits, called qubits [30]. Classical computers use transistors that act like switches; they are either 100% "on" or 100% "off". Qubits on the other hand use nanoscale materials and use their quantum properties, such as entanglement and superposition. A system of one bit needs only one piece of information to describe whether the bit is in the 0 state or the 1 state, while a qubit is described by $a|0\rangle + b|1\rangle$, where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. A system of two bits can be in one of four states: $\{00, 01, 10, 11\}$. On the other hand, a system of two qubits can have infinitely many states $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$, where $a, b, c, d \in \mathbb{C}$ and $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. N classical bits can encode 2^N states, while N qubits can encode an infinitely many states as the superposition of the 2^N states. These infinite states exist in a continuous range between the 2^N basis states. Measurements of a quantum system are stochastic. When measuring a system, the state that was in superposition prior to measurement collapses to one of the 2^N states, and multiple "runs" of the quantum program is used to find the coefficients of the state. These coefficients are normalized, and any operation on the qubit states is unitary, and thus preserving the normalization condition.

4.1 Quantum Annealing

This section is based on our 2021 publication that inspired the work in this thesis [15]. A quantum annealer is a specialized quantum computer that uses qubits but no gates. Annealers are designed to find the ground state of an Ising Hamiltonian as a function of binary variables q_i

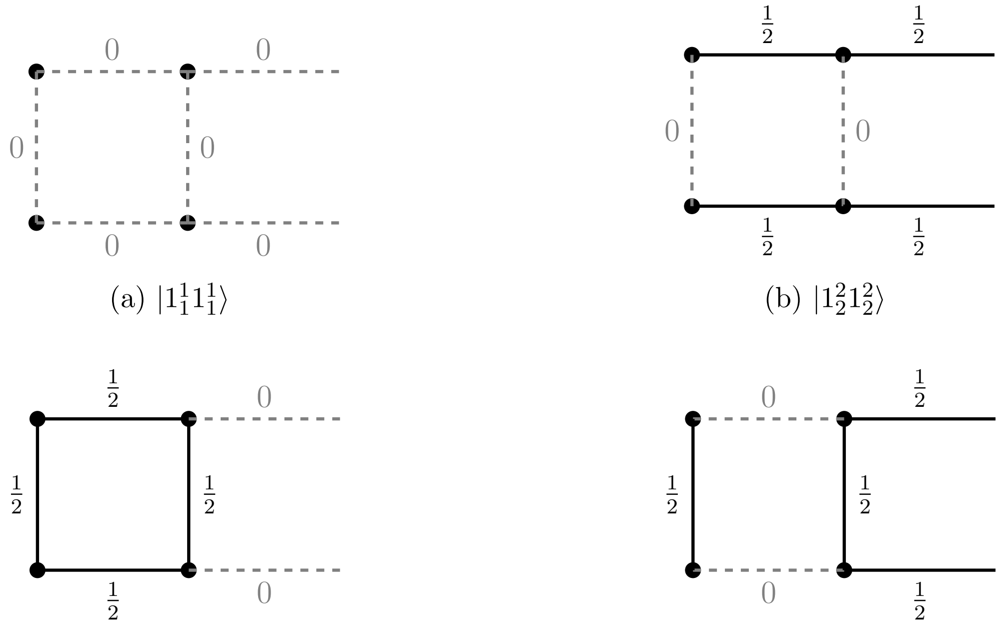
$$H(q) = \sum_{i=1}^{N} h_i q_i + \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij} q_i q_j \tag{4.1}$$

The DWave quantum annealer uses a simple Hamiltonian as the initial state of the system. The ground state of the system is then adiabatically moved to the ground state of the input

Ising Hamiltonian. The work used various lattices of sizes 2, 4, and 6 plaquettes. A distinct feature of these lattices is that they have periodic boundary conditions, which can be thought of as wrapping the rectangle into a cylinder. The Hamiltonian for the 2 plaquette lattice is constructed using the plaquette operator on the vacuum state a number of times until all possible states are constructed.

$$H = \frac{g^2}{2}\left[\begin{array}{cccc} 0 & -2x & -2x & 0 \\ -2x & 3 & 0 & -\frac{x}{2} \\ -2x & 0 & 3 & -\frac{x}{2} \\ 0 & -\frac{x}{2} & -\frac{x}{2} & 3 \end{array}\right]\begin{array}{l} |1^1_1 1^1_1\rangle \\ |2^2_2 2^1_1\rangle \\ |2^1_1 2^2_2\rangle \\ |1^2_2 1^2_2\rangle \end{array} \tag{4.2}$$

The states make use of superscripts and subscripts in order to represent top and bottom horizontal links, while the regular script numbers are the vertical links. The values of these numbers are $A = 2j_A + 1$. The DWave software allows for easy handling of Hamiltonian matrices that are not size $2^N \times 2^N$, like on gate-based quantum computers. This is because



(a) $|1^1_1 1^1_1\rangle$

(b) $|1^2_2 1^2_2\rangle$

(c) $|2^2_2 2^1_1\rangle$

(d) $|2^1_1 2^2_2\rangle$

Figure 8: The physical lattice states of a two plaquette lattice with periodic boundary conditions

only the non-zero matrix elements need to be defined on a quantum annealer, while the entire matrix needs to be $2^N \times 2^N$ and expressed as a sequence of products of quantum gates. This means that there will not be a non-physical sector. The lack of gates in the quantum annealer comes at the cost of requiring many more qubits than gate-based quantum computers. The next step is to consider the possible symmetries that can be exploited to reduce the size of the matrix. We can use translational symmetry to create singlet and triplet states, which allows us to block diagonalize the Hamiltonian into a 3×3 block that includes the ground state

$$H = \frac{g^2}{2} \begin{array}{c} \left[\begin{array}{ccc|c} 0 & -2\sqrt{2}x & 0 & 0 \\ -2\sqrt{2}x & 3 & -\frac{x}{\sqrt{2}} & 0 \\ 0 & -\frac{x}{\sqrt{2}} & 3 & 0 \\ \hline 0 & 0 & 0 & 3 \end{array} \right] \end{array} \begin{array}{c} |1^1_1 1^1_1\rangle \\ \frac{1}{\sqrt{2}}(|2^2_2 2^1_1\rangle + |2^1_1 2^2_2\rangle) \\ |1^2_2 1^2_2\rangle \\ \frac{1}{\sqrt{2}}(|2^2_2 2^1_1\rangle - |2^1_1 2^2_2\rangle) \end{array} \tag{4.3}$$

For larger lattices, there are also left-to-right reflections at a symmetry point, and spatial translations along the periodic direction. For two-dimensional lattices, and for cases where $j_{max} > 1/2$, there are top-to-bottom reflections. The procedure is then to collect the set of states belonging to a symmetry category and create new symmetric and anti-symmetric states. A 4-plaquette lattice with 64 states was used as an example. The states are described by $|Q_i^{(j)}\rangle$, which is the i^{th} set with the excitation translated by j sites. Symmetric and anti-symmetric combinations are created from states that have left-to-right symmetry. The translation symmetry is expressed using e^{ipx}, where the allowed momenta are calculated by distributing six points over a circle, $p = \{-2\pi/3, -\pi/3, 0, \pi/3, 2\pi/3, \pi\}$, and x is the integer location of the plaquette. The real and imaginary parts are used separately. The result of such operations is the reduction of the size of the Hamiltonian to a 13×13 ground state block. The quantum annealer was used to compute eigenvalues and eigenvectors. The variational method is used to approximate the ground state energy

$$E_0 \leq \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \tag{4.4}$$

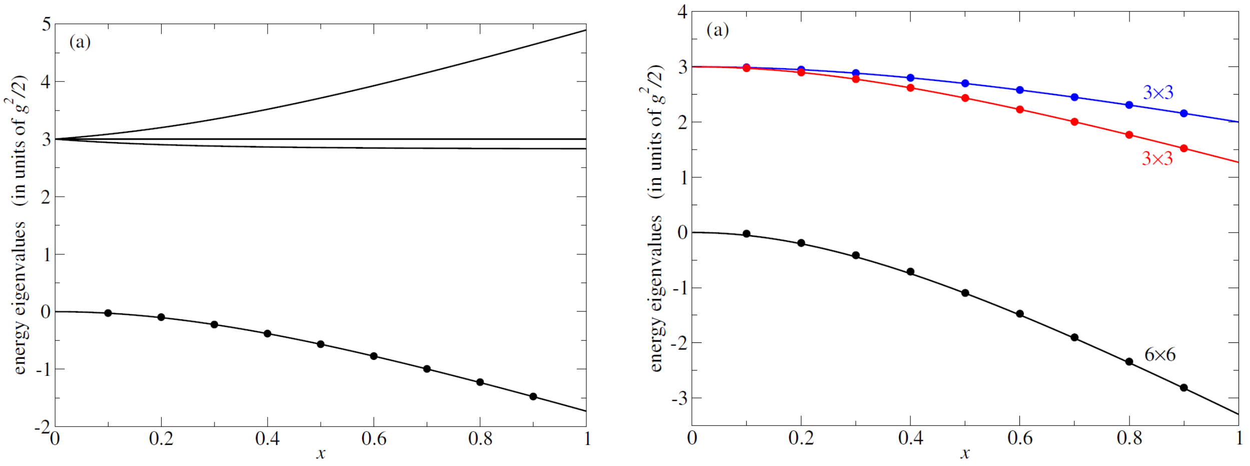
This is called the quantum annealer eigensolver (QAE). The two modifications that are made for the quantum annealer application, in the QAE algorithm, so that all states ψ can be used, as opposed to binary valued states, and the null vector, where $q_i = 0, \ \forall i$, must be dealt with to not give a false minimum. Thus, introducing an adjustable offset parameter λ that differentiates between the null vector and the ground state

$$\langle\psi|H|\psi\rangle \rightarrow F = \langle\psi|H|\psi\rangle - \lambda\langle\psi|\psi\rangle \tag{4.5}$$

The states are extended beyond the single binary option by using multiple binary variables. Thus, the i^{th} element of the state, described by K binary variables, is

$$a_i = -q_K^i + \sum_{k=1}^{K-1} \frac{q_K^i}{2^{K-k}} \ , \quad a_i \in [-1, 1) \ , \quad i = 1, \ldots, N \tag{4.6}$$

Thus, for a $N \times N$ matrix, the state is expected to have N elements, and NK logical qubits are needed. The set of physical qubits that are used to represent one logical qubit is called a chain. The length of the chain is determined by the inter-qubit connections. The strength of the chain is an adjustable parameter that is implemented in the background of the DWave system. The agreement on the state of each logical qubit in the chain is influenced by the value of the chain strength, where disagreement is called chain breaking. The abundance of qubits on the DWave hardware allowed for the use of 21 logical qubits for the vacuum block of the Hamiltonian in 4.3, where each state is described by $K = 7$ logical qubits. The calculation is performed in 1000 annealing cycles for the 3×3 case. A new version of QAE called adaptive quantum annealer eigensolver (AQAE) was developed in the work. The AQAE algorithm adjusts the number of logical qubits in a chain by zooming in until



(a)

(b)

Figure 9: [15] The curves are the classically computed four eigenvalues of Eq. 4.3, and the data points are computed on DWave hardware for the two-plaquette lattice in 9a and the four-plaquette lattice in 9b. This figure also appears in Ref. [15]

the desired accuracy is achieved. The AQAE algorithm uses a value of $K = 4$, and adds a logical qubit at each adaptive step until the measured eigenvalue accuracy stops.

Vacuum expectation values were calculated via the differences of the energies, $E_i - E_0$, that were calculated using AQAE.

$$\langle 0|H|0\rangle = \langle 0|H_E|0\rangle + \langle 0|H_\Box|0\rangle \tag{4.7}$$

The data points from the quantum annealer calculations in figure 11a show agreement with the classical results for $j_{\text{max}} = \frac{1}{2}$, and show small deviations for $j_{\text{max}} = 1$. The sum of the electric and the magnetic parts is closer to the classical curve than the parts are to their respective classical curves. The dependence on the volume of the lattice is addressed by dividing equation 4.7 by the number of plaquettes, which gives the energy density.

Figure 11b shows the data points for lattices with two, four, and six plaquettes at $j_{\text{max}} = \frac{1}{2}$. The reason why there are no distinct points for N=6 is because of the agreement of the N=4 and 6 data points. This figure shows that there is no significant dependence on lattice volume at the range of x that is used. This is an indication of the locality of the

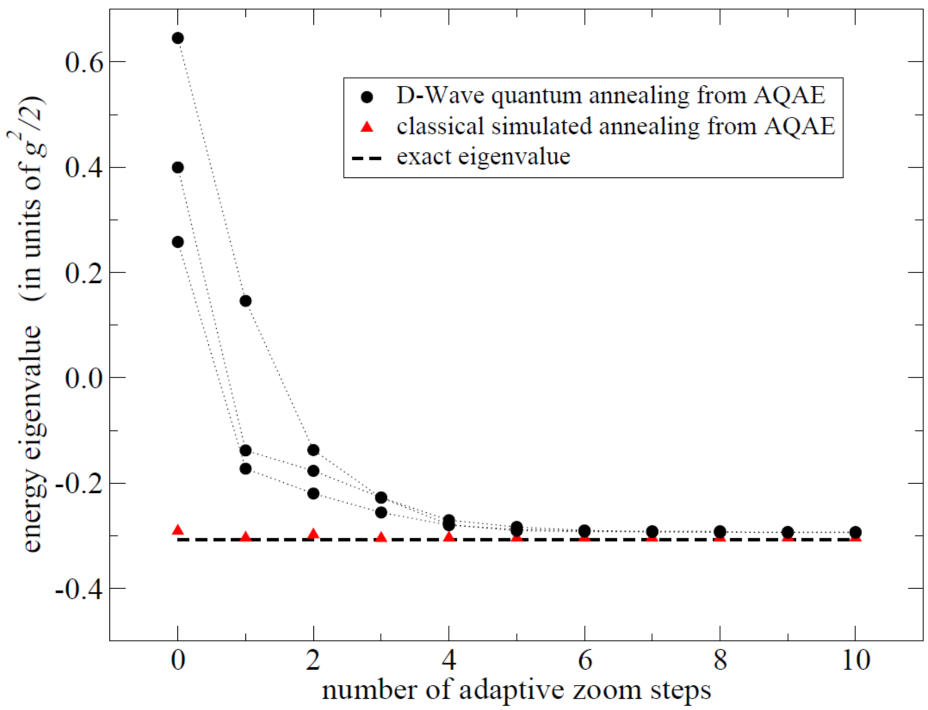
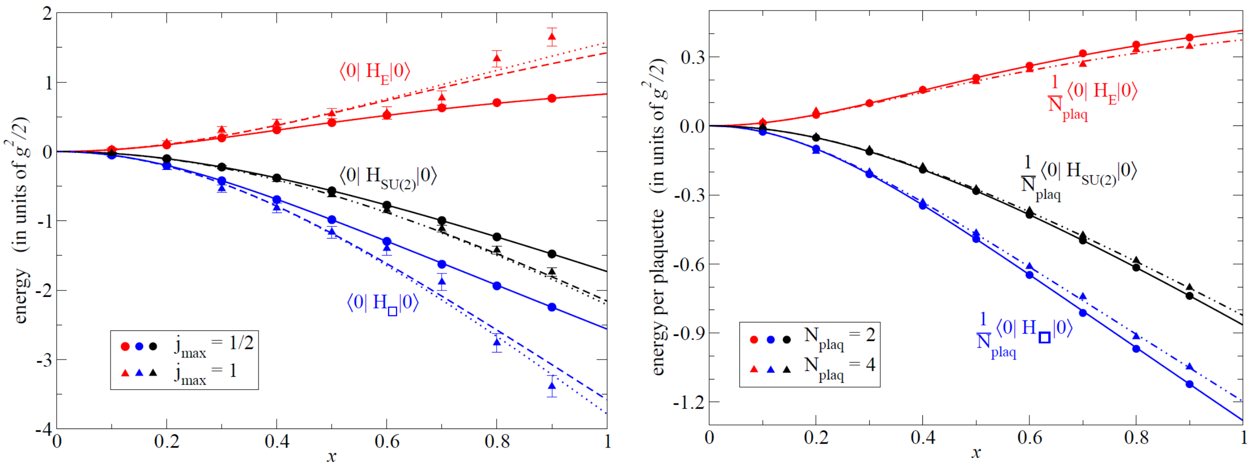


Figure 10: AQAE with runs=1000, K=4 per entry, j=1/2, and x=0.2 for the ground state of a six plaquette lattice. This figure also appears in Ref. [15]

energy densities. The figure also shows that the total energy density has less error than the chromoelectric and chromomagnetic energy densities.

The next piece of the work is dedicated to the time-evolution calculations. Traditional lattice gauge theory methods rely on the usage of Euclidean time in its calculations. Euclidean time, or imaginary time, is defined as the result of rotating real time around a complex time space via a Wick rotation. This prevents real-time dynamics calculations on classical computers. Due to the lack of a gate set from which operators can be built, the time evolution problem must be translated into a ground-state eigenvalue problem. This is done using Kitaev-Feynman clock states, which will be motivated by following the work [31]. The state $|\psi(t)\rangle$ of a system described by the Hamiltonian $H(t)$ can be written as the time evolution of the initial state

$$|\psi(t)\rangle = \mathcal{T}\left(e^{-i\int_{t_0}^{t} dt' H(t')}\right)|\psi(t_0)\rangle = U(t,t_0)|\psi(t_0)\rangle \tag{4.8}$$



(a)

(b)

Figure 11: 11a shows the chromoelectric and chromomagnetic parts of the $SU(2)$ Hamiltonian on a two-plaquette lattice. Solid, dashed, and dotted curves are the classical calculations for $j_{\text{max}} = \frac{1}{2}$, 1, and $\frac{3}{2}$ respectively. 11b shows the energy density calculations with $j_{\text{max}} = \frac{1}{2}$. The solid, dashed, and dotted curves are classical calculations for $N_{\text{plaq}} = 2$, 4, and 6. This figure also appears in Ref. [15]

where $\mathcal{T}$ is the time ordering operator, and $U(t, t_0)$ is the time evolution operator. The time evolution operator evolves a system from time t_0 to time t. The time evolution operator can be written as the successive applications of time evolution operators

$$|\psi(t)\rangle = U(t, t_0)|\psi(t_0)\rangle = U(t, t_n)U(t_n, t_{n-1}) \ldots U(t_1, t_0)|\psi(t_0)\rangle \tag{4.9}$$

where $t < t_1 < \ldots < t_n < t$. For numerical calculations, time is discretized with T discrete time steps through the usage of states integer indices $|0\rangle$, $|1\rangle, \ldots,$ $|T-1\rangle$. Thus, a history state can be constructed with the idea that the physical system representing $|\psi\rangle$ can be entangled with an auxiliary system to represent discrete time. The history state is thus a superposition of the entangled states at all times

$$|\phi\rangle = \frac{1}{\sqrt{T}} \sum_t |\psi_t\rangle \otimes |t\rangle \tag{4.10}$$

where $|\psi_i\rangle = |\psi(t)\rangle|_{t=t_i}$ is time-independent. The time evolution operator is also discretized by acting on the physical system with $U(t_1, t_0) = U_0$, and on the time system by an outer product that acts on $|0\rangle$ and returns $|1\rangle$. Thus, the operator is $U_0 \otimes |1\rangle\langle 0|$. This is generalized to all time steps. The clock Hamiltonian is defined as

$$\mathcal{H} = C_0 + \frac{1}{2}\sum_{t=0}\Big(I \otimes |t\rangle\langle t| - U_t \otimes |t+1\rangle\langle t| - U_t^\dagger \otimes |t\rangle\langle t+1| + I \otimes |t+1\rangle\langle t+1|\Big) \tag{4.11}$$

$$C_0 = (I - |\psi_0\rangle\langle\psi_0|) \otimes |0\rangle\langle 0| \tag{4.12}$$

where C_0 is a penalty term that specifies the initial state. The function to be minimized is therefore

$$\mathcal{L} = \sum_{t,t'}\langle t'|\langle\psi_{t'}|C|\psi_t\rangle|t\rangle - \lambda\left(\sum_{t,t'}\langle t'|\langle\psi_{t'}|\psi_t\rangle|t\rangle\right) \tag{4.13}$$

This is known as time-embedded discrete variational principle (TEDVP). A QAE+TEDVP can be used to perform calculations after performing a transformation that turns the Hamiltonian into an imaginary matrix. Taking the 3×3 block from equation 4.3, and adding a heavy state to get

$$H_{\text{new}} = \frac{g^2}{2}\begin{bmatrix} 0 & -2\sqrt{2}x & 0 & 0 \\ -2\sqrt{2}x & 3 & -\frac{x}{\sqrt{2}} & 0 \\ 0 & -\frac{x}{\sqrt{2}} & 3 & -2\sqrt{2}x \\ 0 & 0 & -2\sqrt{2}x & 6 \end{bmatrix} \tag{4.14}$$

which is re-written as a purely imaginary matrix using transformation matrices P

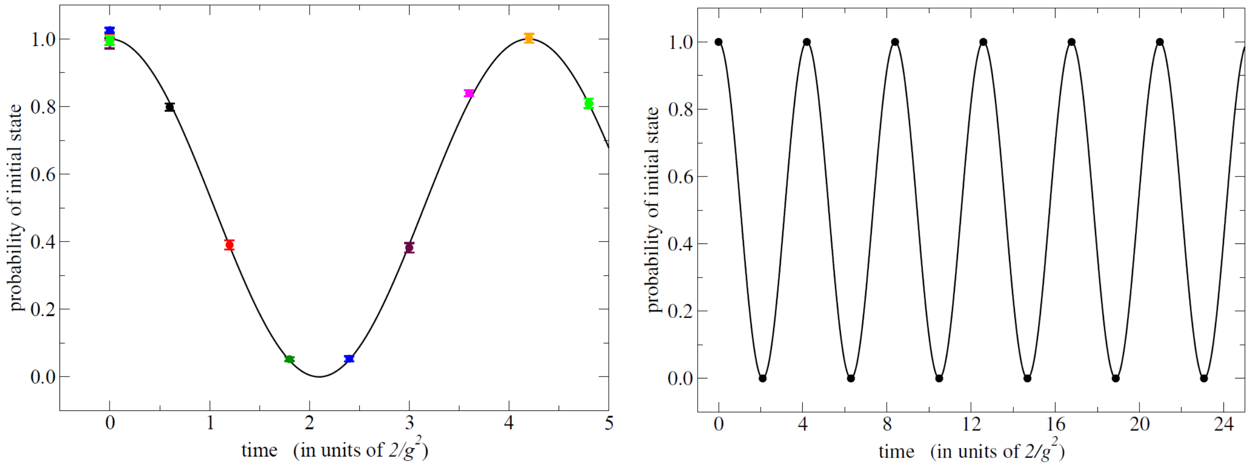
$$P^{-1}H_{\text{new}}P = \frac{g^2}{2}\begin{bmatrix} 3 & 0 & 0 & -ih_- \\ 0 & 3 & -ih_+ & 0 \\ 0 & ih_+ & 3 & 0 \\ ih_- & 0 & 0 & 3 \end{bmatrix} \tag{4.15}$$

$$h_\pm = \frac{1}{2}\sqrt{18 + 33x^2 \pm \sqrt{65x^4 + 1116x^2 + 324}} \tag{4.16}$$

We have two blocks in the imaginary Hamiltonian. The h_+ block contains the ground state and the heavy state we created. The h_- block contains the two intermediate states, which are the states describing the excited lattice and the superposition of single excited plaquettes, which are of interest. The last step in turning the Hamiltonian purely imaginary is to remove the $3II$ part (on the diagonal) because it is only a constant and only energy differences matter. Thus, we take the block of interest, and we use it to calculate the time-evolution operator, which is equivalent to a Pauli-Y rotation

$$U_t = \begin{bmatrix} \cos(\omega\epsilon) & -\sin(\omega\epsilon) \\ \sin(\omega\epsilon) & \cos(\omega\epsilon) \end{bmatrix}, \quad \omega = \frac{g^2}{2}h_- \tag{4.17}$$

For the case of small x, the effect of adding the heavy state is less than the effect of truncating the matrix to 3 × 3. Due to the restrictions of the DWave quantum annealer, only certain angles are allowed. DWave's quantum annealer was successfully utilized for lattice gauge theory calculations. QAE and AQAE were were used to calculate the ground state energy without the need for quantum gates, but needed lots of pre-processing in the form of using symmetries and block diagonalization. Vacuum expectation values were also successfully computed. Also, for the first time, a time evolution calculation was performed on a quantum annealer using QAE+TEDVP. The time-evolution calculation required intensive pre-processing and was heavily restricted with regards to time points that could be



(a)

(b)

Figure 12: The probability oscillations between states $|1^2_2 1^2_2\rangle$ and $\frac{1}{2}\left(|2^2_2 2^1_1\rangle + |2^1_1 2^2_2\rangle\right)$. The data points were computed with $x = 0.1$. Figure 12a was calculated using two time steps per run ($t = 0$ and one $t \neq 0$) with $K = 7$ for a total of 28 logical qubits per calculation and 5×10^4. Each run has its own color at two locations on the graph. Figure 12b was calculated using 10^4 reads with $\lambda = 0.12$ and chain strength $= 0.3$. Each state uses $K = 2$ for a total of 28 logical qubits for all the 12 data points. The solid lines of both figures are the classically calculated result with $\cos^2(\omega t)$. This figure also appears in Ref. [15]

calculated. The difficulty with time-evolution calculations on a quantum annealer makes gate-based quantum computers a more ideal choice for such calculations.

4.2 Gate-Based Quantum Computing

This section will introduce IBM's gate-based quantum computation and its application to ground state energy calculations, as well as the need for noise mitigation. Gate-based quantum computers provide a gate set that can be used to encode "quantum circuits". These circuits describe a problem or an algorithm. For the sake of this work, the circuits will represent matrices. Thus, expressing qubit states as vectors is useful for understanding how gates are used. A single qubit system is represented by the following kets and vectors

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad\qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

IBM provides a set of gates that are used to manipulate the system. Such gates are the Pauli gates X, Y, and Z, the identity gate I, the Hadamard gate H, and the $\sqrt{Z}$ phase gate S.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

The Identity gate and Pauli gates are sufficient to describe any Hermitian matrix. The Hadamard gate is concerned with superposition of states, and the S gate introduces a phase into the state. Another set of gates is the rotation gates, defined as the exponential of the Pauli matrices $\exp(-i\theta P/2) = \cos(\theta/2)I - i\sin(\theta/2)P$

$$RX(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos((\theta/2)) \end{bmatrix} \quad RY(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad RZ(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

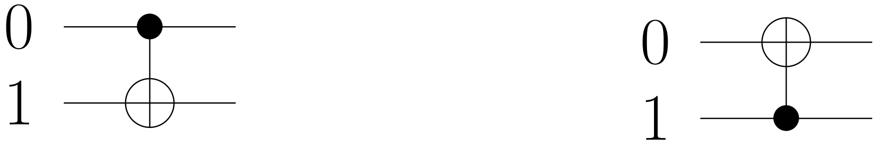
For the state of a two qubit system, the states of single qubits are multiplied via the Kronecker product in order to produce the entangled state

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Likewise, the compound gates matrices for two qubit systems are constructed by Kronecker products of two gates applied to the different qubits

$$X_1 Z_0 |01\rangle = X|0\rangle \otimes Z|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

There are multi-qubit gates too, the most prominent of which is the CNOT gate. This is a gate that is applied to one qubit on the condition that the other qubit is in the $|1\rangle$ state. This gate is involved with quantum entanglement of the two qubits. The inputs for this gate are the control qubit and target qubits.



(a) The circuit for CNOT(0,1)

(b) The circuit for CNOT(1,0)

$$CNOT(c=0,t=1) = I \otimes \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \otimes X \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$CNOT(c=1,t=0) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes I + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The Hamiltonian for a two plaquette lattice with periodic boundary conditions can be written in terms of Pauli gates and the identity gate

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} - x \begin{bmatrix} 0 & 2 & 0 & 2 \\ 2 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 2 & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

$$= \frac{3}{4}(3II - IZ - ZI - ZZ) - \frac{x}{4}(5IX + 5XX + 3ZX - 3YY) \qquad (4.18)$$

In order to use the variational method, the expectation value of the Hamiltonian must be worked out. For an arbitrary two-qubit state, the expectation value of the diagonal(chromoelectric) part of the Hamiltonian is

$$\langle\psi|H|\psi\rangle = \begin{bmatrix} a^* & b^* & c^* & d^* \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \tag{4.19}$$

$$3(P_{01} + P_{10} + P_{11}) = 3(1 - P_{00})$$

where P_{ij} is the probability of measuring the system in state $|ij\rangle$ for $i, j \in \{0, 1\}$. This is not the case for the off-diagonal (chromomagnetic) part of the Hamiltonian because measurements are done in the z-basis. Thus, unitary transformations must be made.

$$\langle G\rangle = \langle\psi|IGI|\psi\rangle = \langle\psi|U^\dagger UGU^\dagger U|\psi\rangle = \langle U\psi|UGU^\dagger|U\psi\rangle = \langle\tilde{\psi}|\tilde{G}|\tilde{\psi}\rangle \tag{4.20}$$

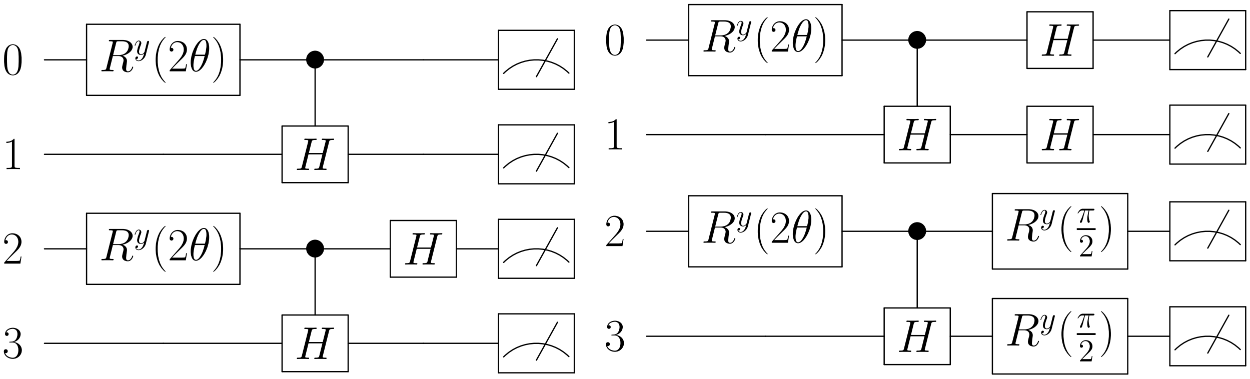
The unitary transformation needed to turn the expectation value of the Pauli X gate into the expectation value of the Pauli Z gate is $U = H$, where H is the Hadamard gate. For the Pauli Y gate,the unitary transformation is $U = HS^\dagger$. Therefore, the expectation value of the magnetic Hamiltonian is:

$$\langle\psi|\mathcal{H}_M|\psi\rangle = \tag{4.21}$$
$$\frac{1}{2}[5(P_{00}^{XX} + P_{11}^{XX}) + 5(P_{00}^{IX} + P_{10}^{IX}) + 3(P_{00}^{ZX} + P_{11}^{ZX}) - 3(P_{00}^{YY} + P_{11}^{YY})]$$

The second part is preparing the initial state. The Hamiltonian is dominated by the chromoelectric part at low x values. At high x, the Hamiltonian becomes dominated by the chromomagnetic part. At low x values, we expect the ground state to be dominated by the state $|00\rangle$. We also expect there to be an equal contribution from the single-plaquette operator states, represented by qubit states $|01\rangle$ and $|11\rangle$. Finally, we expect a negligible contribution from the double-plaquette operator state, $|10\rangle$. Therefore the test state can be written as:

$$|\psi(\theta)\rangle = \cos(\theta)|00\rangle + \sin(\theta)\left(\frac{|01\rangle + |11\rangle}{\sqrt{2}}\right) = CH(c=0,t=1)R^y(2\theta,t=1)|00\rangle \qquad (4.22)$$

where CH is the control Hadamard gate, and θ is the variational parameter that will help determine the approximation for the ground state energy. Once the system is initialized in the test state through the appropriate gate sequences, a second sequence of gates needs to be applied to account for the unitary transformations needed to get the probabilities for off-diagonal matrix elements. For example, both P^{IX} and P^{ZX} have a unitary transformation of $U = H$, therefore both probabilities can be calculated from the same experiment, since the experiment is initialized to $|U\psi(\theta)\rangle$. The four necessary experiments needed were split into two sets of parallel circuits. The first circuit contained the two circuits needed for the P_{ij} and $P_{ij}^{IX/ZX}$ values running in parallel. The P_{ij} circuit used qubits 0 and 1, while the $P_{ij}^{IX/ZX}$ circuit used qubits 2 and 3. The second circuit contained the P_{ij}^{XX} and P_{ij}^{YY} circuits running in parallel, using qubits 0&1 and 2&3 respectively in figure 14. IBM's Melbourne quantum computer was used, with 1000 runs per data point. The results deviate from the exact value by a lot. The simulator results for figure 15 indicate that the reason is due to noise. Using IBM's error mitigation function, and rerunning the data at IBM's Yorktown quantum



(a) The circuits to calculate P and $P^{IX/ZX}$ in parallel

(b) The circuits to calculate P^{XX} and P^{YY} in parallel

Figure 14: The two circuits used to calculate the probabilities needed for the energy of the system

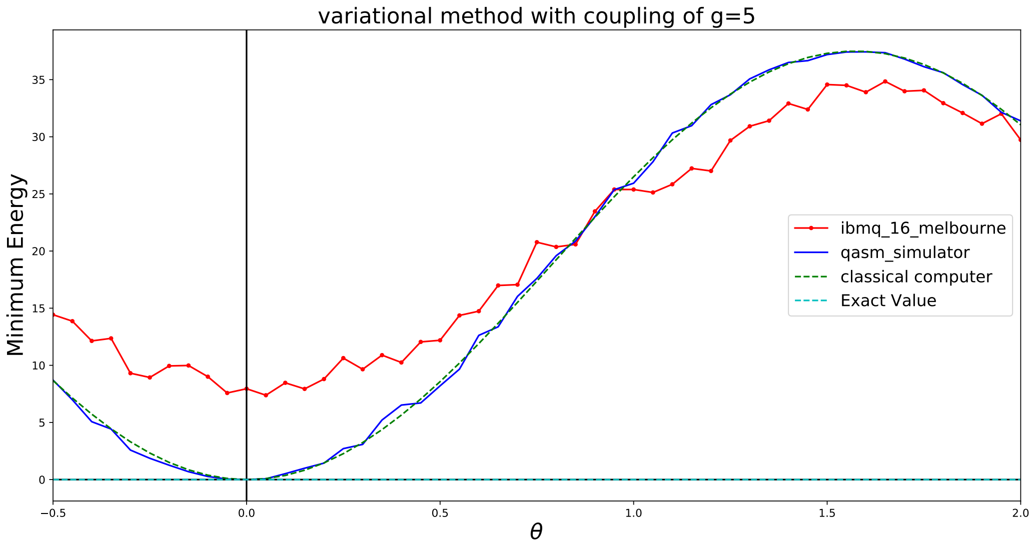


Figure 15: Variational Method for a two plaquette lattice with periodic boundary conditions with $\theta \in [-0.5, 2.00]_{\Delta=0.05}$. IBM's Melbourne was used. The exact value and the classical curve were computed using Matlab.

computer yields figure 16, which was constructed from data created using IBM's Melbourne quantum computer, which has 15 qubits available. 10 qubits were used on the Melbourne quantum computer using a circuit that was a combination of figures 14a and 14b, with a separate circuit for P^{IX} and P^{ZX}, hence the need for 10 qubits. The results from the quantum computer are very far off from the true minimum, where the minimum of the red curve is $E \approx 7.3$. The reason why the data is so far off from the curve calculated using the simulator is due to bit-flip errors and measurement errors. On a real quantum computer, some qubits can flip spins. The quantum computers at IBM are calibrated daily in order to reduce these errors, so the best time to get data is right after calibration. However, the curve in figure 15 was created using data from the quantum computer that were collected many hours after. However, the error due to time at which the data was collected is a very small part of the error in the data. The errors are inherent to the quantum computer, and the error must be mitigated. IBM provides standard measurement error mitigation functions[32] which were used in a second round of data collection, this time on the Yorktown quantum computer

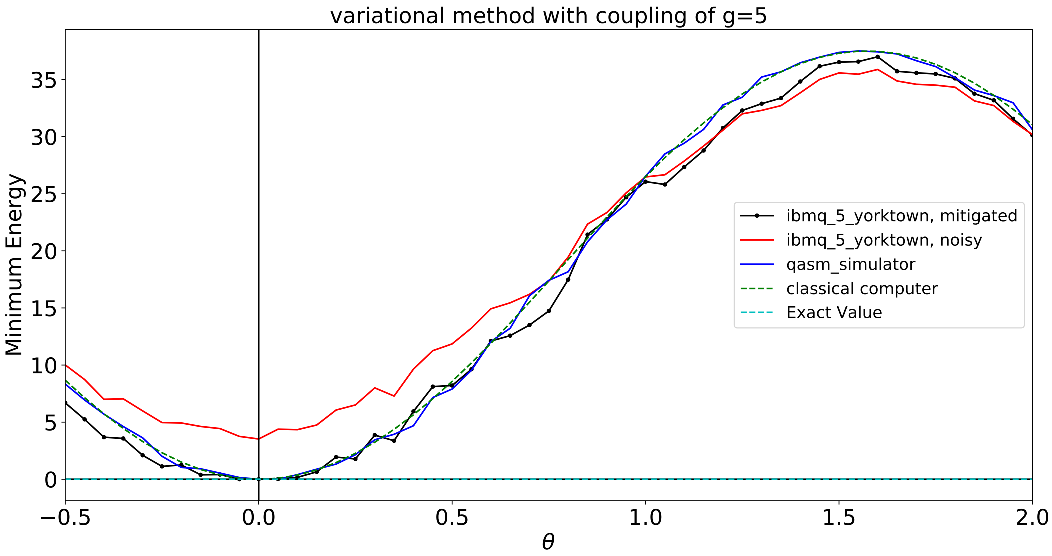


Figure 16: Variational Method with measurement error mitigation for a two plaquette lattice with periodic boundary conditions with $\theta \in [-0.5, 2.00]_{\Delta=0.05}$. IBM's Yorktown was used. The exact value and the classical curve were computed using Matlab.

with the circuits in figures 14a and 14b run separately. The QISKIT measurement error mitigation procedure also sends a few circuits to the quantum computer for the purpose of collecting data on the error rates between qubits, and QISKIT uses those values to construct a correction matrix to the data. The result from the mitigated-data agreed with the theoretical data, with a ground state energy of $E \approx 0$. This demonstrates that gate-based quantum computing can yield impressive results for the purposes of studying lattice gauge theories.

Time evolution on a gate-based quantum computer is a much more computationally costly calculation. The time evolution circuit is calculated by the time evolution operator. The increased difficulty lies in using the least number of gates through ordered Trotterizing to cancel neighboring gates. While it was possible to deal with each term in a sum inside the Hamiltonian as separate circuits, this is not the case for time evolution. The sums turn into products, and produce longer circuits that contain CNOT gates. The effect is a large increase in noise, and less useful data points, as shown in figure 17. Thus, in order to perform reliable time-evolution calculations, extensive error mitigation must be used. In

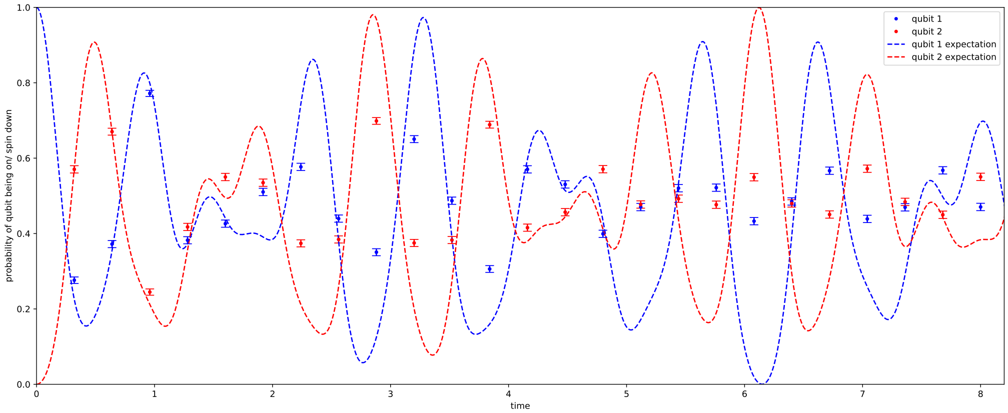


Figure 17: Time evolution of a two plaquette lattice with the initial state $|01\rangle$.

the next section, a more involved error mitigation process will be invented to deal with time-evolution calculations.

5 Self-Mitigating Circuits

This section will be based on the publication titled “Self-mitigating Trotter circuits for SU(2) lattice gauge theory on a quantum computer” [1].

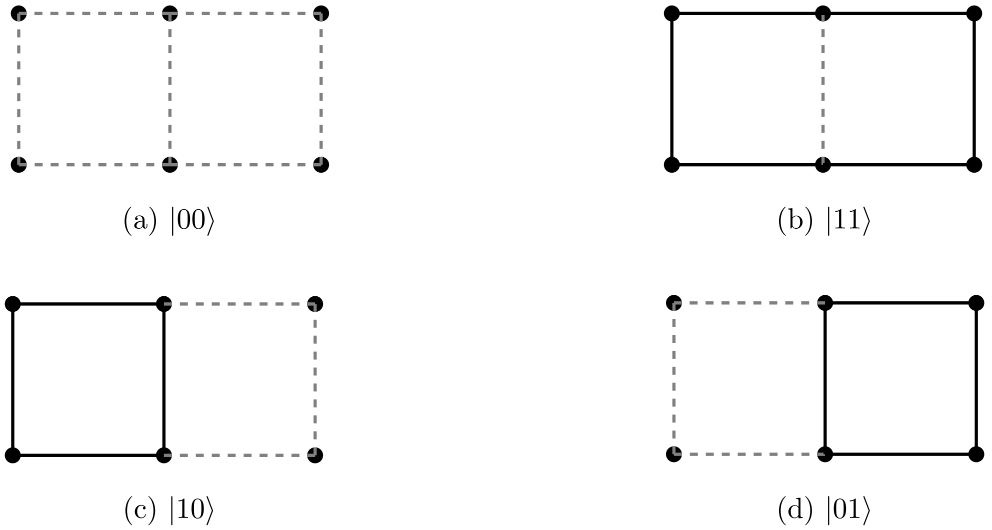
5.1 Common Error Types

The common error types that occur when using the quantum computer can be dealt with using various algorithms. One source of errors is measurement errors, also called readout errors, which occur due to imperfect measurements of the qubit states. The errors appear as readouts of incorrect states of the qubits in our system. For example, the qubit is measured as $|1\rangle$ while it should be $|0\rangle$. The technique for mitigating these errors is called readout error correction. Each of the 2^N states that represent the n qubits are recreated as separate circuits, and multiple runs are executed. Each qubit is measured, and the measurements create a $2^N \times 2^N$ calibration matrix. This matrix is applied to the results of the runs from the desired circuit via sqeuential least squares programming [33, 34].

Another source of errors are gate errors. A special concern is the control gate, as this gate implement quantum entanglement on two or more qubits. Due to the multi-qubit utility of control gates, the errors are naturally higher. An especially prominent source of errors is the $CNOT$ gate. A method that was developed to deal with $CNOT$ errors is Zero-Noise Extrapolation. The concept is quite simple, we can create duplicate circuits where each $CNOT$ gate is replaced by an odd multiple (three $CNOT$s, five $CNOT$, ...), causing a build up of $CNOT$-associated noise. The $CNOT$ noise-free value can then be extrapolated from the line of best fit [35]. The other method of dealing with gate errors is randomized compiling. Gate errors are either coherent or incoherent. Incoherent errors can be modelled as depolarizing noise, which makes it easier to mitigate. Randomized compiling works by turning coherent noise into incoherent noise by surrounding the noisy gates with pairs of Pauli gates. We used randomized compiling by replacing each $CNOT$ gate with one of sixteen combinations of Pauli gates and a $CNOT$ gate, selected randomly [36, 37].

5.2 Time Evolution with Self Mitigation

The notion of using error estimation circuits was inspired by [38]. For a given quantum circuit, a similar circuit called the estimation circuit was constructed. The estimation circuit has a similar structure to the original circuit, and is executed to measure the depolarizing noise rate. The measurements are then used to correct the output of the original circuit. Self mitigation follows the same line of reasoning, but uses the original circuit as the estimation circuit instead of a different circuit. This ensures the highest degree of noise estimation and the least amount of pre-processing in creating a non-identical similar estimation circuit. The physical circuit will be constructed first, then the theory behind self mitigation will be discussed.


(a) $|00\rangle$

(b) $|11\rangle$

(c) $|10\rangle$

(d) $|01\rangle$

Figure 18: The physical lattice states of a two plaquette lattice with periodic boundary conditions

The lattices that will be studied in this section will differ from before by the solid boundary conditions and the Hamiltonian will be different than equation 4.2. The state representing the lattice with the plaquette operator applied to both plaquettes now accounts for the new link that closes the boundary conditions, and has chromoelectric energy of $\frac{9}{4}$. The states used in this section will assign a qubit to each plaquette, which is possible because $j_{max} = \frac{1}{2}$. The links have j values of either $j = 0$ or $j = \frac{1}{2}$, and the shared links will depend on the state of the adjacent plaquettes. The top and bottom links are then dependent on the excitation of the plaquette itself. If the plaquette label in the state describes one of the plaquettes on the ends of the lattice, the qubit further describes the "caps" of the lattice, along with the top and bottom links of said plaquettes. Thus, the Hamiltonian and the states are

$$\frac{2}{g^2}\mathcal{H} = \left[\begin{array}{cccc} 0 & -2x & -2x & 0 \\ -2x & 3 & 0 & -x \\ -2x & 0 & 3 & -x \\ 0 & -x & -x & \frac{9}{4} \end{array}\right] \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \tag{5.1}$$

The matrix can then be translated into a qubit expression

$$\frac{2}{g^2}\mathcal{H} = \frac{3}{8}(7 - 3Z_0 - 3Z_1 - Z_0Z_1) - \frac{x}{2}[(3 + Z_1)X_0 + (3 + Z_0)X_1] \tag{5.2}$$

We seek to exponentiate the Hamiltonian to calculate the time evolution operator. The exponentiated gates are shown below

$$e^{-i\theta X_j} = RX_j(2\theta), \qquad e^{-i\theta Y_j} = RY_j(2\theta), \qquad e^{-i\theta Z_j} = RZ_j(2\theta) \tag{5.3}$$

which are the rotation gates from equation 4.2. Equation 5.2 contains products of gates, which don't have a direct single gate to represent them. The only option is to use unitary transformations to derive the gates that represent the exponentiated Pauli products. The result is $\exp(-i\theta UPU^\dagger) = U\exp(-i\theta P)U^\dagger$. Some useful unitary transformations are

$$XZX = -Z, \qquad XYX = -Y, \qquad HXH = Z, \qquad S^\dagger YS = X \tag{5.4}$$

This can be used to rewrite exponentials of X and of Y into Z rotations: $\exp(-i\theta X) =$

$H\,RZ(2\theta)\,H$, and $\exp(-i\theta Y) = S\,H\,RZ(2\theta)\,H\,S^\dagger$. Another useful result is the unitary transformation needed to go from a product of Z gates to a single Z

$$CX(0,1)\,Z \otimes Z\,CX(0,1) = Z \otimes I \tag{5.5}$$

$$CX(1,0)\,Z \otimes Z\,CX(1,0) = I \otimes Z \tag{5.6}$$

This can be generalized by using indices to specify the qubit that the gates act on. Let $D^0 = |0\rangle\langle 0|$ and $D^1 = |1\rangle\langle 1|$, where $D^0 + D^1 = I$. Then,

$$\begin{aligned} CX_{jk}Z_jZ_kCX_{jk} &= (D^0_jI_k + D^1_jX_k)Z_jZ_k(D^0_jI_k + D^1_jX_k) \\ &= (D^0_jZ_jD^0_j)Z_k + (D^1_jZ_jD^1_j)(X_kZ_kX_k) \\ &= D^0_jZ_k + (-D^1_j)(-Z_k) \\ &= (D^0_j + D^1_j)Z_k \\ &= Z_k \end{aligned}$$

Using these identities, we consider the following three exponentiated gate products:

$$e^{-i\theta Z_jZ_k} = CX_{jk}\,RZ_k(2\theta)\,CX_{jk} \tag{5.7}$$

$$e^{-i\theta Z_jY_k} = CX_{jk}\,RY_k(2\theta)\,CX_{jk} \tag{5.8}$$

$$e^{-i\theta Z_jX_k} = H_k\,CX_{jk}\,RZ_k(2\theta)\,CX_{jk}\,H_k \tag{5.9}$$

This is due to the fact that the X gate is used to change the sign of Z and Y when used as a unitary transformation, as shown in equation 5.4, but $XXX = X$. Since the native entanglement gate is the $CNOT$ gate, any other control gate (like CH or CZ) are composed of other gates containing $CNOT$. This is a crucial point because having Y gates instead of X gates in the Hamiltonian can make it easier to deal with its Z products because they both use $CNOT$ gates as unitary transformations, and neighbouring Z_jZ_k and Z_jY_k products will result in less CX_{jk} gates in the time evolution operator. We use the unitary transformation that changes X to Y in equation 5.4 in order to change the Hamiltonian. Since $S = \sqrt{Z}$, then $SZS^\dagger = Z$, which leaves the chromoelectric part of the Hamiltonian unchanged. The chromomagnetic part will be changed and the modified Hamiltonian is

$$\frac{2}{g^2}\mathcal{H} = \frac{3}{8}(7 - 3Z_0 - 3Z_1 - Z_0Z_1) - \frac{x}{2}[(3 + Z_1)Y_0 + (3 + Z_0)Y_1] \tag{5.10}$$

The second order Suzuki-Trotter approximation of exponential of matrix sums is

$$e^{A+B} \approx e^{\frac{A}{2}} e^{B} e^{\frac{A}{2}} \tag{5.11}$$

The single gates are well understood. The product gates are of concern. The ordering matters, since if we can put exponentials of Z_jZ_k and Z_jY_k next to each other, the exponential is represented by two CX_{jk} gates less than if there was a single gate in between them. Thus, if we consider the product gates only, we get

$$\begin{aligned} e^{-it\mathcal{H}} &= e^{i(xt/4)Z_1Y_0} e^{i(3t/16)Z_0Z_1} e^{i(3xt/4)Y_0} e^{i(9t/16)Z_1} e^{i(9t/16)Z_0} e^{i(3xt/4)Y_1} e^{i(x/2)Z_0Y1} \\ &\times e^{i(3xt/4)Y_1} e^{i(9t/16)Z_0} e^{i(9t/16)Z_1} e^{i(3xt/4)Y_0} e^{i(3t/16)Z_0Z_1} e^{i(xt/4)Z_1Y_0} \end{aligned} \tag{5.12}$$

This strategic choice results in the following ordered gate product for a single Trotter step

$$\begin{gathered} CX_{10}RY_0(-\frac{xt}{2})RZ_0(-\frac{3t}{4})CX_{10}RY_0(-\frac{3xt}{2})RZ_1(-\frac{9t}{4})RZ_0(-\frac{9t}{4})RY_1(-3xt)CX_{01}RY_1(-xt)CX_{01} \\ RY_1(-3xt)RZ_0(-\frac{9t}{4})RZ_1(-\frac{9t}{4})RY_0(-\frac{3xt}{2})RZ_0(-\frac{3t}{4})CX_{10}CX_{10}RY_0(-\frac{xt}{2}) \end{gathered} \tag{5.13}$$

The total time t is divided into small time intervals dt in order to compensate for this approximation. The time evolution calculations in [14] were composed of one or two Trotter steps, containing a mere six $CNOT$ gates. Self mitigation aims to increase the capabilities. To implement self-mitigation, there must be two runs. The first run is called the "physics

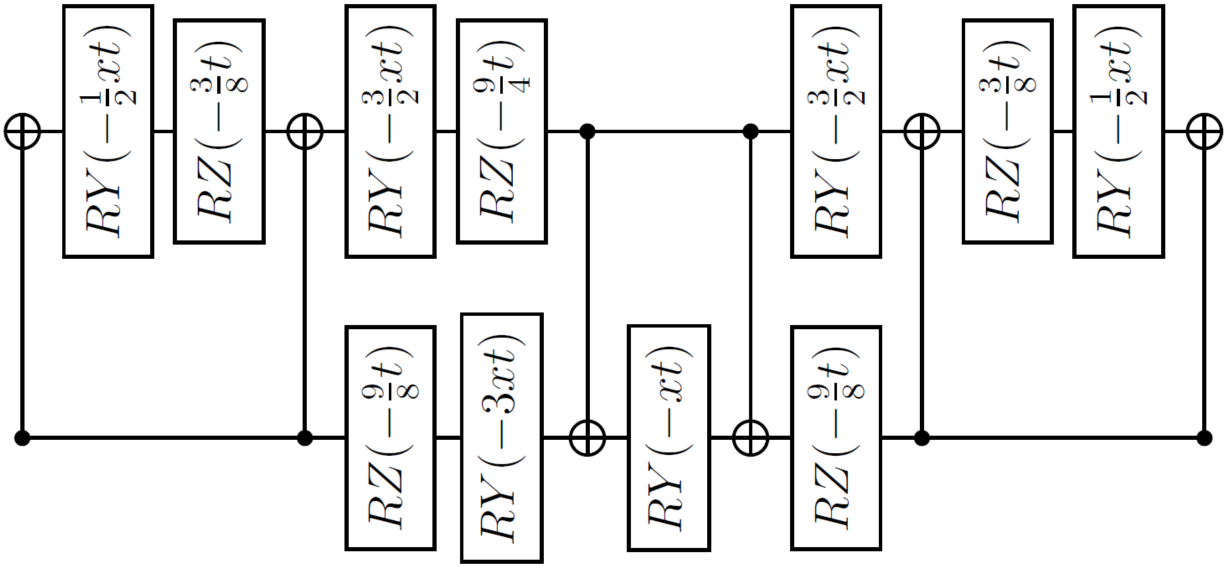


Figure 19: A second-order Suzuki-Trotter step for the two plaquette lattice. The first and the last CX gates cancel with the CX gates of the neighbouring Trotter steps, leaving four CX per Trotter step. This figure also appears in Ref. [1].

run", which applies N Trotter steps in the forward time direction to arrive at the final state at time t. The second run is called the "mitigation run", where $N/2$ Trotter steps are applied in the forward time direction, and then $N/2$ Trotter steps are applied in the backward time direction ($dt \to -dt$). On an ideal quantum computer, the mitigation run would return the system to the initial state. Measuring the qubits at the end provides a measure of the acquired errors along the forward-backward time evolution trip. The acquisition of the same amount of error as the physics circuit is the motivation behind using the physics circuit for the mitigation run. Randomized compiling is also applied to both physics and mitigation runs. Each $CNOT$ gate is replaced by one of 16 randomized versions of the gates in appendix A.4. The extra Pauli gates introduced by randomized compiling can be combined with those from neighboring $CNOT$ gates and with the rotation gates in between to reduce the circuit size.

This is encoded on IBM's quantum computer using IBM's python package Qiskit. The circuit in figure 19 is a single time-step. Our code submitted 300 runs with 10^4 hits per run. The 300 runs are sent to the IBM hardware as a single python list to ensure the physics and mitigation circuits run consecutively. This is important because hardware conditions affect the results due to the calibration of the hardware. Runs occurring near the calibration time produce less noisy results than runs occurring a long time after the hardware calibration. Measurement mitigation/ readout error mitigation is also used, but we wrote our own code for it. The upper panel of figure 20 displays the averaged results of the physics run and the averaged results from the mitigation run at $x = 2.0$. For each data point, a statistical error bar for 148 runs is obtained from 1480 bootstrap samples and then combined in quadrature with the statistical error from 10^4 hits per run. Ideally, the mitigation results would indicate

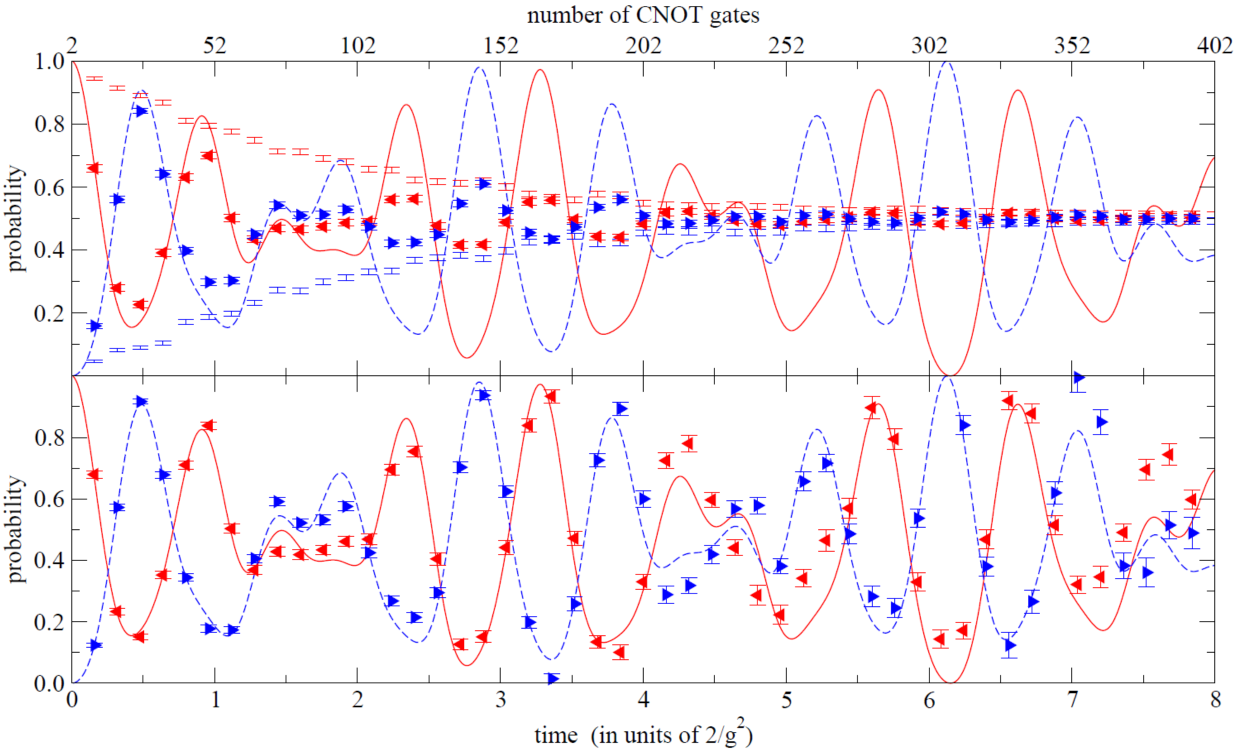


Figure 20: Time evolution of an excitation on a two-plaquette lattice from an initial state of a excited left plaquette □□. The inverse gauge coupling is $x = 2.0$ and time step $dt = 0.08$. The red solid (blue dashed) curve is the exact probability of the left (right) plaquette being measured to have $j = \frac{1}{2}$. For the upper panel: The red left-pointing (blue right-pointing) triangles are the physics data and the red (blue) error bars without symbols are the mitigation data from the same circuit but with half the steps forward in time and then half backward in time. All these points were computed on **ibm_lagos**. For the lower panel: The triangles are the physics results obtained by applying the self-mitigation correction to the data from the upper panel. The results were obtained from 148 physics runs and 148 mitigation runs, each with different $CNOT$ randomizations. This figure also appears in Ref. [1].

that there is a 100% probability of returning to the initial state. That is, the probability of measuring a qubit in state $|1\rangle$ would be 1 for the left plaquette, and 0 for the right plaquette. Instead, we observe that there is a smooth decline approaching the line at $\frac{1}{2}$, which is called pure noise because there would be a 50% chance of being in either state. The raw physics results are bounded by the mitigation results since the physics probabilities are between 0 and 1.

If self mitigation and randomized compiling leads to incoherent noise that is independent of the $dt \to -dt$ sign flip in the rotation gates of the mitigation circuit, then we can construct a simplistic model of error that can help approximate the true result.

$$\frac{P_{\text{true}}^{(\text{physics})} - \frac{1}{2}}{P_{\text{computed}}^{(\text{physics})} - \frac{1}{2}} = \frac{P_{\text{true}}^{(\text{mitigation})} - \frac{1}{2}}{P_{\text{computed}}^{(\text{mitigation})} - \frac{1}{2}} \tag{5.14}$$

The true and computed probabilities are compared to the probability of pure noise. In this case, there are only two possibilities, which means that $P_{\text{noise}} = 0.5$. The idea is that the ratio of the true value P_{true} and random noise difference to the computed value P_{computed} and random noise difference is equivalent between the physics and the mitigation circuits. The computed probabilities are obtained from the runs on the quantum hardware. The true value for the physics run is the desired value, and the true value for the mitigation run is either 0 or 1, depending on the initial state. Thus, the true physics result is therefore

$$P_{\text{true}}^{(\text{physics})} = \frac{1}{2} + \left(P_{\text{computed}}^{(\text{physics})} - \frac{1}{2}\right)\left(\frac{P_{\text{true}}^{(\text{mitigation})} - \frac{1}{2}}{P_{\text{computed}}^{(\text{mitigation})} - \frac{1}{2}}\right) \tag{5.15}$$

The lower panel of figure 20 shows the true values plotted. Comparison between the two panels show the significant extension of the range of data that agree with the classical curves.

5.3 Travelling Excitation

This section will use self mitigation to create travelling excitation plots. While figure 20 was calculated with $x = 2.0$, which means that the chromomagnetic part of the Hamiltonian overshadowed the chromoelectric part. A choice of $x \lessapprox 1$ would put the emphasis on the chromoelectric eigenstates while still allowing for transition between the states. The propagation time will be larger for smaller x, becoming infinite at $x = 0$. This is predicted by the dependence of the off-diagonal x-dependent terms, which act as transition elements between the chromoelectric eigenstates. Thus, a choice of $x = 0.8$ is made so that excitation propagation can occur within an observable time window. As we evaluate the time evolution at larger times, the $CNOT$ gates accumulate, approaching 300 $CNOT$ gates, and the data deviates from the classically calculated exact curves.

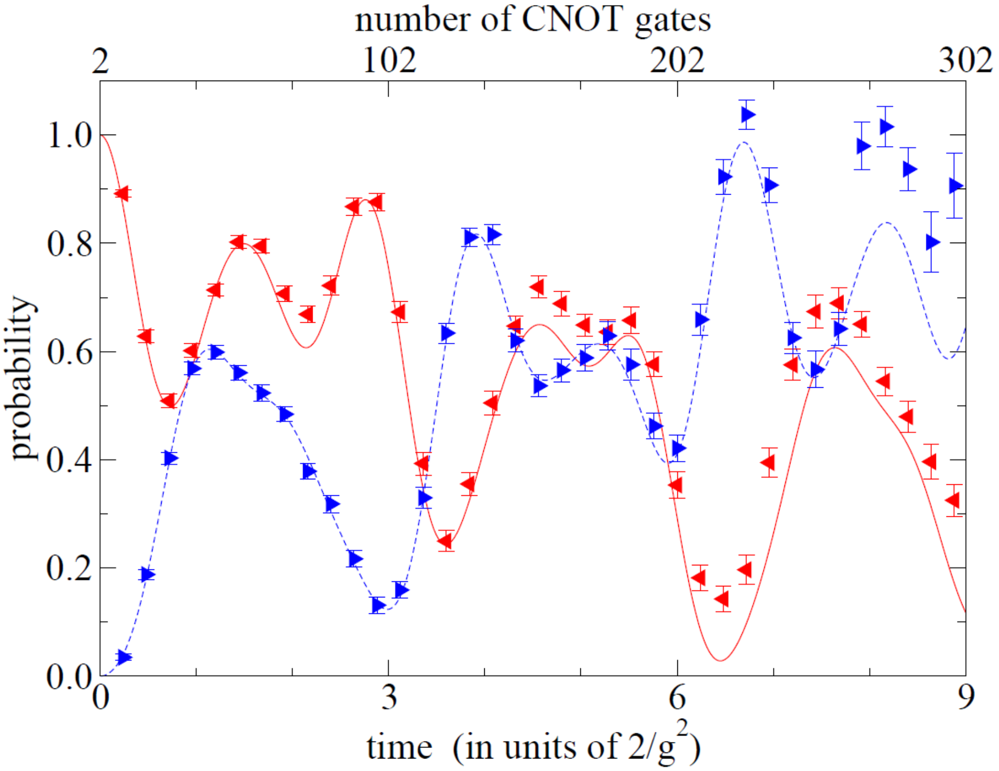


Figure 21: Excitation moving from one plaquette to another on a two-plaquette lattice with $x = 0.8$ and $dt = 0.12$. The initial state is an excited left plaquette, □□. The red solid (blue dashed) curve is the exact probability of the left(right) plaquette being measured to have $j = \frac{1}{2}$. The red left-pointing (blue right-pointing) triangles are the calculations on **ibm_lagos** after self mitigation. This figure also appears in Ref. [1].

5.4 Time evolution on a Five Plaquette Lattice

This section will consider a five-plaquette lattice. This requires 5 qubits. The **ibm_lagos** hardware provides 7 qubits. However, the longest string of nearest-neighbor qubits is of length 5, thus the choice of a five-plaquette lattice.

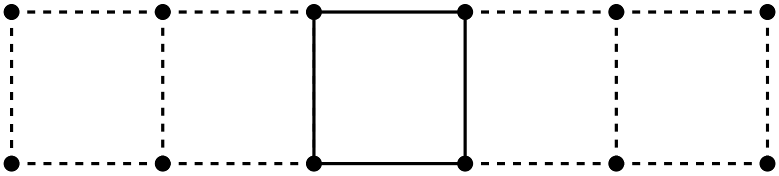


Figure 22: The initial state for the time evolution of a five-plaquette lattice, $|00100\rangle$.

For a general N plaquette one dimensional lattice, the Hamiltonian can be generalized as

$$\mathcal{H}_E^{(N)}(x) = \frac{3}{8}(3N+1) - \frac{9}{8}(Z_0 + Z_{N-1}) - \frac{3}{4}\sum_{n=1}^{N-2} Z_n - \frac{3}{8}\sum_{n=0}^{N-2} Z_n Z_{n+1} \tag{5.16}$$

$$\mathcal{H}_M^{(N)}(x) = -\frac{x}{2}\left((3+Z_1)X_0 + (3+Z_{N-2})X_{N-1} + \sum_{n=1}^{N-2}(3+Z_{n-1})(\frac{X_n}{4})(3+Z_{n+1})\right) \tag{5.17}$$

This is used to construct the Hamiltonian for a five-plaquette lattice with $N = 5$. As per the previous treatment, the X gates will be changed to Y gates. A new occurrence must be addressed. There are terms that contain three qubits in the chromomagnetic Hamiltonian. This can be easily dealt with using the prescription from equation 5.9 for the two gate exponentials to deal with the three gate exponential

$$e^{-i\theta Z_j Y_k Z_l} = CX_{lk} CX_{jk} RY_k(2\theta) CX_{jk} CX_{lk} \tag{5.18}$$

A strategic ordering of the gates in the time-evolution operator can reduce the $CNOT$ gates to 16 $CNOT$ gates for a first-order Trotter step and 22 $CNOT$ gates for the second-order Trotter step. The result is shown in figure 23. For the two-plaquette lattice, we use a constant time step $|dt|$ while changing the number of Trotter steps, but for the five-plaquette

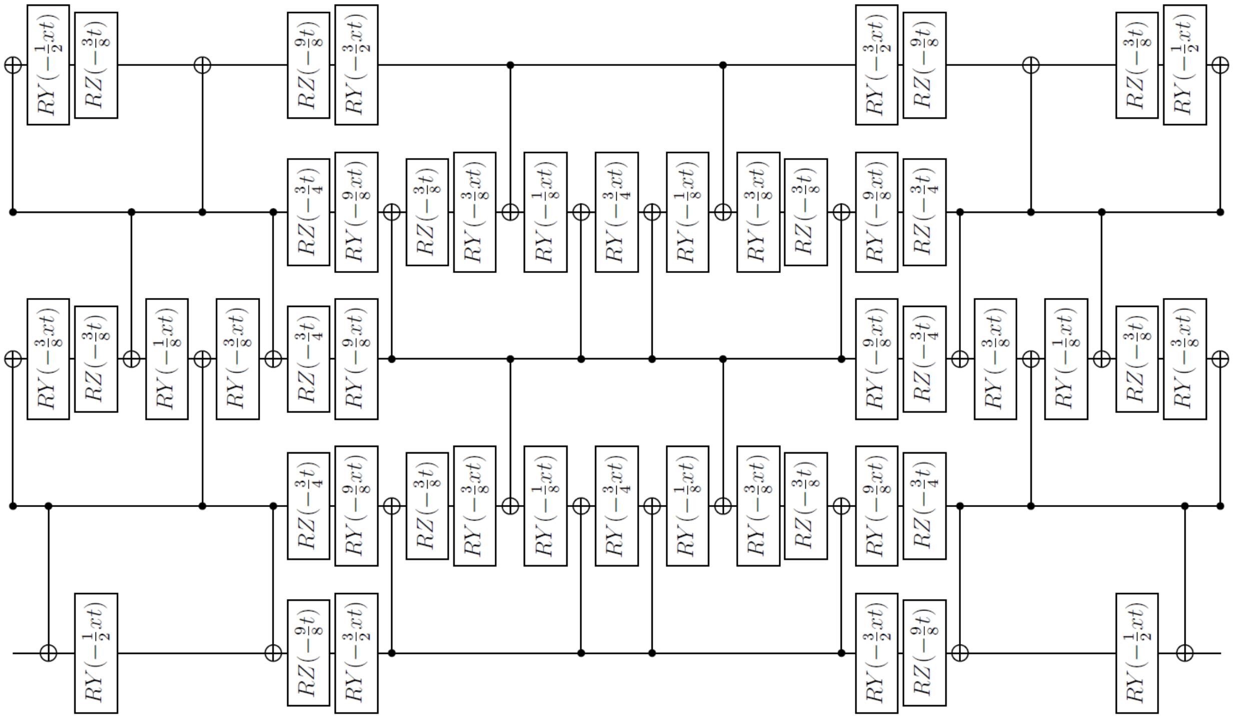


Figure 23: A second order Suzuki-Trotter step for the five-plaquette lattice. This figure also appears in Ref. [1].

lattice we do the opposite. We use various time step sizes while keeping the number of Trotter steps to four. Therefore, the total circuit contains only 94 randomized $CNOT$ gates. These circuits are sent to the hardware with 300 runs(each with 10^4 hits), where $2^5 = 32$ runs are used for measurement-error mitigation, 134 runs for the physics calculation, and 134 for the self-mitigation circuit. Four separate jobs were used per data point and analyzed separately. The averaged results are shown in figure 24. The zero-noise extrapolation method was applied to calculate the open symbols. Each $CNOT$ gate is replaced by a triplet of identical $CNOT$ gates. A linear extrapolation of the triplet results and the filled symbols results ($CNOT$ singlets) are used to calculate the expected zero-noise result.

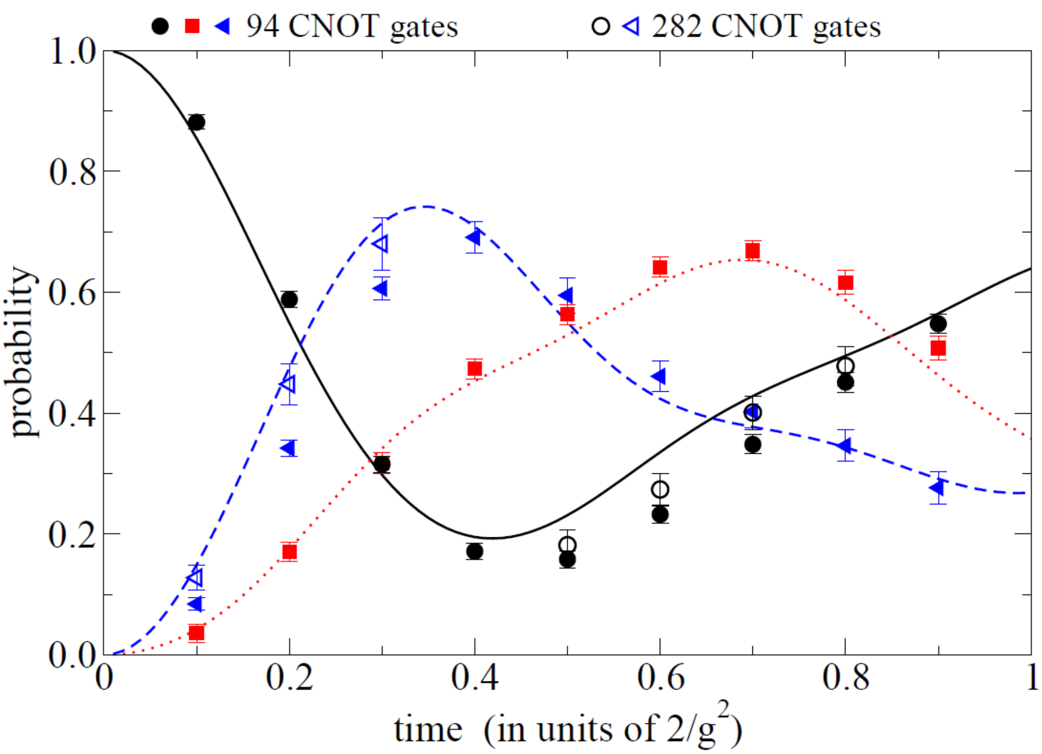


Figure 24: Time evolution on a five-plaquette lattice from the initial state of figure 22 with $x = 2.0$. The black solid, red dotted, and blue dashed curves are respectively the exact probability of measuring $j = \frac{1}{2}$ for the center plaquette, its neighbor, and the outer plaquette, □□□□□. The black circles, red squares, and blue triangles are the corresponding calculations on the **ibm_lagos** quantum processor after self mitigation. Each filled data point uses four second-order Trotter steps and 94 $CNOT$ gates. Open data points are augmented by zero-noise extrapolation. This figure also appears in Ref. [1].

This work has shown a new method for error correction called self-mitigation. This method was used to study time evolution of $SU(2)$ lattice gauge theory and has shown significant improvements to the range of coherent data. From weaker inverse coupling to stronger inverse coupling, useful physical results were produced using self-mitigation. This method is uniquely suited for time evolution, and the use of the physics circuit in the mitigation run, as opposed to a similar circuit containing the most noisy gates, and ensures very close approximation to the errors accumulated in the physics circuit. Combining other mitigation techniques with self-mitigation has shown even further improvements to noise reduction. Zero noise extrapolation is especially useful to further mitigate the errors. However, randomized compiling is crucial to our scheme. Randomized compiling converts gate errors from coherent errors into incoherent errors, which can then be dealt with.

6 Pauli Decomposition Algorithm

This section will revolve around the Pauli decomposition algorithm that was written for the purpose of dealing with large Hamiltonians. This will be useful when attempting to consider larger $j_{\max}$ values, or different lattice geometries that extend to the second dimension. The structure of this section is to firstly discuss the one qubit example, then the two qubit example will be studied from which symmetry techniques will be developed that allow for the Pauli decomposition of any matrix.

A quick review of Pauli decomposition for 2×2 matrices will be shown. This will motivate the symmetries of diagonal and anti-diagonal matrices. Any 2×2 matrix can be decomposed into the weighted sum of four basis matrices M_i that I shall call the basis matrices,

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} = \sum_{i=1}^{4} m_i M_i = a_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + a_2 \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + a_3 \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + a_4 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Since we are working with quantum gates, it is natural for us to use Pauli basis, which is composed of the Pauli matrices and the identity matrix:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = i\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

and the basis matrices can be rewritten in terms of the Pauli matrices

$$M_1 = \frac{I+Z}{2} \qquad M_2 = \frac{X+iY}{2} \qquad M_3 = \frac{X-iY}{2} \qquad M_4 = \frac{I-Z}{2}$$

Therefore, we can write any 2×2 matrix as:

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} = \sum_{i=1}^{4} p_i P_i = \frac{a_1+a_4}{2} I + \frac{a_2+a_3}{2} X + \frac{a_2-a_3}{2} iY + \frac{a_1-a_4}{2} Z$$

Separating the diagonal and the anti-diagonal parts of the matrix sheds light on an obvious property of Pauli matrices. The diagonal part is written in terms of the mixing of the I and Z gates, since they are purely diagonal

$$\frac{a_1 + a_4}{2} I + \frac{a_1 - a_4}{2} Z = \frac{1}{2} \begin{bmatrix} I & Z \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_4 \end{bmatrix}$$

The anti-diagonal part is described by the anti-diagonal Pauli gates X and Y

$$\frac{a_2 + a_3}{2} X + \frac{a_2 - a_3}{2} iY = \frac{1}{2} \begin{bmatrix} X & iY \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \end{bmatrix}$$

This rewriting of the diagonal and anti-diagonal in terms of quadratic forms will be useful in generalizing this approach. A general Hermitian 4 × 4 matrix can be written as

$$\left[\begin{array}{cc|cc} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ \hline a_9 & a_{10} & a_{11} & a_{12} \\ a_{13} & a_{14} & a_{15} & a_{16} \end{array}\right]$$

To write the Pauli decomposition of this matrix, products of Pauli matrices are needed. Following the same line of thought from before, products of the diagonal Pauli gates will give

$$\frac{a_1 + a_6 + a_{11} + a_{16}}{4} II + \frac{a_1 - a_6 + a_{11} - a_{16}}{4} IZ + \frac{a_1 + a_6 - a_{11} - a_{16}}{4} ZI + \frac{a_1 - a_6 - a_{11} + a_{16}}{4} ZZ$$

$$= \frac{1}{4} \begin{bmatrix} II & IZ & ZI & ZZ \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_6 \\ a_{11} \\ a_{16} \end{bmatrix}$$

This contains similarities to the 2×2 case, and a pattern emerges. The first matrix in the middle is recognized to be the Hadamard gate in the 2×2 case, and the Kronecker product of two Hadamard gates. Thus, we begin the generalization to N qubits by introducing a $2^N \times 2^N$ matrix B^N that generates a Kronecker product of N Hadamard gates

$$B^{(N)} = 2^{-N/2} \bigotimes_{k=1}^{N} H \tag{6.1}$$

The second observation is how the diagonal of a matrix is always described by the diagonal gates. Likewise, the anti-diagonal is described by the anti-diagonal gates

$$\begin{bmatrix} XX & iXY & iYX & -YY \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a_4 \\ a_7 \\ a_{10} \\ a_{13} \end{bmatrix}$$

The other off-diagonal elements are described by products of diagonal and anti-diagonal gates. To generalize this, we define the following

$$A_0^{(1)} = \begin{bmatrix} I & Z \end{bmatrix} \tag{6.2}$$

$$A_1^{(1)} = \begin{bmatrix} X & iY \end{bmatrix} \tag{6.3}$$

where the superscript is the number of qubits, and the subscript is 0 for diagonal and 1 for anti-diagonal. We can create the row vector containing the gates needed via Kronecker products of the A^i row vectors.

$$A_{00}^{(2)} \equiv A_0^{(1)} \otimes A_0^{(1)} = \begin{bmatrix} II & IZ & ZI & ZZ \end{bmatrix}$$
$$A_{01}^{(2)} \equiv A_0^{(1)} \otimes A_1^{(1)} = \begin{bmatrix} IX & iIY & ZX & iZY \end{bmatrix}$$
$$A_{10}^{(2)} \equiv A_1^{(1)} \otimes A_0^{(1)} = \begin{bmatrix} XI & XZ & iYI & iYZ \end{bmatrix}$$
$$A_{11}^{(2)} \equiv A_1^{(1)} \otimes A_1^{(1)} = \begin{bmatrix} XX & iXY & iYX & -YY \end{bmatrix}$$

The third observation is the column vectors on the right of the quadratic forms. They contain information from the matrix being decomposed. For the 1 qubit case, the diagonal element can be extracted using the **diag** function, which creates a column vector containing the N diagonal elements of an $2^N \times 2^N$ input matrix. For the off diagonal elements, a rotation is performed using the X gate to position the anti-diagonal elements on the diagonal, then **diag** is used for extraction. Thus, the result for a 2 × 2 matrix M is

$$\mathbf{diag}(MI) = \begin{bmatrix} a_1 \\ a_4 \end{bmatrix}$$
$$\mathbf{diag}(MX) = \begin{bmatrix} a_2 \\ a_3 \end{bmatrix}$$

This can be generalized by introducing the following

$$C_{p_1 p_2 \ldots p_N}^{(N)} = \bigotimes_{k=1}^{N} X^{p_k}$$

If we want the coefficients associated with $A^{(2)}_{01}$, then $C^{(2)}_{01} = X^0 \otimes X^1 = IX$. Thus,

$$A^{(N)}_{p_1 \ldots p_N} B^{(N)} \mathbf{diag}(MC^{(N)}_{p_1 \ldots p_N}) \tag{6.4}$$

where M is the input matrix. To test this, consider a Hermitian 4×4 matrix. The algorithm will first multiply sectors of the matrix by $C^{(2)}_{ij}$

$$MC^{(2)}_{00} = \left[\begin{array}{cc|cc} a_1 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 \\ \hline 0 & 0 & a_3 & 0 \\ 0 & 0 & 0 & a_4 \end{array}\right] \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{cc|cc} a_1 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 \\ \hline 0 & 0 & a_3 & 0 \\ 0 & 0 & 0 & a_4 \end{array}\right]$$

$$MC^{(2)}_{01} = \left[\begin{array}{cc|cc} 0 & b_1 & 0 & 0 \\ b_1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & b_2 \\ 0 & 0 & b_2 & 0 \end{array}\right] \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right] = \left[\begin{array}{cc|cc} b_1 & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 \\ \hline 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & b_2 \end{array}\right]$$

$$MC^{(2)}_{10} = \left[\begin{array}{cc|cc} 0 & 0 & c_1 & 0 \\ 0 & 0 & 0 & c_2 \\ \hline c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \end{array}\right] \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] = \left[\begin{array}{cc|cc} c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \\ \hline 0 & 0 & c_1 & 0 \\ 0 & 0 & 0 & c_2 \end{array}\right]$$

$$MC^{(2)}_{11} = \left[\begin{array}{cc|cc} 0 & 0 & 0 & d_1 \\ 0 & 0 & d_2 & 0 \\ \hline 0 & d_2 & 0 & 0 \\ d_1 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{cc|cc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right] = \left[\begin{array}{cc|cc} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ \hline 0 & 0 & d_2 & 0 \\ 0 & 0 & 0 & d_1 \end{array}\right]$$

then the next step is the **diag** function

$$diag(MC_{00}^{(2)}) = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}^T$$
$$diag(MC_{01}^{(2)}) = \begin{bmatrix} b_1 & b_1 & b_2 & b_2 \end{bmatrix}^T$$
$$diag(MC_{10}^{(2)}) = \begin{bmatrix} c_1 & c_2 & c_1 & c_2 \end{bmatrix}^T$$
$$diag(MC_{11}^{(2)}) = \begin{bmatrix} d_1 & d_2 & d_2 & d_1 \end{bmatrix}^T$$

The python code for this algorithm is in appendix A.5

7 Theoretical Extensions

This section contains attempts to go beyond the systems studied in this thesis. The first part will discuss weaker coupling systems, characterized by $j_{\text{max}} = 1$. The second part will show attempts at describing a four-plaquette lattice in the shape of a square, or a cloverleaf. This section is meant to set up the theoretical foundations for a proper computational study of these new extensions.

7.1 Weaker Coupling Two-Plaquette Lattice

The previous studies focused on $j_{\text{max}} = \frac{1}{2}$, which allowed for mapping plaquettes onto individual qubits. The links were either excited ($j = \frac{1}{2}$) or not excited ($j = 0$). To probe a weaker coupling domain, the Hamiltonian will be truncated at $j_{\text{max}} = 1$. This introduces a new type of link, which will be referred to as the double-excited link. The shared links can no longer be determined by the states of the adjacent plaquettes. This means that there is no longer a one-to-one mapping between plaquettes and qubits.

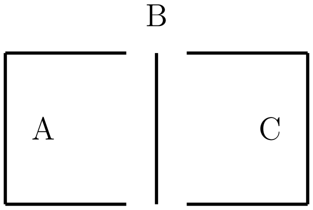


Figure 25: The labels for a two-plaquette lattice with $j_{\text{max}} = 1$ and $A = 2j_A + 1$, $B = 2j_B + 1$, and $C = 2j_C + 1$.

The states are constructed via the repeated application of the plaquette operator to the vacuum state. The lattices are visualized in appendix A.6. The Hamiltonian of the two plaquette system is thus

$$\left|\begin{array}{ccccccccccc||c}
\mathbf{0} & -2x & -2x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 111 \\
-2x & \mathbf{3} & 0 & -x & -\sqrt{3}x & -2x & 0 & 0 & 0 & 0 & 0 & 221 \\
-2x & 0 & \mathbf{3} & -x & -\sqrt{3}x & 0 & -2x & 0 & 0 & 0 & 0 & 122 \\
0 & -x & -x & \mathbf{\frac{9}{2}} & 0 & 0 & 0 & -\sqrt{3}x & -\sqrt{3}x & 0 & 0 & 212 \\
0 & -\sqrt{3}x & -\sqrt{3}x & 0 & \mathbf{\frac{13}{2}} & 0 & 0 & -\frac{x}{3} & -\frac{x}{3} & 0 & 0 & 232 \\
0 & -2x & 0 & 0 & 0 & \mathbf{8} & 0 & -\frac{2\sqrt{3}x}{3} & 0 & 0 & 0 & 331 \\
0 & 0 & -2x & 0 & 0 & 0 & \mathbf{8} & 0 & -\frac{2\sqrt{3}x}{3} & 0 & 0 & 133 \\
0 & 0 & 0 & -\sqrt{3}x & -\frac{x}{3} & -\frac{2\sqrt{3}x}{3} & 0 & \mathbf{9} & 0 & -\frac{2\sqrt{3}x}{3} & -\frac{4x}{3} & 322 \\
0 & 0 & 0 & -\sqrt{3}x & -\frac{x}{3} & 0 & -\frac{2\sqrt{3}x}{3} & 0 & \mathbf{9} & -\frac{2\sqrt{3}x}{3} & -\frac{4x}{3} & 223 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{2\sqrt{3}x}{3} & -\frac{2\sqrt{3}x}{3} & \mathbf{12} & 0 & 313 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{4x}{3} & -\frac{4x}{3} & 0 & \mathbf{14} & 333
\end{array}\right|$$

This 11×11 matrix can not be mapped onto qubits because it is not a $2^N \times 2^N$ matrix. Common methods of dealing with this issue is through addition of a non-physical sector containing only zeroes. This would bring the size of the matrix to 16×16, and hence would require four qubits. Another option is to further truncate the matrix down to 8×8. This would defeat the purpose of exploring the $j_{\max} = 1$ sector.

This section proposes a different approach, which is to attempt to exploit the translation symmetry of the system. There are three pairs of states that can be combined into symmetric and anti-symmetric states.

$$|221\rangle,\ |122\rangle \to \frac{|221\rangle \pm |122\rangle}{\sqrt{2}}$$
$$|331\rangle,\ |133\rangle \to \frac{|331\rangle \pm |133\rangle}{\sqrt{2}}$$
$$|322\rangle,\ |223\rangle \to \frac{|322\rangle \pm |223\rangle}{\sqrt{2}}$$

This allows for the block diagonalization of the Hamiltonian matrix into a 8×8 vacuum block and a 3 × 3 block. The vacuum block is therefore

$$\left|\begin{array}{cccccccc||c}
\mathbf{0} & -2\sqrt{2}x & 0 & 0 & 0 & 0 & 0 & 0 & |111\rangle \\
-2\sqrt{2}x & \mathbf{3} & -\sqrt{2}x & -\sqrt{6}x & -2x & 0 & 0 & 0 & \frac{|221\rangle+|122\rangle}{\sqrt{2}} \\
0 & -\sqrt{2}x & \mathbf{\frac{9}{2}} & 0 & 0 & -\sqrt{6}x & 0 & 0 & |212\rangle \\
0 & -\sqrt{6}x & 0 & \mathbf{\frac{13}{2}} & 0 & \frac{-\sqrt{2}x}{3} & 0 & 0 & |232\rangle \\
0 & -2x & 0 & 0 & \mathbf{8} & -\frac{2x}{\sqrt{3}} & 0 & 0 & \frac{|331\rangle+|133\rangle}{\sqrt{2}} \\
0 & 0 & -\sqrt{6}x & -\frac{\sqrt{2}x}{3} & -\frac{2x}{\sqrt{3}} & \mathbf{9} & -\frac{2\sqrt{6}x}{3} & -\frac{2\sqrt{8}x}{3} & \frac{|322\rangle+|223\rangle}{\sqrt{2}} \\
0 & 0 & 0 & 0 & 0 & -\frac{2\sqrt{6}x}{3} & \mathbf{12} & 0 & |313\rangle \\
0 & 0 & 0 & 0 & 0 & -\frac{2\sqrt{8}}{3} & 0 & \mathbf{14} & |333\rangle
\end{array}\right| \tag{7.1}$$

This allows for the utilization of three qubits without sacrificing sections of the matrix that include interesting high energy physics. This symmetric block is ideal for ground-state energy calculations and vacuum expectation value calculations. This form focuses on the amount of excitation rather than the location of the excitation in the lattice. For time evolution, this is not useful for observing propagation of the excitation, specially because the notion of double-excitation is now an occurrence. The utility of the symmetric block lies in the ease of studying the change between the different excitation types (i.e. from single-excited lattice to double-excited).

The gate representation of the symmetric block is obtained using the Pauli decomposition algorithm. Due to the size of the matrix, the gate representation is shown below in pieces as the coefficients of the gates with the respective gates on the right

$$\begin{bmatrix}7.1250\\-1.0000\\-2.1250\\0\end{bmatrix}\begin{matrix}III\\IIZ\\IZI\\IZZ\end{matrix}, \quad \begin{bmatrix}-3.6250\\-0.2500\\0.1250\\-0.2500\end{bmatrix}\begin{matrix}ZII\\ZIZ\\ZZI\\ZZZ\end{matrix}, \quad \begin{bmatrix}-0.9958\\-0.9958\\-0.4184\\-0.4184\end{bmatrix}\begin{matrix}IIX\\IZX\\ZIX\\ZZX\end{matrix}, \quad \begin{bmatrix}-1.0838\\1.0838\\-0.1410\\0.1410\end{bmatrix}\begin{matrix}IXI\\IXZ\\ZXI\\ZXZ\end{matrix},$$

$$\begin{bmatrix}-0.7618\\0.7618\\0.0547\\-0.0547\end{bmatrix}\begin{matrix}IXX\\-IYY\\ZXX\\-ZYY\end{matrix}, \quad \begin{bmatrix}-0.5000\\-0.5000\\-0.5000\\-0.5000\end{bmatrix}\begin{matrix}XYX\\-YIY\\XZX\\-YZY\end{matrix}, \quad \begin{bmatrix}-0.1179\\0.1179\\0.1179\\-0.1179\end{bmatrix}\begin{matrix}XXI\\XXZ\\-YYI\\-YYZ\end{matrix}, \quad \begin{bmatrix}-0.6124\\-0.6124\\0.6124\\0.6124\end{bmatrix}\begin{matrix}XXX\\-YXY\\-YYX\\-XYY\end{matrix}$$

Now that the coefficients are calculated, the goal in the future would be to construct the time evolution operator and calculate the results on the quantum computer.

7.2 Cloverleaf Lattice

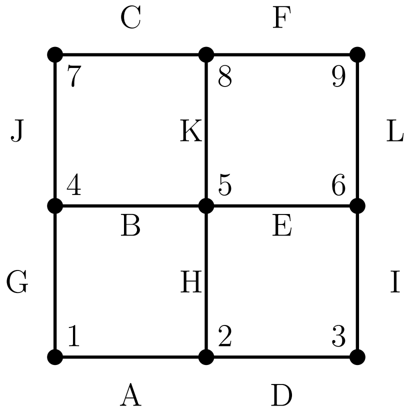


Figure 26: The cloverleaf lattice

The other extension to the theory is going from one-dimensional lattices to two-dimensional ones. The cloverleaf lattice is a four-plaquette lattice arranged in a square. A new type of site appears that was not discussed before. This four-linked site results in different representations depending on the grouping of links. I will group the positive x and y axes to get j_{12}, then I will group the negative x and y axes to get j_{34}, and then I will add the results to get J. Thus, the resultant state is

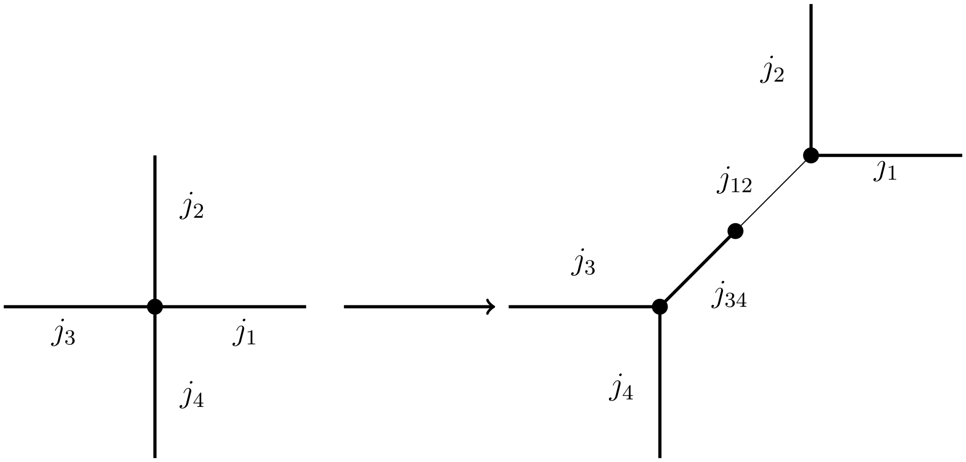


Figure 27: The link groupings in a four-linked site

$$|J,M\rangle = \sum_{\substack{m_1,m_2,m_3,\\ m_4,m_{12},m_{34}}} (-1)^{-j_{12}+j_{34}-j_1+j_2-j_3+j_4-m_{12}-m_{34}-M}\sqrt{2j_{12}+1}\sqrt{2j_{34}+1}\sqrt{2J+1}$$
$$\begin{pmatrix} j_1 & j_2 & j_{12} \\ m_1 & m_2 & -m_{12} \end{pmatrix} \begin{pmatrix} j_3 & j_4 & j_{34} \\ m_3 & m_4 & -m_{34} \end{pmatrix} \begin{pmatrix} j_{12} & j_{34} & J \\ m_{12} & m_{34} & -M \end{pmatrix}$$
$$|j_1,m_1\rangle|j_2,m_2\rangle|j_3,m_3\rangle|j_4,m_4\rangle \tag{7.2}$$

The next step is to ensure there is zero color at the sites, and then apply Gauss' Law

$$|0,0\rangle = \sum_{\substack{m_1,m_2,m_3,\\ m_4,m_{12}}} (-1)^{-j_1+j_2-j_3+j_4+j_{12}-m_{12}}\sqrt{2j_{12}+1}$$
$$\begin{pmatrix} j_1 & j_2 & j_{12} \\ m_1 & m_2 & -m_{12} \end{pmatrix} \begin{pmatrix} j_3 & j_4 & j_{12} \\ m_3 & m_4 & m_{12} \end{pmatrix} |j_1,m_1\rangle_1|j_2,m_2\rangle_2|j_3,m_3\rangle_3|j_4,m_4\rangle_4 \tag{7.3}$$

The tests for the validity of this solution are provided in appendix A.7 which confirm the solution accurately represents all types of sites in two-dimensions. The lattice is then constructed from the product of nine states; four states for the two-linked sites on the corners of the square, four states for the three-linked sites on the midpoints of the square, and one state for the four-linked site at the center. The object of our study will be a unit plaquette

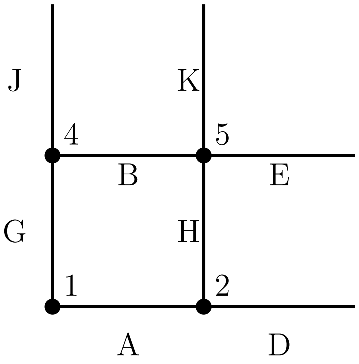


Figure 28: The cloverleaf lattice

on the cloverleaf lattice shown in figure 28. These are the states of the sites in the unit plaquette

$$|j_A, j_B, j_D, j_E, j_G, j_H, j_J, j_K\rangle =$$

$$\sum_{m_A,m_G,m_{AG}} \sum_{m_D,m_H,m'_A,m_{DH}} \sum_{m_B,m_J,m'_G,m_{BJ}} \sum_{m_E,m_K,m_B,m_H,m_{EK}}$$

$$|j_A, m_A, m'_A\rangle|j_H, m_H, m'_H\rangle|j_B, m_B, m'_B\rangle|j_G, m_G, m'_G\rangle|j_E, m_E\rangle|j_K, m_K\rangle|j_D, m_D\rangle|j_J, m_J\rangle$$

$$(-1)^{-j_A+j_G+j_{AG}-m_{AG}}\sqrt{2j_{AG}+1}\begin{pmatrix} j_A & j_G & j_{AG} \\ m_A & m_G & -m_{AG} \end{pmatrix}\begin{pmatrix} 0 & 0 & j_{AG} \\ 0 & 0 & m_{AG} \end{pmatrix}$$

$$(-1)^{-j_D+j_H-j_A+j_{DH}-m_{DH}}\sqrt{2j_{DH}+1}\begin{pmatrix} j_D & j_H & j_{DH} \\ m_D & m_H & -m_{DH} \end{pmatrix}\begin{pmatrix} j_A & 0 & j_{DH} \\ m'_A & 0 & m_{DH} \end{pmatrix}$$

$$(-1)^{-j_B+j_J+j_G+j_{BJ}-m_{BJ}}\sqrt{2j_{BJ}+1}\begin{pmatrix} j_B & j_J & j_{BJ} \\ m_B & m_J & -m_{BJ} \end{pmatrix}\begin{pmatrix} 0 & j_G & j_{BJ} \\ 0 & m'_G & m_{BJ} \end{pmatrix}$$

$$(-1)^{-j_E+j_K-j_B+j_H+j_{EK}-m_{EK}}\sqrt{2j_{EK}+1}\begin{pmatrix} j_E & j_K & j_{EK} \\ m_E & m_K & -m_{EK} \end{pmatrix}\begin{pmatrix} j_B & j_H & j_{EK} \\ m'_B & m'_H & m_{EK} \end{pmatrix}$$

The identities used can be found in appendix A.2. The unit plaquette state can be simplified by collapsing the 3j symbol with two zero columns to enforce $j_{AG} = m_{AG} = 0$. That makes it possible to rewrite the remaining 3j symbol for that line with $\delta_{j_A,j_G}\delta_{m_A,-m_G}$. The second and third lines can be simplified with $\delta_{j_A,j_{DH}}\delta_{m_A,m_{DH}}$ and $\delta_{j_{BJ},j_G}\delta_{m_{BJ},m_G}$ respectively.

$$|j_A, j_B, j_D, j_E, j_G, j_H, j_J, j_K\rangle = \sum_{m_A,m_G} \sum_{m_D,m_H,m'_A} \sum_{m_B,m_J,m'_G} \sum_{m_E,m_K,m_B,m_H,m_{EK}} \delta_{j_A,j_G}\delta_{m_A,-m_G}$$

$$\frac{(-1)^{j_A-m_A}}{\sqrt{2j_A+1}}(-1)^{-j_D+j_H-j_A}(-1)^{-j_B+j_J+j_G}(-1)^{-j_E+j_K-j_B+j_H+j_{EK}-m_{EK}}\sqrt{2j_{EK}+1}$$

$$\begin{pmatrix} j_D & j_H & j_A \\ m_D & m_H & m'_A \end{pmatrix}\begin{pmatrix} j_B & j_J & j_G \\ m_B & m_J & m'_G \end{pmatrix}\begin{pmatrix} j_E & j_K & j_{EK} \\ m_E & m_K & -m_{EK} \end{pmatrix}\begin{pmatrix} j_B & j_H & j_{EK} \\ m'_B & m'_H & m_{EK} \end{pmatrix}$$

$$|j_A, m_A, m'_A\rangle|j_H, m_H, m'_H\rangle|j_B, m_B, m'_B\rangle|j_G, m_G, m'_G\rangle|j_E, m_E\rangle|j_K, m_K\rangle|j_D, m_D\rangle|j_J, m_J\rangle$$

Applying the plaquette operator to the unit plaquette gives

$$\Box_1 |j_A, j_B, j_D, j_E, j_G, j_H, j_J, j_K\rangle =$$

$$\sum_{s_1,s_2,s_4,s_5} (-1)^{s_1+s_2+s_4+s_5} \sum_{J_A,M_A,M'_A} \sum_{J_H,M_H,M'_H} \sum_{J_B,M_B,M'_B} \sum_{J_G,M_G,M'_G}$$

$$\sum_{m_A,m_G} \sum_{m_D,m_H,m'_A} \sum_{m_B,m_J,m'_G} \sum_{m_E,m_K,m_B,m_H,m_{EK}} \delta_{j_A,j_G} \delta_{m_A,-m_G}$$

$$\frac{\sqrt{2j_{EK}+1}}{\sqrt{2j_A+1}} (-1)^{j_A-m_A-j_D+j_H-j_A-j_B+j_J+j_G-j_E+j_K-j_B+j_H+j_{EK}-m_{EK}}$$

$$\sqrt{(2j_A+1)(2J_A+1)}\sqrt{(2j_H+1)(2J_H+1)}\sqrt{(2j_B+1)(2J_B+1)}\sqrt{(2j_G+1)(2J_G+1)}$$

$$(-1)^{1-2j_A+M_A+M'_A}(-1)^{1-2j_H+M_H+M'_H}(-1)^{1-2j_B+M_B+M'_B}(-1)^{1-2j_G+M_G+M'_G}$$

$$\begin{pmatrix} j_D & j_H & j_A \\ m_D & m_H & m'_A \end{pmatrix} \begin{pmatrix} j_B & j_J & j_G \\ m_B & m_J & m'_G \end{pmatrix} \begin{pmatrix} j_E & j_K & j_{EK} \\ m_E & m_K & -m_{EK} \end{pmatrix} \begin{pmatrix} j_B & j_H & j_{EK} \\ m'_B & m'_H & m_{EK} \end{pmatrix}$$

$$\begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m_A & -s_1 & -M_A \end{pmatrix} \begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m'_A & s_2 & -M'_A \end{pmatrix} \begin{pmatrix} j_H & \frac{1}{2} & J_H \\ m_H & -s_2 & -M_H \end{pmatrix} \begin{pmatrix} j_H & \frac{1}{2} & J_H \\ m'_H & s_5 & -M'_H \end{pmatrix}$$

$$\begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m_B & s_4 & -M_B \end{pmatrix} \begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m'_B & -s_5 & -M'_B \end{pmatrix} \begin{pmatrix} j_G & \frac{1}{2} & J_G \\ m_G & s_1 & -M_G \end{pmatrix} \begin{pmatrix} j_G & \frac{1}{2} & J_G \\ m'_G & -s_4 & -M'_G \end{pmatrix}$$

$$|J_A, M_A, M'_A\rangle |J_H, M_H, M'_H\rangle |J_B, M_B, M'_B\rangle |J_G, M_G, M'_G\rangle |j_E, m_E\rangle |j_K, m_K\rangle |j_D, m_D\rangle |j_J, m_J\rangle$$

The next step is to project a final state to the result above, which is the initial state being acted upon by the plaquette operator. The following Kronecker deltas appear

$$\delta_{j_A,j'_A}\delta_{m_A,n_A}\delta_{m'_A,n'_A}\delta_{j_B,j'_B}\delta_{m_B,n_B}\delta_{m'_B,n'_B}\delta_{j_H,j'_H}\delta_{m_H,n_H}\delta_{m'_H,n'_H}\delta_{j_G,j'_G}\delta_{m_G,n_G}\delta_{m'_G,n'_G}$$

$$\delta_{j_E,j'_E}\delta_{m_E,n_E}\delta_{j_K,j'_K}\delta_{m_K,n_K}\delta_{j_D,j'_D}\delta_{m_D,n_D}\delta_{j_J,j'_J}\delta_{m_J,n_J}$$

This results in the following

$$\langle j'_A, j'_B, j'_D, j'_E, j'_G, j'_H, j'_J, j'_K | \Box_1 | j_A, j_B, j_D, j_E, j_G, j_H, j_J, j_K \rangle =$$

$$\sum_{s_1,s_2,s_4,s_5} (-1)^{s_1+s_2+s_4+s_5} \sum_{J_A,M_A,M'_A} \sum_{J_H,M_H,M'_H} \sum_{J_B,M_B,M'_B} \sum_{J_G,M_G,M'_G}$$

$$\sum_{m_A,m_G} \sum_{m_D,m_H,m'_A} \sum_{m_B,m_J,m'_G} \sum_{m_E,m_K,m_B,m_H,m_{EK}} \sum_{n_{EK}} \delta_{j_A,j_G} \delta_{m_A,-m_G}$$

$$\frac{\sqrt{2j_{EK}+1}\sqrt{2j'_{EK}+1}}{2j_A+1} (-1)^{2(j_A-m_A-j_D+j_H-j_A-j_B+j_J+j_G-j_E+j_K-j_B+j_H)+j_{EK}-m_{EK}+j'_{EK}-n_{EK}}$$

$$\sqrt{(2j_A+1)(2J_A+1)}\sqrt{(2j_H+1)(2J_H+1)}\sqrt{(2j_B+1)(2J_B+1)}\sqrt{(2j_G+1)(2J_G+1)}$$

$$(-1)^{1-2j_A+M_A+M'_A}(-1)^{1-2j_H+M_H+M'_H}(-1)^{1-2j_B+M_B+M'_B}(-1)^{1-2j_G+M_G+M'_G}$$

$$\begin{pmatrix} j_D & J_H & J_A \\ m_D & M_H & M'_A \end{pmatrix} \begin{pmatrix} J_B & j_J & J_G \\ M_B & m_J & M'_G \end{pmatrix} \begin{pmatrix} j_E & j_K & j_{EK} \\ m_E & m_K & -m_{EK} \end{pmatrix} \begin{pmatrix} J_B & J_H & j_{EK} \\ M'_B & M'_H & m_{EK} \end{pmatrix}$$

$$\begin{pmatrix} j_D & j_H & j_A \\ m_D & m_H & m'_A \end{pmatrix} \begin{pmatrix} j_B & j_J & j_G \\ m_B & m_J & m'_G \end{pmatrix} \begin{pmatrix} j_E & j_K & j'_{EK} \\ m_E & m_K & -n_{EK} \end{pmatrix} \begin{pmatrix} j_B & j_H & j'_{EK} \\ m'_B & m'_H & n_{EK} \end{pmatrix}$$

$$\begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m_A & -s_1 & -M_A \end{pmatrix} \begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m'_A & s_2 & -M'_A \end{pmatrix} \begin{pmatrix} j_H & \frac{1}{2} & J_H \\ m_H & -s_2 & -M_H \end{pmatrix} \begin{pmatrix} j_H & \frac{1}{2} & J_H \\ m'_H & s_5 & -M'_H \end{pmatrix}$$

$$\begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m_B & s_4 & -M_B \end{pmatrix} \begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m'_B & -s_5 & -M'_B \end{pmatrix} \begin{pmatrix} j_G & \frac{1}{2} & J_G \\ m_G & s_1 & -M_G \end{pmatrix} \begin{pmatrix} j_G & \frac{1}{2} & J_G \\ m'_G & -s_4 & -M'_G \end{pmatrix}$$

The next step is to attempt to introduce 6j symbols, which are explained in appendix A.2. There is a way to represent 3j and 6j symbols [39], which can give insight into which direction to go. Using [39], it is clear that nine of the 3j symbols can be manipulated to forming 6j symbols, as in the plaquette operator expression for a one-dimensional lattice in appendix A.3. The difference here is that the four-linked site introduced j_{EK} from the initial state and j'_{EK} from the final state. Now, j_{EK} is the angular momentum resulting from the addition of j_E and j_K. Both these links are outside the plaquette of interest, and thus the assumption is made that $j_{EK} = j'_{EK}$ since it should not change under the plaquette operator. We will further assume that $m_{EK} = n_{EK}$. This way, we can follow the same steps in appendix A.3, we find that the plaquette operator on the unit plaquette on the cloverleaf lattice is

$$\begin{gathered}\langle \text{final}|\Box_1|\text{initial}\rangle = \\ \sqrt{(2j_A+1)(2J_A+1)}\sqrt{(2j_B+1)(2J_B+1)}\sqrt{(2j_G+1)(2J_G+1)}\sqrt{(2j_H+1)(2J_H+1)} \\ (-1)^{j_D+j_J+j_{EK}+2(J_A+J_B-j_G-j_H)} \\ \begin{Bmatrix} 0 & j_G & j_A \\ \frac{1}{2} & J_A & J_G \end{Bmatrix} \begin{Bmatrix} j_D & j_H & j_A \\ \frac{1}{2} & J_A & J_H \end{Bmatrix} \begin{Bmatrix} j_J & j_G & j_B \\ \frac{1}{2} & J_B & J_G \end{Bmatrix} \begin{Bmatrix} j_{EK} & j_H & j_B \\ \frac{1}{2} & J_B & J_H \end{Bmatrix}\end{gathered} \tag{7.4}$$

This is similar to the result for the one-dimensional lattice. Thus, the initial and final states are defined by the links that touch the sites on the plaquette, with the links on the plaquette being necessarily different in the final state. The intuition gained from working with one-dimensional lattices indicates that there would be six distinct types of states: a vacuum state, a single excited state, an edge-adjacent double excited state, a corner-adjacent double excited state, a triple excited state, and an all-excited state.

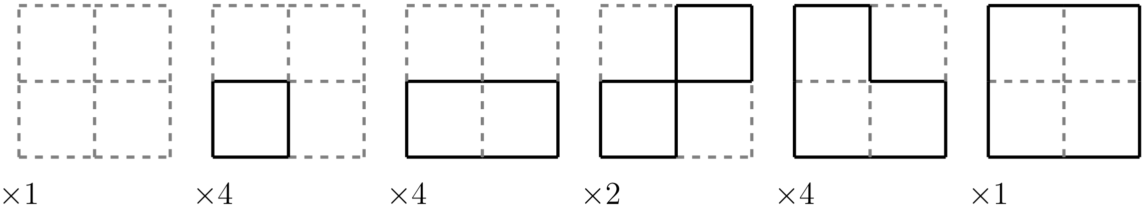


Figure 29: The types of possible states on a cloverleaf lattice with $j_{\max} = 1$. The numbers at the bottom of the diagrams indicate the number of states symmetric under rotation of the lattice.

However, this is not the end. The existence of j_{EK} as a variable in the plaquette operator means that there is hidden physics that is a result of the four-linked site. In other words, what states the individual links are in is not sufficient to describe the lattice, and additional information from the addition of angular momenta needs to be specified. Thus, in the future, the goal is to carefully study and understand the full extent of the four-linked site and its physical implications.

8 Conclusion

In this thesis, we provided three contributions to the field of lattice gauge theories on quantum computers.

In section 2, the topic of Abelian and non-Abelian gauge theories was introduced, after which space-time was discretized, and so was the gauge theory. Lastly, the Kogut-Susskind Hamiltonian formulation of lattice gauge theories was described.

Section 3, we use the angular momentum basis to write down a matrix form of the Hamiltonian. The truncation is discussed, along with calculations for the chromoelectric and chromomagnetic parts of the Hamiltonian.

Section 4 introduces two different types of quantum computers. Quantum annealers from DWave were used to perform VQE calculations, vacuum expectation value calculations, and restricted time evolution calculations. Then, gate-based quantum computers from IBM were used to study ground-state calculations as well as time evolution calculations. The conclusion is that time evolution calculations require extensive error correction for long-range results.

Section 5 discusses our publication that introduced a new method of error mitigation specifically tailored to time evolution calculations. Self mitigation used the time evolution circuit and evolved it half a step forward and then half a step backward. This approximates the error accumulated during a physical time evolution run. This was used to study two-plaquette and five-plaquette lattices. Propagation of excitation along the lattice was shown as well at long timescales.

In section 6, a Pauli decomposition algorithm was developed that has the unique feature of exploiting visual symmetries that can make it easy to decompose any $2^N \times 2^N$ matrix visually. This method was adopted into a python code.

In section 7, future prospects for study are discussed. The theoretical foundations were developed for weaker coupling Hamiltonians via a truncation of $j_{\max} = 1$. Another extension is upgrading to two-dimensional lattices. The four-linked site is developed, from which the cloverleaf lattice state is constructed. Then, the unit plaquette operator on the cloverleaf is derived, and future paths of study are discussed for this topic.

A Appendix

A.1 The Wilson Line

The product $\bar{\psi}(y)\psi(x)$ is not gauge invariant. We wish to introduce a gauge covariant field that transforms in such a way that can produce a gauge invariant bilinear product $\bar{\psi}(y)W(y,x)\psi(x)$. In the absence of any gauge fields on the continuum, the translation operator $\hat{T}(a)$ acts on a field $\psi(x)$ by displacing the field by a quantity a along a straight line

$$\hat{T}(a)\psi(x) = e^{-a^\mu \partial_\mu}\psi(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \left(-a^\mu \partial_\mu\right)^n \psi(x) = \psi(x-a) \tag{A.1}$$

In the presence of gauge fields, the covariant derivative can be used to redefine the translation operator $exp(a^\mu D_\mu)$. Thus, we consider infinitesimal translations $\hat{T}(dx^\mu) = 1 - dx^\mu D_\mu$. This modified translation contains two parts. The action of successive infinitesimal translations will translate to a single translation by the integral of the exponent

$$\prod (1 - dx^\mu D_\mu) = \exp\left(\int_\Gamma dx^\mu D_\mu\right) \tag{A.2}$$

where Λ is a curved path connecting two points. The first part is described by the regular derivative, which describes a change in the coordinates and can thus be neglected. The second part describes the resulting changes due to moving through the gauge field. This part produces the angles needed to achieve the bilinear transformation needed for gauge invariance

$$W(y,x) = exp\left(ig\int_x^y dz^\mu A_\mu(z)\right) \tag{A.3}$$

$$W(y,x) \to e^{[ig\int_x^y dz^\mu A_\mu(z)] + i\alpha(y) - i\alpha(x)} = e^{i\alpha(y)} W(y,x) e^{-i\alpha(x)} \tag{A.4}$$

Thus, we have finally arrived at a gauge covariant field that parallel transports matter fields across the gauge potential. This is the Wilson line. For non-Abelian gauge transformations, the path ordering operator is needed. A proof of the applicability of this derivation to non-Abelian transformations can be found in [40].

A.2 The Wigner 3j and 6j Symbols

This appendix is based on [**Math˙Handbook**]. The 3j symbols are defined as

$$\langle j_1, m_1; j_2, m_2 | j_3, -m_3 \rangle = (-1)^{j_1-j_2-m_3}\sqrt{2j_3+1}\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \tag{A.5}$$

The angular momenta must follow the triangle conditions, and the projections must add to zero

$$|j_1 - j_2| \le j_3 \le j_1 + j_2 \tag{A.6}$$

$$m_1 + m_2 + m_3 = 0 \tag{A.7}$$

The 3j columns of the 3j symbols are allowed to commute cyclically. A negative sign is gained during anti-cyclic permutation or sign inversion of the angular momentum projections

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = (-1)^{j+j_1+j_2}\begin{pmatrix} j_2 & j_1 & j_3 \\ m_2 & m_1 & j_3 \end{pmatrix} = (-1)^{j+j_1+j_2}\begin{pmatrix} j_1 & j_2 & j_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix} \tag{A.8}$$

If one column is composed of zeros, the result is

$$\begin{pmatrix} j_1 & j_2 & 0 \\ m_1 & m_2 & 0 \end{pmatrix} = \delta_{j_1 j_2}\delta_{m_1\ -m_2}\frac{(-1)^{j_1-m_1}}{\sqrt{2j_1+1}} \tag{A.9}$$

The orthogonality relations are

$$(2j_3+1)\sum_{m_1 m_2}\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}\begin{pmatrix} j_1 & j_2 & j'_3 \\ m_1 & m_2 & m'_3 \end{pmatrix} = \delta_{j_3,j'_3}\delta_{m_3,m'_3} \tag{A.10}$$

$$\sum_{j_3 m_3}(2j_3+1)\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}\begin{pmatrix} j_1 & j_2 & j_3 \\ m'_1 & m'_2 & m_3 \end{pmatrix} = \delta_{m_1,m'_1}\delta_{m_2,m'_2} \tag{A.11}$$

$$\sum_{m_1 m_2 m_3}\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = 1 \tag{A.12}$$

The 6j symbols are defined as

$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ l_1 & l_2 & l_3 \end{Bmatrix} = \sum_{m_1}\sum_{m_2}\sum_{m_3}\sum_{m'_1}\sum_{m'_2}\sum_{m'_3}(-1)^{(l_1+m'_1)+(l_2+m'_2)+(l_3+m'_3)}$$
$$\times \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}\begin{pmatrix} j_1 & l_2 & l_3 \\ m_1 & m'_2 & -m'_3 \end{pmatrix}\begin{pmatrix} l_1 & j_2 & l_3 \\ -m'_1 & m_2 & m'_3 \end{pmatrix}\begin{pmatrix} l_1 & l_2 & j_3 \\ m'_1 & -m'_2 & m_3 \end{pmatrix} \tag{A.13}$$

The 6j symbol allows for the permutation of any two columns or the permutation of the upper and lower numbers of two of the columns

$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ l_1 & l_2 & l_3 \end{Bmatrix} = \begin{Bmatrix} j_2 & j_1 & j_3 \\ l_2 & l_1 & l_3 \end{Bmatrix} = \begin{Bmatrix} j_1 & l_2 & l_3 \\ j_1 & j_2 & l_3 \end{Bmatrix} \tag{A.14}$$

The orthogonality relation is

$$\sum_{j_3}(2j_3+1)(2l_3+1)\begin{Bmatrix} j_1 & j_2 & j_3 \\ l_1 & l_2 & l_3 \end{Bmatrix}\begin{Bmatrix} j_1 & j_2 & j_3 \\ l_1 & l_2 & l'_3 \end{Bmatrix} = \delta_{l_3,l'_3} \tag{A.15}$$

A.3 Deriving the Plaquette Operator

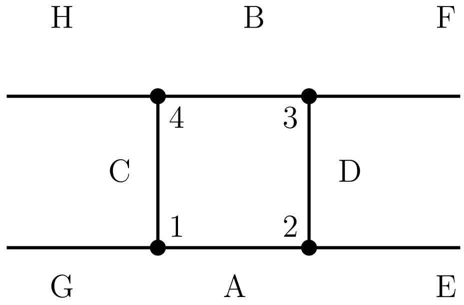


Figure 30: The general unit plaquette on a one dimensional lattice

$$\hat{\Box}(p) = \sum_{MNPQ} U_{MN}(n,\mu)U_{NP}(n+\mu,\nu)(U^{\dagger})_{PQ}(n+\nu,\mu)(U^{\dagger})_{QR}(n,\nu) \tag{A.16}$$

where $M, N, P, Q \in \{-1/2, +1/2\}_{\Delta=1}$. To express this in terms of the link operator, we will first need to define the hermitian conjugate of the link operator [41]

$$(U^{\dagger})_{mn} = (U_{nm})^{\dagger} \tag{A.17}$$

$$(U_{mn})^{\dagger} = (-1)^{m-n}U_{-m-n} \tag{A.18}$$

The second equation is the only one that is relevant for our conventions. The first equation is only useful if the link states are pointing in the direction of the link. For our purposes, it was stated that the links states are only pointing in the positive directions. The state of this lattice is:

$$|\Lambda\rangle = |j_A, j_B, j_C, j_D, j_E, j_F, j_G, j_H\rangle =$$

$$\sum_{m_A,m'_A}\sum_{m_B,m'_B}\sum_{m_C,m'_C}\sum_{m_D,m'_D}\sum_{m_E}\sum_{m_F}\sum_{m'_G}\sum_{m'_H}$$

$$\begin{pmatrix} j_A & j_G & j_{CS} \\ m_A & m'_G & m_{CS} \end{pmatrix}\begin{pmatrix} j_A & j_E & j_D \\ m'_A & m_E & m_D \end{pmatrix}\begin{pmatrix} j_B & j_F & j_D \\ m'_B & m_F & m'_D \end{pmatrix}\begin{pmatrix} j_B & j_H & j_C \\ m_B & m'_H & m'_C \end{pmatrix}$$

$$|j_A, m_A, m'_A\rangle|j_B, m_B, m'_B\rangle|j_C, m_C, m'_C\rangle|j_D, m_D, m'_D\rangle|j_E, m_E\rangle|j_F, m_F\rangle|j_G, m'_G\rangle|j_H, m'_H\rangle$$

The plaquette operator that includes sites 1, 2, 3, and 4 is:

$$Z_1 = \sum_{s_1,s_2,s_3,s_4}(-1)^{s_1+s_2+s_3+s_4}U^A_{-s_1,s_2}U^J_{-s_2,s_3}U^B_{s_4,-s_3}U^I_{s_1,-s_4}$$

$$U_{s,s'}|j,m,m'\rangle = \sum_J\sum_{M,M'}\sqrt{(2j+1)(2J+1)}(-1)^{1-2j+M+M'}$$

$$\begin{pmatrix} j & \frac{1}{2} & J \\ m & s & -M \end{pmatrix}\begin{pmatrix} j & \frac{1}{2} & J \\ m' & s' & -M' \end{pmatrix}|J, M, M'\rangle$$

The plaquette operator is applied to the lattice to get:

$$Z_1|\Lambda\rangle =$$

$$\sum_{m_A,m'_A}\sum_{m_B,m'_B}\sum_{m_C,m'_C}\sum_{m_D,m'_D}\sum_{m_E}\sum_{m_F}\sum_{m'_G}\sum_{m'_H}\sum_{J_A,J_B,J_C,J_D}\sum_{s_1,s_2,s_3,s_4}\sum_{M_A,M'_A}\sum_{M_B,M'_B}\sum_{M_C,M'_C}\sum_{M_D,M'_D}$$

$$(-1)^{s_1+s_2+s_3+s_4-2(j_A+j_B+j_C+j_D)+M_A+M'_A+M_B+M'_B+M_C+M'_C+M_D+M'_D}$$

$$\sqrt{(2j_A+1)(2J_A+1)}\sqrt{(2j_B+1)(2J_B+1)}\sqrt{(2j_C+1)(2J_C+1)}\sqrt{(2j_D+1)(2J_D+1)}$$

$$\begin{pmatrix} j_A & j_G & j_C \\ m_A & m'_G & m_C \end{pmatrix}\begin{pmatrix} j_A & j_E & j_D \\ m'_A & m_E & m_D \end{pmatrix}\begin{pmatrix} j_B & j_F & j_D \\ m'_B & m_F & m'_D \end{pmatrix}\begin{pmatrix} j_B & j_H & j_C \\ m_B & m'_H & m'_C \end{pmatrix}$$

$$\begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m_A & -s_1 & -M_A \end{pmatrix}\begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m'_A & s_2 & -M'_A \end{pmatrix}\begin{pmatrix} j_D & \frac{1}{2} & J_D \\ m_D & -s_2 & -M_D \end{pmatrix}\begin{pmatrix} j_D & \frac{1}{2} & J_D \\ m'_D & s_3 & -M'_D \end{pmatrix}$$

$$\begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m_B & s_4 & -M_B \end{pmatrix}\begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m'_B & -s_3 & -M'_B \end{pmatrix}\begin{pmatrix} j_C & \frac{1}{2} & J_C \\ m_C & s_1 & -M_C \end{pmatrix}\begin{pmatrix} j_C & \frac{1}{2} & J_C \\ m'_C & -s_4 & -M'_C \end{pmatrix}$$

$$|J_A, M_A, M'_A\rangle|J_B, M_B, M'_B\rangle|J_C, M_C, M'_C\rangle|J_D, M_D, M'_D\rangle$$

$$|j_E, m_E\rangle|j_F, m_F\rangle|j_G, m'_G\rangle|j_H, m'_H\rangle$$

We will project the lattice state onto the plaquette operator and ket product. The states are orthogonal, so $\langle j_1, m_1, m'_1|j_2, m_2, m'_2\rangle = \delta_{j_1,j_2}\delta_{m_1,m_2}\delta_{m'_1,m'_2}$.

$$\langle\Lambda|Z_1|\Lambda\rangle = \sum_{J_A,J_B,J_I,J_J}$$

$$\sqrt{(2j_A+1)(2J_A+1)}\sqrt{(2j_B+1)(2J_B+1)}\sqrt{(2j_C+1)(2J_C+1)}\sqrt{(2j_D+1)(2J_D+1)}$$

$$\sum_{s_1,s_2,s_3,s_4}\sum_{M_A,M'_A}\sum_{M_B,M'_B}\sum_{M_C,M'_C}\sum_{M_D,M'_D}$$

$$\sum_{m_A,m'_A}\sum_{m_B,m'_B}\sum_{m_C,m'_C}\sum_{m_D,m'_D}\sum_{m_E}\sum_{m_F}\sum_{m'_G}\sum_{m'_H}$$

$$(-1)^{s_1+s_2+s_3+s_4-2(j_A+j_B+j_C+j_D)+M_A+M'_A+M_B+M'_B+M_C+M'_C+M_D+M'_D}$$

$$\begin{pmatrix} j_A & j_G & j_C \\ m_A & m'_G & m_C \end{pmatrix}\begin{pmatrix} j_A & j_E & j_D \\ m'_A & m_E & m_D \end{pmatrix}\begin{pmatrix} J_A & j_G & J_C \\ M_A & m'_G & M_C \end{pmatrix}\begin{pmatrix} J_A & j_E & J_D \\ M'_A & m_E & M_D \end{pmatrix}$$

$$\begin{pmatrix} j_B & j_F & j_D \\ m'_B & m_F & m'_D \end{pmatrix}\begin{pmatrix} j_B & j_H & j_C \\ m_B & m'_H & m'_C \end{pmatrix}\begin{pmatrix} J_B & j_F & J_D \\ M'_B & m_F & M'_D \end{pmatrix}\begin{pmatrix} J_B & j_H & J_C \\ M_B & m'_H & M'_C \end{pmatrix}$$

$$\begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m_A & -s_1 & -M_A \end{pmatrix}\begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m'_A & s_2 & -M'_A \end{pmatrix}\begin{pmatrix} j_D & \frac{1}{2} & J_D \\ m_D & -s_2 & -M_D \end{pmatrix}\begin{pmatrix} j_D & \frac{1}{2} & J_D \\ m'_D & s_3 & -M'_D \end{pmatrix}$$

$$\begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m_B & s_4 & -M_B \end{pmatrix}\begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m'_B & -s_3 & -M'_B \end{pmatrix}\begin{pmatrix} j_C & \frac{1}{2} & J_C \\ m_C & s_1 & -M_C \end{pmatrix}\begin{pmatrix} j_C & \frac{1}{2} & J_C \\ m'_C & -s_8 & -M'_C \end{pmatrix}$$

We will first get rid of the squared 3j symbols using the following identities:

$$\sum_{m_1,m_2}\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}\begin{pmatrix} j_1 & j_2 & j'_3 \\ m_1 & m_2 & m'_3 \end{pmatrix} = \frac{\delta_{j_3,j'_3}\delta_{m_3,m'_3}}{2j_3+1}$$

$$\sum_{m_3} 1 = 2j_3+1$$

This gives us the following:

$$\langle \Lambda | Z_1 | \Lambda \rangle = \sum_{J_A, J_B, J_I, J_J}$$

$$\sqrt{(2j_A+1)(2J_A+1)}\sqrt{(2j_B+1)(2J_B+1)}\sqrt{(2j_C+1)(2J_C+1)}\sqrt{(2j_D+1)(2J_D+1)}$$

$$\sum_{s_1,s_2,s_3,s_4} \sum_{M_A,M'_A} \sum_{M_B,M'_B} \sum_{M_C,M'_C} \sum_{M_D,M'_D} \sum_{m_A,m'_A} \sum_{m_B,m'_B} \sum_{m_E m_F,m'_G,m'_H} \sum_{m_C,m'_C} \sum_{m_D,m'_D}$$

$$(-1)^{s_1+s_2+s_3+s_4-2(j_A+j_B+j_C+j_D)+M_A+M'_A+M_B+M'_B+M_C+M'_C+M_D+M'_D}$$

$$\begin{pmatrix} j_C & \frac{1}{2} & J_C \\ m_C & s_1 & -M_C \end{pmatrix} \begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m_A & -s_1 & -M_A \end{pmatrix} \begin{pmatrix} j_A & j_G & j_C \\ m_A & m'_G & m_C \end{pmatrix} \begin{pmatrix} J_A & j_G & J_C \\ M_A & m'_G & M_C \end{pmatrix}$$

$$\begin{pmatrix} j_A & \frac{1}{2} & J_A \\ m'_A & s_2 & -M'_A \end{pmatrix} \begin{pmatrix} j_D & \frac{1}{2} & J_D \\ m_D & -s_2 & -M_D \end{pmatrix} \begin{pmatrix} j_A & j_E & j_D \\ m'_A & m_E & m_D \end{pmatrix} \begin{pmatrix} J_A & j_E & J_D \\ M'_A & m_E & M_D \end{pmatrix}$$

$$\begin{pmatrix} j_D & \frac{1}{2} & J_D \\ m'_D & s_3 & -M'_D \end{pmatrix} \begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m'_B & -s_3 & -M'_B \end{pmatrix} \begin{pmatrix} j_B & j_F & j_D \\ m'_B & m_F & m'_D \end{pmatrix} \begin{pmatrix} J_B & j_F & J_D \\ M'_B & m_F & M'_D \end{pmatrix}$$

$$\begin{pmatrix} j_B & \frac{1}{2} & J_B \\ m_B & s_4 & -M_B \end{pmatrix} \begin{pmatrix} j_C & \frac{1}{2} & J_C \\ m'_C & -s_4 & -M'_C \end{pmatrix} \begin{pmatrix} j_B & j_H & j_C \\ m_B & m'_H & m'_C \end{pmatrix} \begin{pmatrix} J_B & j_H & J_C \\ M_B & m'_H & M'_C \end{pmatrix}$$

Each line of 3j symbols will be grouped into a 6j symbol. A lot of work went into manipulating these in order to get

$$\langle \Lambda' | Z_1 | \Lambda \rangle =$$

$$\sqrt{(2j_A+1)(2J_A+1)}\sqrt{(2j_B+1)(2J_B+1)}\sqrt{(2j_C+1)(2J_C+1)}\sqrt{(2j_D+1)(2J_D+1)}$$

$$(-1)^{j_E+j_F+j_G+j_H+2(J_A+J_B-j_C-j_D)}$$

$$\begin{Bmatrix} j_G & j_C & j_A \\ \frac{1}{2} & J_A & J_C \end{Bmatrix} \begin{Bmatrix} j_E & j_D & j_A \\ \frac{1}{2} & J_A & J_D \end{Bmatrix} \begin{Bmatrix} j_F & j_D & j_B \\ \frac{1}{2} & J_B & J_D \end{Bmatrix} \begin{Bmatrix} j_H & j_C & j_B \\ \frac{1}{2} & J_B & J_C \end{Bmatrix}$$

A.4 Randomized CNOT Gates

The sixteen options for randomized compiling are

$$CX_{jk} \tag{A.19}$$

$$X_kCX_{jk}X_k \tag{A.20}$$

$$Y_kCX_{jk}Z_jY_k \tag{A.21}$$

$$Z_kCX_{jk}Z_jZ_k \tag{A.22}$$

$$X_jCX_{jk}X_jX_k \tag{A.23}$$

$$X_jX_kCX_{jk}X_j \tag{A.24}$$

$$X_jY_kCX_{jk}Y_jZ_k \tag{A.25}$$

$$X_jZ_kCX_{jk}Y_jY_k \tag{A.26}$$

$$Y_jCX_{jk}Y_jX_k \tag{A.27}$$

$$Y_jX_kCX_{jk}Y_j \tag{A.28}$$

$$Y_jY_kCX_{jk}X_jZ_k \tag{A.29}$$

$$Y_jZ_kCX_{jk}X_jY_l \tag{A.30}$$

$$Z_jCX_{jk}Z_j \tag{A.31}$$

$$Z_jX_kCX_{jk}Z_jX_k \tag{A.32}$$

$$Z_jY_kCX_{jk}Y_k \tag{A.33}$$

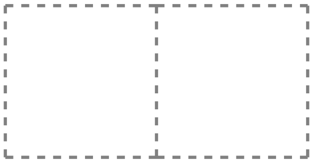
$$Z_jZ_kCX_{jk}Z_k \tag{A.34}$$

A.5 Pauli Decomposition Code

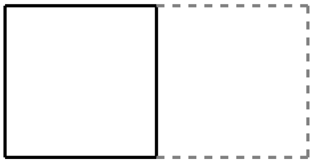
```python
import numpy as np
import sympy

n = 3 # number of qubits
N = int(2**n) # number of elements in the matrix

sI,sX,sY,sZ=sympy.symbols("I,X,Y,Z", commutative=False)
j = complex(0,1)
I = np.array([[1,0],[0,1]]) # Identity gate
X = np.array([[0,1],[1,0]]) # Pauli X gate
Y = np.array([[0,-j],[j,0]]) # Pauli Y gate
Z = np.array([[1,0],[0,-1]]) # Pauli Z gate
H = (Z+X)/np.sqrt(2) # Hadamard gate
Diagonal = np.array([I,Z])
AntiDiagonal = np.array([X,Y])

V1 = np.array([I,X]) # identity and flip gates for 1 qubit
V = V1 # we start with 1 qubit, then we do n-1 tensor products
for i in range(n-1):
    V = np.kron(V,V1)



def Hadamard_kron(n):
    """gives the n kronecker products of the Hadamard gate, as if to apply the Hadamard gate to each qubit"""
    Hn = H
    for i in range(n-1):
        Hn = np.kron(Hn,H)
    return Hn


```

```python
def binary_numbers(n):
    """creates a list of 2**n binary numbers"""
    N=2**n
    binary_list = []
    for i in range(N):
        b = bin(i)[2:]
        l = len(b)
        b = str(0) * (n - l) + b
        binary_list.append(b)

    return binary_list


def symmetry_group_symbols(k):
    """ the way it works:
    if k = ’0’ : returns the array of diagonal gates respectively
    if k = ’1’ : returns the array of anti-diagonal gates respectively
    all symbolically"""

    if k == ’0’:
        return np.array([sI,sZ])
    elif k == ’1’:
        return np.array([sX,sympy.I * sY])
    else:
        raise Exception(’insert string 0 or string 1’)


def symmetry_gates_symbols(bin_str):
    # takes a binary string like ’010’ and creates the set of matrices corresponding to all the gates for
    # the symmetry defined by ’0’ for the diagonal group, and ’1’ for the anti-diagonal group
    # returns an array containing the aforementioned matrices
    group = symmetry_group_symbols(bin_str[0])
    for i in bin_str[1:]:
        group = np.kron(group,symmetry_group_symbols(i))
    return group

## apply element wise multipication of the Hamiltonian by the products of identity and flip
## then multiply the result by the corresponding product to diagonalize the matrix
## then extract the diagonal elements from those diagonal matrices
## this results in the ordered list of elements for each symmetry

def elements_dictionary(Hamiltonian):
    # takes in a Hamiltonian matrix that must be 2**n x 2**n
    # returns 2**n dictionary of values extracted from the matrix, ordered in a certain way
    # that ordering is needed to correctly combine with the hadamard gate
    elements_dict = {}
    bin_nums = binary_numbers(n)

    for i in range(N):
        elements_dict[bin_nums[i]] = np.diag( np.matmul(Hamiltonian,V[i]))

    return elements_dict


### The final result:

def pauli_decomposition(Hamiltonian):
    # takes 2**n x 2**n hamiltonian matrix
    # returns a list of arrays containing the weighted sums of the gates
    result = []
    bin_nums = binary_numbers(n)
    elements_dict = elements_dictionary(Hamiltonian)
    Hn = Hadamard_kron(n)
    for i in bin_nums:
        A = symmetry_gates_symbols(i)
        B = elements_dict[i].reshape((N,1))
        C = 2**(-n/2)*np.round(np.matmul(Hn,B),decimals=5)
        D = np.matmul(A,C)
        result.append(D)
    return sympy.simplify(np.array(result))
# using the pauli_decomposition function, one can input a 2**n x 2**n Hamiltonian
# the output will be the Hamiltonian in terms of pauli gates products
# input Hamiltonian MUST BE NUMPY ARRAY
```

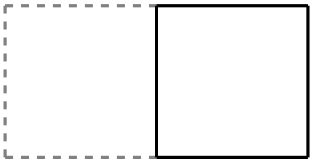
A.6 Two Plaquette Lattice states with $j_{\max} = 1$



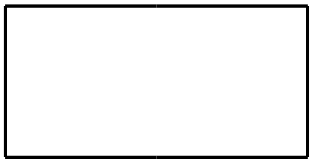
$|111\rangle$



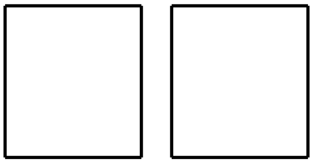
$|221\rangle$



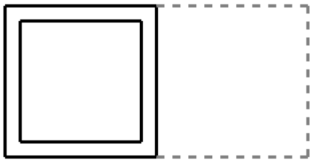
$|122\rangle$



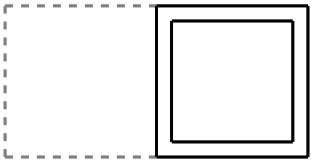
$|212\rangle$



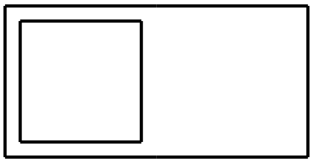
$|232\rangle$



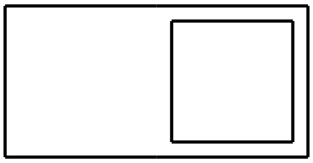
$|331\rangle$



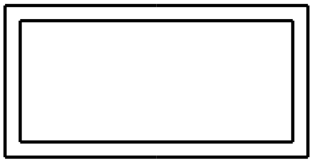
$|133\rangle$



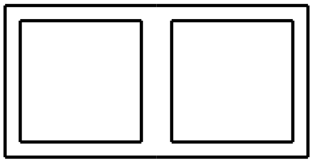
$|322\rangle$



$|223\rangle$



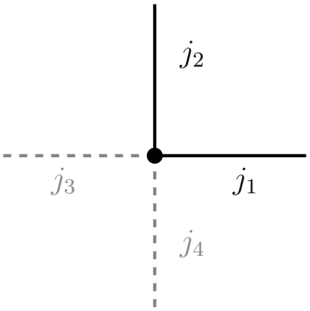
$|313\rangle$



$|333\rangle$

A.7 Four-linked Site Tests

The cloverleaf lattice contains a site with four links connected to it. This section contains calculations



$$|0,0\rangle = \sum_{m_1,m_2,m_{12}} (-1)^{-j_1+j_2+j_{12}-m_{12}} \sqrt{2j_{12}+1} |j_1,m_1\rangle_1 |j_2,m_2\rangle_2 |0,0\rangle_3 |0,0\rangle_4$$

$$\begin{pmatrix} j_1 & j_2 & j_{12} \\ m_1 & m_2 & -m_{12} \end{pmatrix} \begin{pmatrix} 0 & 0 & j_{12} \\ 0 & 0 & m_{12} \end{pmatrix}$$

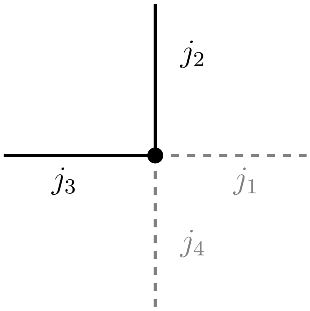
$$= \sum_{m_1,m_2} (-1)^{-j_1+j_2} |j_1,m_1\rangle_1 |j_2,m_2\rangle_2 |0,0\rangle_3 |0,0\rangle_4$$

$$\begin{pmatrix} j_1 & j_2 & 0 \\ m_1 & m_2 & 0 \end{pmatrix} = \sum_{m_1,m_2} (-1)^{-j_1+j_2} |j_1,m_1\rangle_1 |j_2,m_2\rangle_2 |0,0\rangle_3 |0,0\rangle_4 \begin{pmatrix} j_1 & j_2 & 0 \\ m_1 & m_2 & 0 \end{pmatrix}$$

$$= \sum_{m_1} |j_1,m_1\rangle_1 |j_1,-m_1\rangle_2 |0,0\rangle_3 |0,0\rangle_4 \begin{pmatrix} j_1 & j_1 & 0 \\ m_1 & -m_1 & 0 \end{pmatrix}$$

$$= \sum_{m_1} \frac{(-1)^{j_1-m_1}}{\sqrt{2j_1+1}} |j_1,m_1\rangle_1 |j_1,-m_1\rangle_2 |0,0\rangle_3 |0,0\rangle_4$$

This matches our 2-link state solution.



$$|0,0\rangle = \sum_{m_2,m_3,m_{12}} (-1)^{j_2-j_3+j_{12}-m_{12}} \sqrt{2j_{12}+1} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_3,m_3\rangle_3 |0,0\rangle_4$$
$$\begin{pmatrix} 0 & j_2 & j_{12} \\ 0 & m_2 & -m_{12} \end{pmatrix} \begin{pmatrix} j_3 & 0 & j_{12} \\ m_3 & 0 & m_{12} \end{pmatrix}$$
$$= \sum_{m_2,m_3,m_{12}} (-1)^{+j_2-j_3+j_{12}-m_{12}} \sqrt{2j_{12}+1} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_3,m_3\rangle_3 |0,0\rangle_4$$
$$\begin{pmatrix} j_2 & j_{12} & 0 \\ m_2 & -m_{12} & 0 \end{pmatrix} \begin{pmatrix} j_{12} & j_3 & 0 \\ m_{12} & m_3 & 0 \end{pmatrix}$$

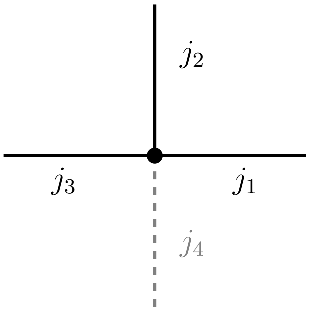
There are two ways to go from here. Either we choose $j_{12} = j_2$ or $j_{12} = j_3$. I will first explore the j_2 option:

$$|0,0\rangle = \sum_{m_2} (-1)^{+j_2-m_2} \sqrt{2j_2+1} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_2,-m_2\rangle_3 |0,0\rangle_4$$
$$\begin{pmatrix} j_2 & j_2 & 0 \\ m_2 & -m_2 & 0 \end{pmatrix} \begin{pmatrix} j_2 & j_2 & 0 \\ m_2 & -m_2 & 0 \end{pmatrix}$$
$$= \sum_{m_2} (-1)^{+j_2-m_2} \sqrt{2j_2+1} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_2,-m_2\rangle_3 |0,0\rangle_4 \frac{(-1)^{2j_2-2m_2}}{2j_2+1}$$
$$= \sum_{m_2} \frac{(-1)^{j_2-3m_2}}{\sqrt{2j_2+1}} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_2,-m_2\rangle_3 |0,0\rangle_4$$
$$= \sum_{m_2} \frac{(-1)^{j_2+m_2}}{\sqrt{2j_2+1}} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_2,-m_2\rangle_3 |0,0\rangle_4$$

The other option is to choose j_3:

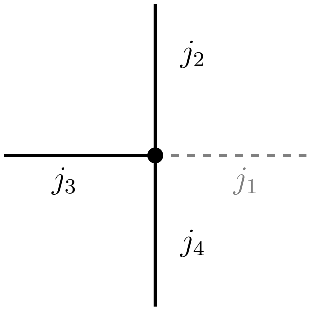
$$\begin{aligned}|0,0\rangle &= \sum_{m_3}(-1)^{j_3-m_3}\sqrt{2j_3+1}|0,0\rangle_1|j_3,-m_3\rangle_2|j_3,m_3\rangle_3|0,0\rangle_4 \\ &\begin{pmatrix} j_3 & j_3 & 0 \\ -m_3 & m_3 & 0\end{pmatrix}\begin{pmatrix} j_3 & j_3 & 0 \\ -m_3 & m_3 & 0\end{pmatrix} \\ &= \sum_{m_3}(-1)^{+j_3+m_3}\sqrt{2j_3+1}|0,0\rangle_1|j_3,-m_3\rangle_2|j_3,-m_3\rangle_3|0,0\rangle_4\frac{(-1)^{2j_3-2m_3}}{2j_3+1} \\ &= \sum_{m_3}\frac{(-1)^{j_3-m_3}}{\sqrt{2j_3+1}}|0,0\rangle_1|j_3,-m_3\rangle_2|j_3,m_3\rangle_3|0,0\rangle_4\end{aligned}$$

These two are essentially equal since we sum over all m values.



$$
\begin{aligned}
|0,0\rangle = &\sum_{m_1,m_2,m_3,m_{12}} (-1)^{-j_1+j_2-j_3+j_{12}-m_{12}}\sqrt{2j_{12}+1}|j_1,m_1\rangle_1|j_2,m_2\rangle_2|j_3,m_3\rangle_3|0,0\rangle_4 \\
&\begin{pmatrix} j_1 & j_2 & j_{12} \\ m_1 & m_2 & -m_{12} \end{pmatrix}\begin{pmatrix} j_3 & 0 & j_{12} \\ m_3 & 0 & m_{12} \end{pmatrix} \\
= &\sum_{m_1,m_2,m_3} (-1)^{-j_1+j_2+m_3}\sqrt{2j_3+1}|j_1,m_1\rangle_1|j_2,m_2\rangle_2|j_3,m_3\rangle_3|0,0\rangle_4 \\
&\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}\begin{pmatrix} j_3 & 0 & j_3 \\ m_3 & 0 & -m_3 \end{pmatrix} \\
= &\sum_{m_1,m_2,m_3} (-1)^{-j_1+j_2+j_3}|j_1,m_1\rangle_1|j_2,m_2\rangle_2|j_3,m_3\rangle_3|0,0\rangle_4 \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}
\end{aligned}
$$

This solution matches our result from before.



$$|0,0\rangle = \sum_{m_2,m_3,m_4,m_{12}} (-1)^{j_2-j_3+j_4+j_{12}-m_{12}} \sqrt{2j_{12}+1} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_3,m_3\rangle_3 |j_4,m_4\rangle_4$$
$$\begin{pmatrix} 0 & j_2 & j_{12} \\ 0 & m_2 & -m_{12} \end{pmatrix} \begin{pmatrix} j_3 & j_4 & j_{12} \\ m_3 & m_4 & m_{12} \end{pmatrix}$$
$$= \sum_{m_2,m_3,m_4} (-1)^{2j_2-j_3+j_4-m_2} \sqrt{2j_2+1} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_3,m_3\rangle_3 |j_4,m_4\rangle_4$$
$$\begin{pmatrix} 0 & j_2 & j_2 \\ 0 & m_2 & -m_2 \end{pmatrix} \begin{pmatrix} j_3 & j_4 & j_2 \\ m_3 & m_4 & m_2 \end{pmatrix}$$
$$= \sum_{m_2,m_3,m_4} (-1)^{2j_2-j_3+j_4-m_2} \sqrt{2j_2+1} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_3,m_3\rangle_3 |j_4,m_4\rangle_4$$
$$\frac{(-1)^{j_2-m_2}}{\sqrt{2j_2+1}} \begin{pmatrix} j_3 & j_4 & j_2 \\ m_3 & m_4 & m_2 \end{pmatrix}$$
$$= \sum_{m_2,m_3,m_4} (-1)^{3j_2-j_3+j_4-2m_2} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_3,m_3\rangle_3 |j_4,m_4\rangle_4 \begin{pmatrix} j_3 & j_4 & j_2 \\ m_3 & m_4 & m_2 \end{pmatrix}$$

There is the problem of the extra factor of $(-1)^{2j_2-2m_2}$. This will show up a lot in the future, so it will be useful to derive a new identity based on what we see here. Since $j = 0, 1/2, 1, 3/2, ...$, and $m = -j, -j+1, ..., j-1, j$, we can say that $m = -j+k$, where $k = 0, 1, 2, ..., 2j$. This allows us to write $j - m = 2j - k = 2j, 2j-1, ..., 0$. This works out to always giving us integers since j is half integer. This means that $(-1)^{2j_2-2m_2} = 1$. Therefore, our final answer should match the results from section 1.3:

$$|0,0\rangle = \sum_{m_2,m_3,m_4} (-1)^{j_2-j_3+j_4} |0,0\rangle_1 |j_2,m_2\rangle_2 |j_3,m_3\rangle_3 |j_4,m_4\rangle_4 \begin{pmatrix} j_3 & j_4 & j_2 \\ m_3 & m_4 & m_2 \end{pmatrix}$$

References

[1] Sarmed A Rahman et al. “Self-mitigating Trotter circuits for SU(2) lattice gauge theory on a quantum computer”. In: *Phys. Rev. D* 106.7 (2022), p. 074502. DOI: 10.1103/PhysRevD.106.074502. arXiv: 2205.09247 [hep-lat].

[2] B.R. Martin and G. Shaw. *Particle Physics*. Manchester Physics Series. Wiley, 2017. ISBN: 9781118911907. URL: https://books.google.ca/books?id=ielRDQAAQBAJ.

[3] G. Münster and M. Walzl. *Lattice Gauge Theory - A short Primer*. 2000. arXiv: hep-lat/0012005 [hep-lat].

[4] Christof Gattringer and Kurt Langfeld. “Approaches to the sign problem in lattice field theory”. In: *International Journal of Modern Physics A* 31.22 (Aug. 2016), p. 1643007. DOI: 10.1142/s0217751x16430077. URL: https://doi.org/10.1142%2Fs0217751x16430077.

[5] Gaopei Pan and Zi Yang Meng. “The sign problem in quantum Monte Carlo simulations”. In: *Encyclopedia of Condensed Matter Physics*. Elsevier, 2024, pp. 879–893. ISBN: 9780323914086. DOI: 10.1016/b978-0-323-90800-9.00095-0. URL: http://dx.doi.org/10.1016/B978-0-323-90800-9.00095-0.

[6] Lena Funcke et al. “Review on Quantum Computing for Lattice Field Theory”. In: *Proceedings of The 39th International Symposium on Lattice Field Theory — PoS(Lattice2022)*. Sissa Medialab, Feb. 2023. DOI: 10.22323/1.430.0228. URL: https://doi.org/10.22323%2F1.430.0228.

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

[8] Randy Lewis and R. M. Woloshyn. *A qubit model for U(1) lattice gauge theory*. 2019. DOI: 10.48550/ARXIV.1905.09789. URL: https://arxiv.org/abs/1905.09789.

[9] Christine Muschik et al. “U(1) Wilson lattice gauge theories in digital quantum simulators”. In: *New Journal of Physics* 19.10 (Oct. 2017), p. 103020. DOI: 10.1088/1367-2630/aa89ab. URL: https://doi.org/10.1088%2F1367-2630%2Faa89ab.

[10] Roland C. Farrell et al. “Scalable Circuits for Preparing Ground States on Digital Quantum Computers: The Schwinger Model Vacuum on 100 Qubits”. In: (Aug. 2023). arXiv: 2308.04481 [quant-ph].

[11] Yasar Y. Atas et al. “SU(2) hadrons on a quantum computer via a variational approach”. In: *Nature Communications* 12.1 (Nov. 2021), p. 6499. ISSN: 2041-1723. DOI: 10.1038/s41467-021-26825-4. URL: https://doi.org/10.1038/s41467-021-26825-4.

[12] Zohreh Davoudi, Indrakshi Raychowdhury, and Andrew Shaw. “Search for efficient formulations for Hamiltonian simulation of non-Abelian lattice gauge theories”. In: *Physical Review D* 104.7 (Oct. 2021). DOI: 10.1103/physrevd.104.074505. URL: https://doi.org/10.1103%2Fphysrevd.104.074505.

[13] Tim Byrnes and Yoshihisa Yamamoto. “Simulating lattice gauge theories on a quantum computer”. In: *Physical Review A* 73.2 (Feb. 2006). DOI: 10.1103/physreva.73.022328. URL: https://doi.org/10.1103%2Fphysreva.73.022328.

[14] Natalie Klco, Martin J. Savage, and Jesse R. Stryker. “SU(2) non-Abelian gauge field theory in one dimension on digital quantum computers”. In: *Physical Review D* 101.7 (Apr. 2020). DOI: 10.1103/physrevd.101.074512. URL: https://doi.org/10.1103%2Fphysrevd.101.074512.

[15] Sarmed A Rahman et al. “SU(2) lattice gauge theory on a quantum annealer”. In: *Phys. Rev. D* 104.3 (2021), p. 034501. DOI: 10.1103/PhysRevD.104.034501. arXiv: 2103.08661 [hep-lat].

[16] Yeinzon Rodriguez. *A New Pedagogical Way of Finding Out the Gauge Field Strength Tensor in Abelian and Non-Abelian Local Gauge Field Theories.* 2016. arXiv: 1512.03827 [physics.gen-ph].

[17] C. N. Yang and R. L. Mills. "Conservation of Isotopic Spin and Isotopic Gauge Invariance". In: *Phys. Rev.* 96 (1 Oct. 1954), pp. 191–195. DOI: 10.1103/PhysRev.96.191. URL: https://link.aps.org/doi/10.1103/PhysRev.96.191.

[18] Heinz J. Rothe. *Lattice Gauge Theories : An Introduction (Fourth Edition).* Vol. 43. World Scientific Publishing Company, 2012. ISBN: 978-981-4365-87-1, 978-981-4365-85-7. DOI: 10.1142/8229.

[19] Ramamurti Shankar. *Principles of quantum mechanics.* New York, NY: Springer, 1994.

[20] Murray Gell-Mann. "Symmetries of Baryons and Mesons". In: *Phys. Rev.* 125 (3 Feb. 1962), pp. 1067–1084. DOI: 10.1103/PhysRev.125.1067. URL: https://link.aps.org/doi/10.1103/PhysRev.125.1067.

[21] Tom Lancaster and Stephen J. Blundell. *Quantum Field Theory for the Gifted Amateur.* Oxford University Press, 2014. ISBN: 978-0-19-969933-9.

[22] Michael E. Peskin and Daniel V. Schroeder. *An Introduction to quantum field theory.* Reading, USA: Addison-Wesley, 1995. ISBN: 978-0-201-50397-5.

[23] Kenneth G. Wilson. "Confinement of quarks". In: *Phys. Rev. D* 10 (8 Oct. 1974), pp. 2445–2459. DOI: 10.1103/PhysRevD.10.2445. URL: https://link.aps.org/doi/10.1103/PhysRevD.10.2445.

[24] Christof Gattringer and Christian B. Lang. *Quantum chromodynamics on the lattice.* Vol. 788. Berlin: Springer, 2010. ISBN: 978-3-642-01849-7, 978-3-642-01850-3. DOI: 10.1007/978-3-642-01850-3.

[25] Hans Mathias Mamen Vege. "Solving SU(3) Yang-Mills theory on the lattice: a calculation of selected gauge observables with gradient flow". MA thesis. U. Oslo (main), 2019.

[26] Steven Weinberg. *The Quantum Theory of Fields*. Vol. 1. Cambridge University Press, 1995. DOI: 10.1017/CBO9781139644167.

[27] John Kogut and Leonard Susskind. “Hamiltonian formulation of Wilson’s lattice gauge theories”. In: *Phys. Rev. D* 11 (2 Jan. 1975), pp. 395–408. DOI: 10.1103/PhysRevD.11.395. URL: https://link.aps.org/doi/10.1103/PhysRevD.11.395.

[28] John B. Kogut. “An introduction to lattice gauge theory and spin systems”. In: *Rev. Mod. Phys.* 51 (4 Oct. 1979), pp. 659–713. DOI: 10.1103/RevModPhys.51.659. URL: https://link.aps.org/doi/10.1103/RevModPhys.51.659.

[29] John B. Kogut, Donald K. Sinclair, and Leonard Susskind. “A Quantitative Approach to Low-Energy Quantum Chromodynamics”. In: *Nucl. Phys. B* 114 (1976), pp. 199–236. DOI: 10.1016/0550-3213(76)90586-1.

[30] Elías F. Combarro et al. “A report on teaching a series of online lectures on quantum computing from CERN”. In: *The Journal of Supercomputing* 77.12 (Dec. 2021), pp. 14405–14435. ISSN: 1573-0484. DOI: 10.1007/s11227-021-03847-9. URL: https://doi.org/10.1007/s11227-021-03847-9.

[31] Jarrod R. McClean, John A. Parkhill, and Alá n Aspuru-Guzik. “Feynman’s clock, a new variational principle, and parallel-in-time quantum dynamics”. In: *Proceedings of the National Academy of Sciences* 110.41 (Sept. 2013). DOI: 10.1073/pnas.1308069110. URL: https://doi.org/10.1073%2Fpnas.1308069110.

[32] Matthew Treinish. *Qiskit/qiskit-metapackage: Qiskit 0.44.0*. Version 0.44.0. July 2023. DOI: 10.5281/zenodo.8190968. URL: https://doi.org/10.5281/zenodo.8190968.

[33] Abhinav Kandala et al. “Error mitigation extends the computational reach of a noisy quantum processor”. In: *Nature* 567.7749 (Mar. 2019), pp. 491–495. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1040-7. URL: https://doi.org/10.1038/s41586-019-1040-7.

[34] Sergey Bravyi et al. "Mitigating measurement errors in multiqubit experiments". In: *Phys. Rev. A* 103 (4 Apr. 2021), p. 042605. DOI: 10.1103/PhysRevA.103.042605. URL: https://link.aps.org/doi/10.1103/PhysRevA.103.042605.

[35] Ying Li and Simon C. Benjamin. "Efficient Variational Quantum Simulator Incorporating Active Error Minimization". In: *Phys. Rev. X* 7 (2 June 2017), p. 021050. DOI: 10.1103/PhysRevX.7.021050. URL: https://link.aps.org/doi/10.1103/PhysRevX.7.021050.

[36] Joel J. Wallman and Joseph Emerson. "Noise tailoring for scalable quantum computation via randomized compiling". In: *Phys. Rev. A* 94 (5 Nov. 2016), p. 052325. DOI: 10.1103/PhysRevA.94.052325. URL: https://link.aps.org/doi/10.1103/PhysRevA.94.052325.

[37] Zhenyu Cai, Xiaosi Xu, and Simon C. Benjamin. "Mitigating coherent noise using Pauli conjugation". In: *npj Quantum Information* 6.1 (Feb. 2020), p. 17. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0233-0. URL: https://doi.org/10.1038/s41534-019-0233-0.

[38] Miroslav Urbanek et al. "Mitigating Depolarizing Noise on Quantum Computers with Noise-Estimation Circuits". In: *Phys. Rev. Lett.* 127 (27 Dec. 2021), p. 270502. DOI: 10.1103/PhysRevLett.127.270502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.270502.

[39] Paul E.S. Wormer and Josef Paldus. "Angular Momentum Diagrams". In: ed. by J.R. Sabin and E. Brändas. Vol. 51. Advances in Quantum Chemistry. Academic Press, 2006, pp. 59–123. DOI: https://doi.org/10.1016/S0065-3276(06)51002-0. URL: https://www.sciencedirect.com/science/article/pii/S0065327606510020.

[40] Igor Olegovich Cherednikov, Tom Mertens, and Frederik Van der Veken. *Wilson Lines in Quantum Field Theory*. Berlin, Boston: De Gruyter, 2020. ISBN: 9783110651690. DOI: doi:10.1515/9783110651690. URL: https://doi.org/10.1515/9783110651690.

[41] D. Robson and D. M. Webber. “Gauge Covariance in Lattice Field Theories”. In: *Z. Phys. C* 15 (1982), p. 199. DOI: `10.1007/BF01475006`.