UNIVERSITY OF CALIFORNIA
RIVERSIDE

Adiabatic Motion of Fault Tolerant Qubits

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Physics

by

David Edward Drummond

August 2014

Dissertation Committee:

Dr. Leonid P. Pryadko, Chairperson
Dr. Kirill Shtengel
Dr. Vivek Aji

The Dissertation of David Edward Drummond is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to acknowledge several people for their help and support in making this work possible. First, I would like to whole-heartedly thank Professors Leonid Pryadko and Kirill Shtengel, for their roles as advisors, mentors, and colleagues. They have patiently shared their time, expertise, and encouragement throughout my studies. I am constantly impressed by their depth of knowledge, their physical intuition, and their ability to convey difficult concepts in multiple ways, repeatedly offering explanations until one "clicked". I am truly grateful for their invaluable help in my growth, both professionally and personally, since I began my graduate studies.

I would also like to thank the other teachers that have helped me reach this point. In particular, I would like to thank Professor Angela Milano and Rick Salas for first sparking my interest in Math and Physics. I also want to thank Professor Mari-Anne Rosario for introducing me to Condensed Matter Physics and encouraging me to pursue my interests in Quantum Computing. Finally, I would like to thank all the professors at University of California, Riverside that have helped me: particularly, Professor Vivek Aji for offering clear explanations and advice in his Condensed Matter Field Theory courses, Professor Alexander Korotkov for his candid views on Quantum Information and science, and Professors Umar Mohideen, Bill Gary, and Kenneth Barish for their help and support to make me a better teacher.

Next, I would like to acknowledge my fellow students Tom, Annie, Brandon, Ian, Gian, Kyle, Ali and Harold for their reassuring support, entertaining stories, and valuable friendship. Specifically, I'd like to thank Ian McQueary and Harold Nguyen for generously giving their time and encouragement, for sharing many re-invigorating meals, and for always lending an empathetic ear. I would also like to thank my colleagues

Yafis Barlas, Amrit De, Alexey Kovalev and Chang-Yu Hou for their honest advice and guidance.

Last, but not least, I would like to thank my parents and my wife for their ongoing support during my research. There have been countless times when they motivated, inspired, and pushed me to continue working when I lost focus. When I was tired, they were there to revitalize me with their companionship; when I succeeded, they were there to cheer me on. I will always be grateful to them for their unconditional kindness, support and love.

Parts of this work have been published in the journal, Physical Review B [32, 31], and are used with permission under the guidelines of the American Physical Society. I would like to thank my co-authors Alexey A. Kovalev and Chang-Yu Hou, as well as my advisors, for their help in the development and analysis of Ref. [31]. This work was supported in part by the U.S. Army Research Office under Grant No. W911NF-11-1-0027, and by the NSF under Grants DMR-0748925 and 1018935, as well as the DARPA-QuEST program.

This work is dedicated to my wife and parents; their love and support made this possible.

ABSTRACT OF THE DISSERTATION

Adiabatic Motion of Fault Tolerant Qubits

by

David Edward Drummond

Doctor of Philosophy, Graduate Program in Physics
University of California, Riverside, August 2014
Dr. Leonid P. Pryadko, Chairperson

This work proposes and analyzes the adiabatic motion of fault tolerant qubits in two systems as candidates for the building blocks of a quantum computer. The first proposal examines a pair of electron spins in double quantum dots, finding that the leading source of decoherence, hyperfine dephasing, can be suppressed by adiabatic rotation of the dots in real space. The additional spin-orbit effects introduced by this motion are analyzed, simulated, and found to result in an infidelity below the error-correction threshold. The second proposal examines topological qubits formed by Majorana zero modes theorized to exist at the ends of semiconductor nanowires coupled to conventional superconductors. A model is developed to design adiabatic movements of the Majorana bound states to produce entangled qubits. Analysis and simulations indicate that these adiabatic operations can also be used to demonstrate entanglement experimentally by testing Bell's theorem.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Quantum Information

1.1 Historical Overview of Quantum Computing

The fundamentally different physics at the atomic scale has interested and confounded scientists ever since the discovery of quantum mechanics in the first quarter of the twentieth century. While physicists did not fully reconcile the philosophical ramifications of the theory of quantum mechanics, they came to terms with it and made incredibly successful predictions in the decades that followed. However, in 1964 Bell made the philosophical differences between the classical and quantum worlds more concrete by conceiving his famous inequality that could be experimentally tested [7]. Subsequent experimental results were consistent with quantum mechanical predictions, suggesting that nanoscopic systems are inherently different from the macroscopic systems that our every-day intuition is based on.

This difference led physicists like Poplavskii and Feynman to believe that it was physically impossible to simulate a quantum mechanical system on a classical computer

[35, 88], suggesting that a new type of computer constructed from a quantum system may be more powerful than a classical Turing machine. Thoughout the 1970's and 1980's many physicists established quantum information theory and elaborated on the specific requirements and potential of a quantum computer [119, 53, 51, 75, 9, 120, 29, 8]. Specifically, Deutsch showed that a quantum Turing machine could not only simulate a classical Turing machine, but could fundamentally out-perform conventional computers at certain tasks [25].

In the 1990's, physicists developed several algorithms that could be implemented on a quantum computer exponentially faster than classical computers [34, 27, 109, 46]. In particular, Shor discovered a quantum algorithm in 1994 capable of factoring large numbers in reasonable times, which was previously considered impossible and thus the crux of many of the cryptographic systems still in use today [106]. These discoveries spurred further interest into the theory of quantum information and the pursuit of a physical realization of a quantum computer.

Over the past 20 years, there have been many proposals for systems that could perform quantum computations and numerous experimental attempts at constructing them. While important developments in the field of quantum error correction made building a quantum computer more feasible, it has proven to be one of the most challenging goals in contemporary physics. At the heart of the difficulty is the delicate requirement of controlling individual particles, yet shielding them from their local environment. In recent years, experimental progress has been made through both incredible effort and advances in nano-technology and material science [80, 112, 28, 3, 30, 4]. Nonetheless, it is far from clear when, or whether, these advances will produce a quantum computer that surpasses the power of classical computers.

At the same time, theorists have proposed innovative new approaches that use the concept of fault tolerance, building a quantum computer from systems that are inherently immune to certain environmental factors [89]. Rather than using a single two-state quantum system (i.e., qubits), these proposals encode quantum information in multiple qubits; some non-local degrees of freedom of a collection of qubits are tolerant to errors from any single local disturbances. Though these proposals often require more complicated setups or more advanced physics, fault tolerant quantum information offers an additional avenue towards a quantum computer.

This dissertation introduces and analyzes two proposals for fault tolerant quantum systems: the spin state of two electrons in a double quantum dot, and the particle parity of a topological state in a Majorana wire. In particular, this dissertation analyzes the gradual, adiabatic motion of their constituents that is crucial to both these proposals. The remainder of this introduction will briefly review the fundamentals of quantum information necessary to understand the two proposals discussed in the later chapters. For a more thorough review of quantum information, the interested reader should see the texts by Nielsen and Chuang[84] or Mermin[79].

1.2 Fundamentals of Quantum Information

1.2.1 Qubits

Rather than the classical bit that only takes the values of 0 or 1, the fundamental unit of quantum information, known as a quantum bit or qubit, can be a superposition state, $|\psi\rangle$, of two orthonormal basis states $|0\rangle$ and $|1\rangle$ of a quantum system

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1.1}$$

where α and β are complex coefficients that satisfy

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.2}$$

When the physical system is measured, the state is projected to either $|0\rangle$ or $|1\rangle$ with probability $|\alpha|^2$ or $|\beta|^2$, respectively. The overall phase of $|\psi\rangle$ is not physically relevant since the probabilties depend only on the absolute value of inner products. For example, the probability of measuring a qubit in the state $|0\rangle$ is

$$\begin{aligned} \left|\langle 0|\psi\rangle\right|^2 &= \left|\langle 0|(\alpha|0\rangle + \beta|1\rangle)\right|^2 & (1.3)\\ &= \left|\alpha\langle 0|0\rangle + \beta\langle 0|1\rangle\right|^2 & (1.4)\\ &= |\alpha|^2. & (1.5) \end{aligned}$$

The choice of basis states used to describe a qubit is not unique; any qubit can be written in terms of any other orthonormal bases such as $|\tilde{0}\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|\tilde{1}\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. In practice, the preferred basis choice usually depends on the physical method of measurement. For example, when encoding information in the spin of electrons, the measurement device typically separates the spin in two directions, which are conventionally defined as $|0\rangle$ and $|1\rangle$.

Mathematically, the superposition in Eq. (1.1) can be represented by a column vector

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \implies \phi = \begin{pmatrix} \alpha \\ \\ \beta \end{pmatrix}. \tag{1.6}$$

This representation is generalized to multiple qubits through the Kronecker product. For example, two general qubits can be represented as

$$|\psi_1\rangle \otimes |\psi_0\rangle \sim \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} = \begin{pmatrix} \alpha_1 \begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} \\ \beta_1 \begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \alpha_1\alpha_0 \\ \alpha_1\beta_0 \\ \beta_1\alpha_0 \\ \beta_1\beta_0 \end{pmatrix}, \tag{1.7}$$

which maintains the binary convention from classical bits. For example, the state $|10\rangle \equiv |1\rangle \otimes |0\rangle$, which is the binary version of 2, is represented as

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \tag{1.8}$$

with only element 2 non-zero (with the count starting from 0). The Kronecker product and this binary rule both generalize as one would expect, for any number of qubits.

1.2.1.1 Superposition

While the probabilistic interpretation of the coefficients in Eq. (1.1) may seem similar to a statistical ensemble of states, superposition is fundamentally different. In addition to the ratio of the coefficient magnitudes, the relative phase difference between α and β also contains information that can lead to physically measurable results. For example, it is tempting to consider the state $(|0\rangle + |1\rangle)/\sqrt{2}$ as simply an equal mixture of $|0\rangle$ and $|1\rangle$. However, this description doesn't discern between the previous state and $(|0\rangle - |1\rangle)/\sqrt{2}$, despite the fact that these two states are distinct in another orthogonal basis. To make this distinction concrete, a superposition such as Eq. (1.1) is known as

a pure state, while a classical mixture of states from an ensemble is known as a mixed state.

Part of the power of quantum algorithms comes from performing computations with superpositions of the basis states. For example, instead of separately calculating the values $f(0)$ and $f(1)$ of some computationally expensive function $f(x)$, a quantum computer can find both values simultaneously by computing the function with a superposition of $|0\rangle$ and $|1\rangle$. More so, a superposition of the two-qubit states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ can be used to simultaneously calculate four values and generally a quantum computer with n qubits could simultaneously calculate 2^n values in principle. This exponential scaling, known as quantum parallelism, is at the heart of the incredible speed-up of quantum algorithms.

However, this description is slightly specious since only one of the results is accessible; when the result is measured the final state is projected to just one of the calculated values. This projection also makes it impossible to check a calculation in the middle of a computation since any intermediate measurement would ruin the superposition required in later steps. Fortunately, this restriction can be overcome with clever algorithms that solve a given problem with just a small subset of the function values. In practice, non-trivial algorithms like those designed by Grover or Shor don't solve problems deterministically, but rather find the correct solution in a relatively small number of attempts, with a probability that tends towards one [46, 106].

It should be noted that this "collapse" of a superposition to a single state upon measurement can also be useful. For example, if a secure cryptographic key is encoded in a superposition, any unintended eavesdropping will collapse the state and can be detected before transimitting sensitive messages [9]. Alternatively, a specific quantum

state can be initialized by projectively measuring the relevant system until the desired state is found.

1.2.1.2 Entanglement

Another feature of quantum mechanics that is responsible for the potential of quantum computers is the property that some multi-particle systems can only be described as a single "entangled" state. For example, the maximally entangled two-qubit system

$$|\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \tag{1.9}$$

cannot be written as separable states (i.e., there are no $|\psi_1\rangle$, $|\psi_0\rangle$ such that $|\Phi^-\rangle = |\psi_1\rangle \otimes |\psi_0\rangle$). Thus measuring one of the qubits automatically determines the outcome of the other, even if the qubits have been spatially separated after they were entangled [33]. This "spooky action at a distance" can be used to implement useful applications like quantum teleportation and cryptography that would be impossible on classical computers [34]. More so, entanglement plays a crucial role in many algorithms and other applications such as superdense coding [10, 55]. For this reason, preparing and demonstrating entanglement is an essential ingredient for any quantum computing proposal. The fundamental distinction between entangled and classical systems can be shown using Bell's theorem, as discussed further in Sec. 1.3.

1.2.1.3 Bloch Sphere Visualization

A single pure state can be visualized by using the real parameters

$$\theta = 2\tan^{-1}(|\beta|/|\alpha|) \tag{1.10}$$

$$\phi = \arg(\beta) - \arg(\alpha) \tag{1.11}$$

to write a qubit as

$$|\phi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle. \tag{1.12}$$

Thus θ and ϕ can serve as the polar and azimuthal angles, respectively, on the surface of what is known as the Bloch sphere, with $|0\rangle$ and $|1\rangle$ at the North and South pole, respectively (see Fig. 1.1). The Bloch sphere representation is quite helpful for understanding single qubits and how they evolve. Unfortunately, this simple visualization doesn't easily generalize to multiple qubits since any visualization would have to contain exponentially more information and account for the possibility of entanglement.

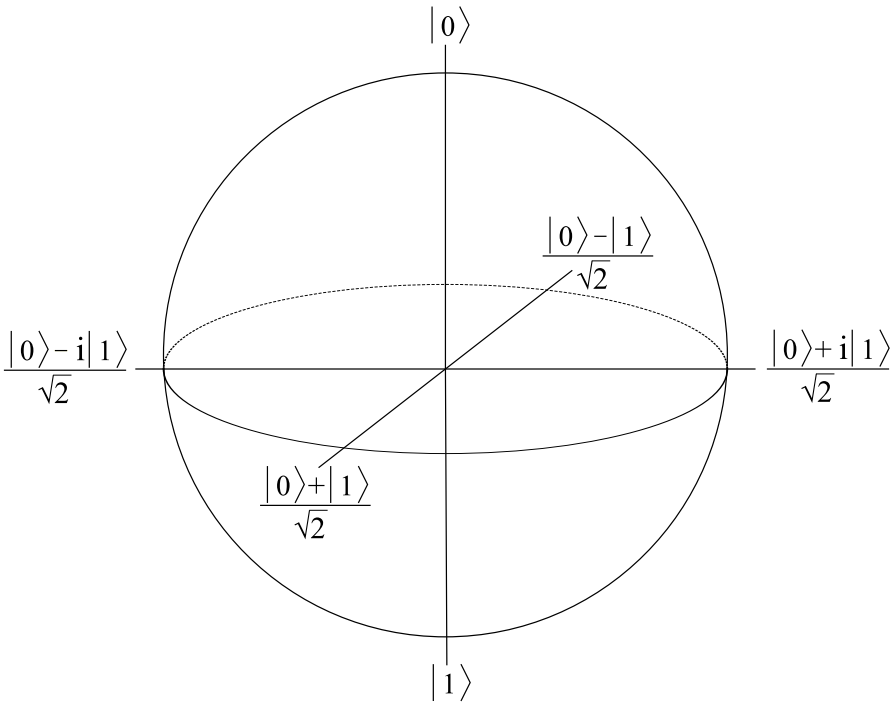

Figure 1.1: A single qubit can be represented as a point on the surface of the Bloch sphere. The latitude of the point describes the probability of measuring $|0\rangle$ or $|1\rangle$; Northern points have higher probabilites of measuring $|0\rangle$. The longitude describes the phase difference between the states $|0\rangle$ and $|1\rangle$ in the superposition.

1.2.2 Quantum Operations

Despite the complexity of qubits compared to classical bits, any quantum computation can be performed with arbitrary accuracy with a small set of operations known as universal quantum operations [26, 22].

1.2.2.1 Single Qubit Operations

The simplest operations involve a single qubit at a time, performing operations such as a bit flip that switches $|0\rangle \leftrightarrow |1\rangle$ or a phase shift that switches $|1\rangle \leftrightarrow -|1\rangle$. More generally, any single qubit operation can be thought of as an a rotation of the states on the Bloch sphere. With this in mind, Euler's rotation theorem ensures that any rotation can be generated if one can perform arbitrary rotations about two orthogonal axes.

Mathematically, single qubit rotations can be generated by the orthogonal Pauli spin matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1.13}$$

which are sometimes referred to as the X, Y, and Z operations. It is easy to see that the X operation performs the bit flip when acting on a general qubit

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}, \tag{1.14}$$

effectively rotating the state π radians about the x-axis of the Bloch sphere. Similarly, the Y and Z operations correspond to π-rotations about their respective axes.

In general, any single-qubit operation corresponds to a unitary matrix and can be performed by allowing an appropiate Hamiltonian to evolve the qubit state. For example, a two-state quantum system driven by a harmonic electric field with a

frequency near the splitting energy of the two states, $\hbar\omega \sim \Delta E$, causes the occupation probabilities of the two states to oscillate. This periodic behavior, known as Rabi oscillation, can be described as the operation of the unitary evolution

$$U(t) = e^{-i\omega t\sigma_x} \tag{1.15}$$

on the two-level system. The matrix representation of this exponential can be found by noting that even powers of the Pauli matrices are just the identity, $\sigma_i^2 = \mathbb{1}$, so the Taylor series can be written

$$\begin{aligned} e^{-i\omega t\sigma_x} &= \mathbb{1} + (-i\omega t)\sigma_x + \frac{(-i\omega t)^2}{2!}\mathbb{1} + \frac{(-i\omega t)^3}{3!}\sigma_x + ... && (1.16)\\ &= \mathbb{1}\left(1 - \frac{(\omega t)^2}{2!} + ...\right) - i\sigma_x\left((\omega t) - \frac{(\omega t)^3}{3!} + ...\right) && (1.17)\\ &= \mathbb{1}\cos(\omega t) - i\sigma_x \sin(\omega t) && (1.18)\\ &= \begin{pmatrix} \cos(\omega t) & -i\sin(\omega t) \\ -i\sin(\omega t) & \cos(\omega t) \end{pmatrix}. && (1.19) \end{aligned}$$

Thus an X rotation of any given angle can be performed in this system by oscillating an electric field for the appropriate duration. Similarly, Y and Z operations can be performed by finding or designing a system with the appropriate terms in the Hamiltonian.

1.2.2.2 Multi-Qubit Operations

Single-qubit operations alone are not sufficient to perform most quantum algorithms; multi-qubit operations are necessary in order to construct a universal quantum computer [27]. Specifically most algorithms use "controlled" gates that perform an operation on a target qubit if and only if another control qubit is $|1\rangle$. For example, the

controlled-not operation, also known as CNOT, given by

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{1.20}$$

performs an X rotation (traditionally known as the NOT gate in the classical computing context) on the right, target qubit if and only if the left, control qubit is 1. Explicitly,

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ 0 \\ 0 \end{pmatrix}, \tag{1.21}$$

while

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \beta \\ \alpha \end{pmatrix}. \tag{1.22}$$

More generally, any single-qubit operation U can be applied to the target qubit, forming the controlled-U operation, represented in block matrix form as

$$\begin{pmatrix} \mathbb{1} & 0 \\ 0 & U \end{pmatrix}. \tag{1.23}$$

Similarly, it is easy to check that the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{1.24}$$

also represents a CNOT operation, but with the target and control qubits interchanged. These controlled operations add multi-particle interactions, enabling logical operations where the outcome of one qubit depends on the input of another qubit. Together with the single-qubit operations, any non-trivial controlled operation forms a universal set of quantum operations capable of performing any quantum computation [27].

The importance of controlled operations can be seen by observing that they can be used, together with single-qubit rotations, to entangle two qubits. For example, a qubit initialized in the state $|0\rangle$, then acted on by a $\pi/2$-angle Y rotation, given by

$$e^{-i\frac{\pi}{2}\sigma_y} = \frac{\sqrt{2}}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{1.25}$$

[cf. Eq. (1.16)], produces the superposition $(|0\rangle - |1\rangle)/\sqrt{2}$. If this state is used as the control qubit of a CNOT operation, with a target qubit initialized in the state $|0\rangle$, the resulting state is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} \sqrt{2}/2 \\ 0 \\ -\sqrt{2}/2 \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 \\ 0 \\ 0 \\ -\sqrt{2}/2 \end{pmatrix}. \tag{1.26}$$

which represents $(|00\rangle - |11\rangle)/\sqrt{2}$, the maximally entangled state $|\Phi^-\rangle$.

The fact that controlled operations like CNOT can entangle qubits leads to a very useful consequence: together with entangled qubits, single-qubit rotations are sufficient to form a universal set of quantum operations [12]. For example, in the context of qubits formed by Ising anyons discussed in Ch. 3, universal quantum computation can only be achieved by complementing the topologically protected operations with entangled qubits in a process known as magic state distillation [11].

1.2.3 Decoherence

Perhaps the most difficult aspect of constructing a quantum computer is to overcome decoherence, undesired interactions of the qubits with their environment. Almost all scalable physical systems are incredibly sensitive to environmental influences such as stray electromagnetic fields, disorder, and finite-temperature effects. One can consider the environment as a large set of unknown qubits that entangle themselves with the computational qubits and thus introduce unpredictable errors into the calculations.

In order to reliably execute algorithms, a quantum computer needs to perform several hundreds or thousands of operations without errors. Unfortunately, there is a difficult trade-off; systems with weaker environmental interactions tend to have longer operation times. Thus, longer decoherence times usually come at the cost of longer operation times.

1.2.3.1 Fidelity

In order to track progress in preventing decoherence, a number of mathematical distances and measures have been introduced in quantum information [41]. While the details of these measures often depend on the relevant physical system, the most ubiquitous measure is known as "fidelity", which generally describes the similarity of two quantum states. This dissertation only considers the fidelity that compares a pure state before (ρ_0) and after unitary evolution ($U\rho_0U^\dagger$), which leads to the simpler definition [84]

$$F = \mathrm{Tr}(\rho_0 U \rho_0 U^\dagger), \tag{1.27}$$

where ρ_0 is the density matrix of the pure state, U is the time-evolution operator, and "Tr" is the trace operator. Since the initial state is not always known a priori, it is

often useful to average the above fidelity over the subspace of possible pure states to calculate the average fidelity, $\langle F \rangle$, as discussed in more detail for the relevant systems in Sec. 2.4 and Sec. 4.4. Under these assumptions, the fidelity describes the ability of a system to preserve the value of its computational qubits. In practice, the fidelity can be quite close to 1, so the infidelity, $1 - F$, is usually reported instead.

1.2.3.2 Quantum Error Correction

Even with significant improvements in fidelity, quantum algorithms would be severely restricted without the ability to correct errors during computations. However, quantum error correction is much more complicated than its classical counterpart. A quantum error, which can involve both a bit-flip and a phase-shift, cannot be simply reversed between steps of an algorithm since measuring a superposition projects the state to either $|0\rangle$ or $|1\rangle$, losing the information contained in the superposition and destroying any entanglement required for further steps. Furthermore, it is impossible to copy a general superposition prior to measuring for errors, due to a quantum mechanical rule known as the no-cloning theorem [120, 29].

Nevertheless, there are more advanced ways to correct errors by encoding the information of a single qubit into several entangled qubits using quantum error correction codes [107, 111]. While these codes require many more qubits, they make it possible to find the error syndrome, which can be used to detect and correct errors without directly measuring the state. Since these codes are self-correcting in a sense, quantum error correction codes are one form of fault tolerant qubits. For a more thorough review of the field of quantum error correction, the interested reader should read the overview by Gottesman [44].

Crucially, quantum error correction codes can be concatenated by encoding one logical qubit with several physical qubits, then using several of those coded qubits to create another layer of coded qubits, continuing in this manner indefinitely until the fidelity is sufficiently high. This repeated use of codes only helps if the increased overhead of operations needed for the coding scheme does not cause more errors than it corrects. While the details of this requirement depend on the specific system and type of errors possible, this important threshold for the infidelity per operation typically ranges from 10^{-4} to 10^{-2}. In practice, the infidelity needs to be well below the threshold to avoid needing several levels of error correction codes, and thus, a prohibitve number of qubits. For this reason, the goal of most quantum computing proposals is to perform quantum operations while remaining below this infidelity threshold, which is the focus of the proposal discussed in Ch. 2.

1.3 Bell's Theorem

This section briefly reviews Bell's theorem and the relevant entanglement inequalities. It only covers the basic aspects needed for the proposal in Ch. 4; the interested reader should refer to the numerous works on the topic for a more comprehensive review [90, 78, 104, 79, 84].

1.3.1 Bell's Inequality

To derive a simplified version of Bell's inequality, consider an experiment that separately measures each qubit of the state $|\Phi^-\rangle = (|00\rangle - |11\rangle)/\sqrt{2}$ with one of three different methods, denoted a, b, and c. When both qubits are measured using the same method, the results are always the same. However, when the two qubits are measured

using different methods, the results are completely uncorrelated. Thus the possible results for one qubit measurement depend on what method is used for the other qubit, even if the measurement events are well separated spatially (i.e., space-like).

Einstein, Podolsky, and Rosen famously objected to this type of non-local behavior [33], citing it as motivation for a more complete theory that removes the probabilistic nature of quantum mechanics by introducing "hidden variables". Hidden variable theories predict, with full certainty, the outcomes of different measurement methods on a single qubit, even though only one measurement at a time is possible. Bell's theorem states that any local hidden variable theory makes predictions that are inconsistent with quantum mechanics [7]. Thus any experiment that agrees with quantum mechanics rather than hidden variable theories, implies that the qubits in the system are entangled.

To see where the two theories are inconsistent, consider the interpretation of $|\Phi^-\rangle$ in hidden variable theories. Instead of a pure state, it is viewed as one instance from a classical ensemble of states, prepared with different hidden variables. If an experimentalist could measure a single preparation with all three methods at once, the two qubits' results would match for each method. In this view the two qubits only seem uncorrelated when using different methods, but are actually correlated regardless of the method chosen. Thus the possible results of one qubit measurement don't depend on the method chosen for the other.

While this interpretation avoids non-local behavior, it replaces a superposition of states with a classical ensemble. Thus any single preparation in the ensemble must be either 0 or 1, not a more general superposition. Since there are three measurement methods, but only two possible outcomes, the pigeonhole principle states that at least two of the methods must give matching results. By defining $P_=(a, b)$ as the probability that the results match when one qubit is measured with a and the other is measured

with b, this statement can be written

$$P_{=}(a,b) + P_{=}(b,c) + P_{=}(a,c) \geq 1, \tag{1.28}$$

which is one version of Bell's inequality. Meanwhile, quantum mechanics predicts that this inequality is invalid for certain measurement methods, which demonstrates Bell's theorem. Specifically, each of the probabilities can be 1/4, leading to a violation of the inequality by 25%.

1.3.2 Clauser-Horne-Shimony-Holt Inequality

While this inequality can be tested experimentally in principle, it requires method b to be tested for both qubits, which would be difficult to accomplish exactly in many proposals. Instead, consider the case where the left qubit of $|\Phi^-\rangle$ is measured with either method L_1 or L_2, while the right qubit is measured using either method R_1 or R_2. Without superposition, each hidden variable preparation of the left qubit must have either $L_1 = 0$ or $L_1 = 1$, meaning that measuring the left qubit with method L_1 would yield 0 or 1, respectively. It is simpler to derive the inequality by considering the parity of these quantities so we use 1 and -1 for even and odd parity, respectively, for the remainder of this appendix. Thus, each preparation must have L_1, L_2, R_1, and R_2 as either 1 or -1 according to the hidden variable interpretation.

Consider the quantities L_1+L_2 and L_1-L_2; either $L_1+L_2 = \pm 2$ and $L_1-L_2 = 0$, or $L_1 + L_2 = 0$ and $L_1 - L_2 = \pm 2$. This implies that the quantity

$$|(L_1 + L_2)R_1 + (L_1 - L_2)R_2| = 2 \tag{1.29}$$

for each preparation since one of the terms vanishes in either case.

If an experimentalist could measure the qubits with more than one method at a time, this prediction could be tested directly. Instead, one must extract a statistical

prediction that only requires a single measurement of each qubit for any given preparation. With that in mind, note that the expectation value of any constant is simply that constant, and any variable X satisfies $|\langle X\rangle| \leq \langle |X|\rangle$ for any probability distribution. Applying these arguments to Eq. (1.29) yields the eponymous inequality first derived by Clauser, Horne, Shimony, and Holt[19]

$$|\langle L_1, R_1\rangle + \langle L_2, R_1\rangle + \langle L_1, R_2\rangle - \langle L_2, R_2\rangle| \leq 2, \tag{1.30}$$

where $\langle L, R\rangle = P_{=}(L, R) - P_{\neq}(L, R)$ is the expectation value for the combined parity of the left and right qubits when measured with methods L and R, respectively. Since each term only involves one measurement per qubit, it is possible to predict the left side of the inequality with quantum mechanics. For several measurement method combinations the left side can be as large as $2\sqrt{2} \simeq 2.8$, a violation of over 40%. Thus the quantum mechanical predictions are inconsistent with the local hidden variably theories and any experiment that violates the CHSH inequality negates the local hidden variable theories, demonstrating entanglement in the system.

Chapter 2

Double Quantum Dots Spatial Exchange Proposal

2.1 Introduction

In recent years there has been great interest in the prospect of using scalable solid-state devices to implement qubits for potential applications such as quantum computation. One promising candidate for a qubit is a pair of electron spins in quantum dots, which forms a fault-tolerant subspace that is immune to collective decoherence [69]. This section briefly reviews double quantum dots, though the interested reader should refer to texts such as Marder [76] or Refs. [63, 97] for more details.

2.1.1 Quantum Dot Background

The quantum dots in quantum computing proposals are built from semiconductor heterostructures that combine different semiconductor alloys with very similar properties to realize novel band structures not typically possible in nature. For example, since GaAs and AlAs have similar lattice sizes and structures, yet different band gaps,

constructing a layer of $Ga_{0.7}Al_{0.3}As$ on GaAs using molecular beam epitaxy leads to an abrupt change in the band structure at the junction (see Fig. 2.1B).

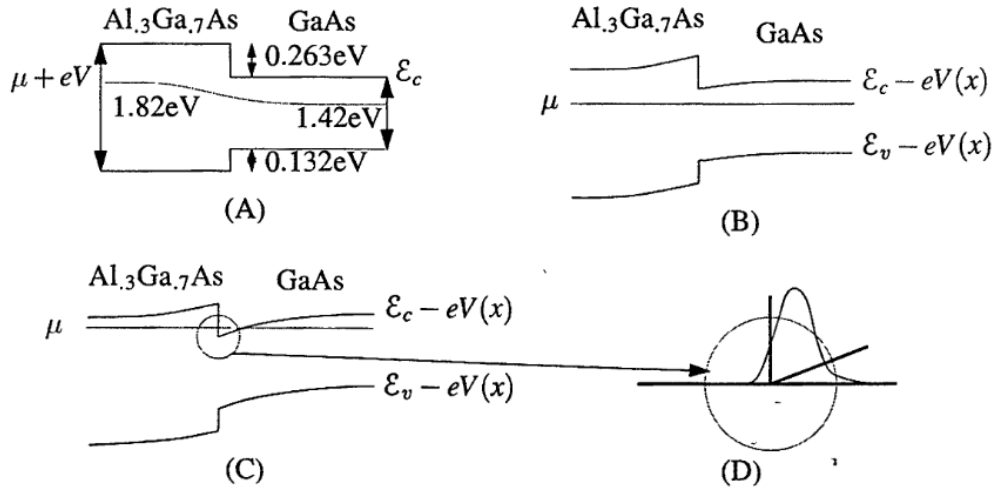

Figure 2.1: Band structure of a heterostructure showing the band bending and abrupt change at the junction in (A) and (B). The chemical potential can be tuned to occupy only a small inversion layer, shown in (C) and (D). Figure from Ref. [76].

With appropriate doping of the semiconductor, the chemical potential can be tuned to produce an inversion layer (see Fig. 2.1C), a small set of states that are bound for sufficiently small temperatures (typically on the order of a Kelvin). Thus the junction of the two alloys of the heterostructure forms a plane of confined electrons known as a two-dimensional electron gas (2DEG).

The electrons can be further confined by placing electrostatic gates on the top or bottom of the sample, parallel to the 2DEG. By placing a long, thin gate across the sample with a small opening, known as a quantum point contact, the path of electrons from a source to a drain can be narrowed so much that the resulting conductance through the opening is quantized in units of $2e^2/h$, allowing for very accurate detection of conductance [76].

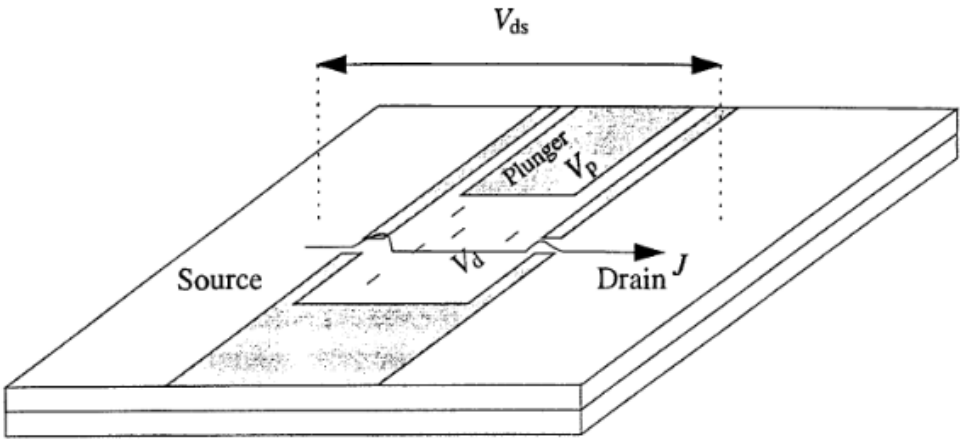

Figure 2.2: Quantum dot confining a small area of a two-dimensional electron gas between two quantum point contacts and a plunger gate. Figure from Ref. [76].

A quantum dot, which is essentially a zero-dimensional quantum well, can be built with two quantum point contacts that confine an area of a 2DEG with a radius on the order of 100nm. Together with the quantum point contacts, another "plunger" gate is used to control the number of electrons in the interior of the dot using Coulomb repulsion, known as the Couloumb blockade in this context. With careful control of the voltages on the plunger and across the two sides of the sample (see Fig. 2.2), it is possible to control and detect the presence of a single electron in the dot [63, 97, 87].

2.1.2 Double Quantum Dot Qubit

With the ability to confine a single electron in a quantum dot, it is theoretically possible to encode a qubit with the spin of that electron. Unfortunately, the spin of the electron would couple to any stray magnetic field in the system and quickly decohere before any meaningful operations could be performed.

However, Levy proposed a fault-tolerant modification of this qubit that is immune to a uniform magnetic field [69]. By using a pair of electrons shared between a

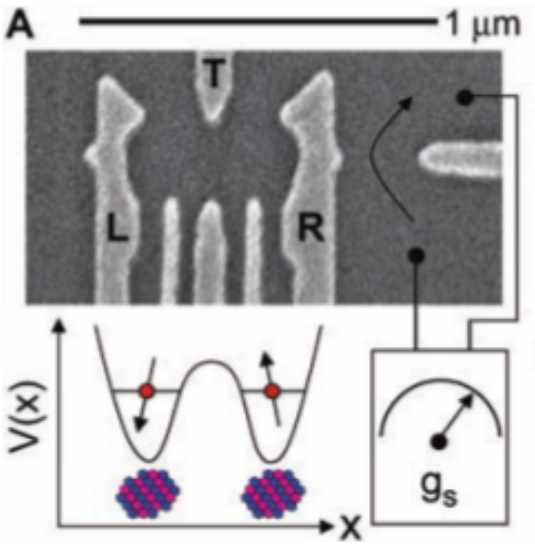

Figure 2.3: Double quantum dots confining a pair of electrons to encode a fault-tolerant qubit immune to a uniform magnetic field formed from the $m_s = 0$ triplet and singlet spin states. From Ref. [87].

double quantum dot (see Fig. 2.3), with the $m_s = 0$ triplet and singlet states as the logical basis states

$$|0\rangle = \frac{|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle}{\sqrt{2}}, \qquad |1\rangle = \frac{|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle}{\sqrt{2}}. \tag{2.1}$$

Since these states have an equal superposition of up and down spin states, a uniform magnetic field has the same effect on the spins of each dot, producing an innocuous overall phase. Nonetheless, each electron is still subject to the local hyperfine interaction from the nuclear spins of the semiconductor lattice of each individual dot, which are generally different for each quantum dot and thus lead to dephasing of the individual electron spins [14].

There have been several proposals to suppress this dephasing such as nuclear polarization [14, 59, 47], state narrowing [62], and spin-echo pulse correction [60, 123]. While improved coherence has been experimentally demonstrated using these techniques, the coherence times desired for applications have proven very difficult to achieve. For

example, pumping methods have been used to partially polarize nuclei, but the nearly full polarization needed has yet to be achieved [62, 52, 113, 97].

Promising results have been shown using spin-echo sequences through the exchange interaction between two spins [87, 5, 117]. The exchange is controlled by lowering the tunneling barrier between the two quantum dots using quickly-controlled electric gates. This leads to Rabi oscillations; a single π-pulse corresponds to exchanging the two spins. Sequences of such pulses effectively couple both spins to the same average hyperfine interaction resulting in improved coherence times. While several echo sequences can be performed using this exchange, it is likely that the electrical gate noise and spatial variations in the Overhauser field remain the dominant sources of dephasing [5].

Rather than relying on interdot tunneling, the current proposal uses a spatial exchange of the two quantum dots, allowing the two electrons to traverse the same path, spending the same time coupled to the local nuclei, as shown in Figure 2.4. Compared to exchange via tunneling, ideally, this should eliminate the effect of the electrical gate noise. On the other hand, in the presence of spin-orbit coupling, the motion of electrons may introduce additional errors. In the remainder of this chapter, we analyze how these errors depend on the parameters of the motion and discuss the constraints and possible parameters for potential implementation of this proposal.

It should be noted that a similar setup with spatial exchange of electrostatically-defined quantum dots has been discussed in relation to holonomic quantum computation [42, 105]. However, the corresponding coherence estimates have been done in the absence of a magnetic field. This proposal focuses on the parameter range characteristic of double-quantum dot qubits and accounts for typical magnetic fields of order $\gtrsim 0.1\,\mathrm{T}$.

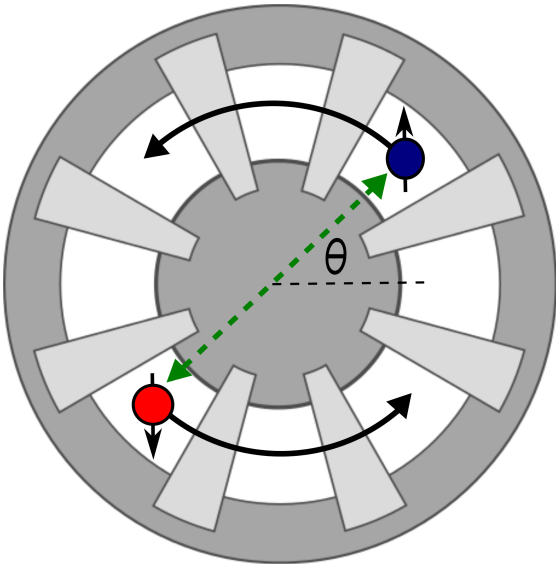

Figure 2.4: Suggested electrode geometry for a rotating double-quantum-dot qubit with top and bottom gates in different shades of gray. Exchange gates via real space rotation, as opposed to tunneling, are expected to strongly reduce the qubit sensitivity to charge noise.

The outline of the remainder of this chapter is as follows. In Sec. 2.2, the Hamiltonian of the double-quantum dot qubit with spatial exchange is defined. The effective spin-only Hamiltonian for a single electron in a moving quantum dot is derived in Sec. 2.3, and the single-qubit fidelity associated with a sequence of double-dot rotations is found in Sec. 2.4. Simulations of a possible protocol are shown in Sec. 2.5, and the constraints and corresponding characteristic time and distance scales are discussed in Sec. 2.6, followed by a summary in Sec. 2.7.

2.2 Double Quantum Dot Setup

Consider a qubit formed by a pair of quantum dots electrostatically defined in a III-V semiconductor (e.g., GaAs/AlGaAs) heterostructure using a system of top and bottom gates similar to that discussed in the previous section and illustrated in Fig. 2.4, with the parameters of the dots similar to the experiments in Refs. [87, 5, 117].

Specifically, each dot contains a single electron, with a typical dot-size quantization energy $\hbar\omega_d \sim 1\text{meV}$. The qubit is defined as the subspace of the singlet and $m_s = 0$ triplet states of the two electrons. The triplet degeneracy is removed by a uniform, constant magnetic field B_0 of at least $\sim 0.1\text{T}$, applied perpendicular to the sample, which creates a Zeeman gap of $\Delta \sim 2.5\ \mu\text{eV}$. The electrons interact with $N \sim 10^6$ spins of the lattice nuclei, leading to a different local hyperfine interaction for each electron. This can be approximated as a Zeeman interaction with a random, fluctuating, non-uniform magnetic "Overhauser" field $B_\text{N} \sim 1\text{mT}$.

To prevent dephasing, both quantum dots will be moved along the same trajectory in a time much shorter than the relaxation time of the nuclear spins, $t_\text{nuc} \sim 100\ \mu\text{s}$, so it is assumed that the Overhauser field is quasi-static. In order to reduce the sensitivity to charge noise the distance between the dots must be significantly greater than the size of each quantum dot, $a \sim 100\text{nm}$. Due to this spatial separation between the dots, the Hamiltonians of each electron can be treated separately,

$$H = H_\text{d}(\mathbf{r}_0) + H_\text{Z} + V_\text{Z}(\mathbf{r}, t) + H_\text{SO} + V(\mathbf{r}, t). \tag{2.2}$$

Here the dot Hamiltonian is given by

$$H_\text{d}(\mathbf{r}_0) = \frac{p^2}{2m} + U(\mathbf{r} - \mathbf{r}_0(t)), \tag{2.3}$$

with the canonical momentum $\mathbf{p} = \mathbf{P} + e\mathbf{A}/c$ and the confining potential U centered at $\mathbf{r}_0 \equiv \mathbf{r}_0(t)$; the Zeeman Hamiltonians for the externally-applied and Overhauser fields are respectively

$$H_\text{Z} = \frac{g\mu_\text{B}}{2}\mathbf{B}_0 \cdot \boldsymbol{\sigma}, \tag{2.4}$$

$$V_\text{Z}(\mathbf{r}, t) = \frac{g\mu_\text{B}}{2}\mathbf{B}_\text{N}(\mathbf{r}, t) \cdot \boldsymbol{\sigma}, \tag{2.5}$$

and the spin-orbit Hamiltonian is given by

$$H_{\mathrm{SO}} = \sigma^i C^{ij} p^j. \tag{2.6}$$

The last term, $V(\mathbf{r}, t)$, accounts for additional effects originating from disorder, variation of the dot potential as it moves due to imperfections of the confining potential, as well as phonons. The spin orbit coupling coefficients C_{ij} in Eq. (2.6) incorporate both Dresselhaus (originating from the lack of the inversion symmetry of the lattice) with $C^{yy} = -C^{xx} = \beta$, and Rashba terms (structural inversion asymmetry due to the quantum well) with $C^{xy} = -C^{yx} = \alpha$. This specific form assumes that the growth of the semiconductor heterostructure and the quantum well asymmetry are in the positive z direction,[108, 121] so all the matrix elements that involve z are zero.

For numerical estimates the effective electron mass used is $m \sim 0.067 m_{\mathrm{e}}$, and the spin-orbit parameter is assumed to satisfy $\alpha \simeq \beta$ with values ranging from 10^3 to 10^4 m/s, as appropriate for typical GaAs heterostructures.[108] Based on the above values, $m\beta^2 \ll g\mu_{\mathrm{B}} B \ll \hbar\omega_{\mathrm{d}}$, so terms quadratic in the spin-orbit coupling can be ignored for the following analysis.

2.3 Effective Single-Dot Hamiltonian

Consider the moving reference frame of the dot using the translation operator

$$\Psi(t) \to e^{-\frac{i}{\hbar}\mathbf{P}\cdot\mathbf{r}_o(t)}\Psi(t), \tag{2.7}$$

$$H \to e^{\frac{i}{\hbar}\mathbf{P}\cdot\mathbf{r}_o(t)} H e^{-\frac{i}{\hbar}\mathbf{P}\cdot\mathbf{r}_o(t)} - \mathbf{v}_0(t)\cdot\mathbf{P}, \tag{2.8}$$

which introduces the additional term proportional to the dot's velocity, $\mathbf{v}_0 \equiv \dot{\mathbf{r}}_0$, and removes the $\mathbf{r}_0$ from the confining potential, $U(\mathbf{r} - \mathbf{r}_0) \to U(\mathbf{r})$. This also affects the vector potential, $\mathbf{A}(\mathbf{r}) \to \mathbf{A}(\mathbf{r} + \mathbf{r}_0)$, in the dot Hamiltonian and spin-orbit term, which

can be reversed with an appropriate gauge transformation. Using the symmetric gauge, $\mathbf{A}(\mathbf{r}+\mathbf{r}_0) = \frac{1}{2}\mathbf{B}\times(\mathbf{r}+\mathbf{r}_0)$, and transforming $\mathbf{A} \to \mathbf{A} + \nabla f$ with $f = -\frac{1}{2}\mathbf{r}\cdot(\mathbf{B}\times\mathbf{r}_0)$ results in

$$\Psi \to \Psi \exp(-\frac{ie}{\hbar c}f), \tag{2.9}$$

$$\mathbf{P} \to \mathbf{P} - \frac{e}{c}\nabla f. \tag{2.10}$$

These transformations introduce two additional terms in the time-dependent Schrödinger equation $i\hbar\partial\Psi/\partial t = H\Psi$. The first term, $-(e/2c)\mathbf{v}_0\cdot(\mathbf{B}\times\mathbf{r}_0)$, arises when Eq. (2.10) is substituted into the $-\mathbf{v}_0(t)\cdot\mathbf{P}$ term in Eq. (2.8). The second term, $-(e/2c)\mathbf{r}\cdot(\mathbf{B}\times\mathbf{v}_0)$, appears on the left hand side as a result of taking the time derivative of the exponent in Eq. (2.9). These two terms can be moved to one side of the equation and combined using the cyclic property of the mixed product:

$$-\frac{e}{2c}\mathbf{v}_0\cdot(\mathbf{B}\times\mathbf{r}_0) + \frac{e}{2c}\mathbf{r}\cdot(\mathbf{B}\times\mathbf{v}_0) = -\frac{e}{2c}\mathbf{v}_0\cdot\mathbf{B}\times(\mathbf{r}+\mathbf{r}_0) \tag{2.11}$$

which is precisely the vector potential term in $-\mathbf{v}_0\cdot\mathbf{p}$. This leads to the moving-frame Hamiltonian

$$H = H_{\mathrm{d}} + H_{\mathrm{Z}} + H_{\mathrm{SO}} + V\big(\mathrm{r}+\mathbf{r}_0(t),t\big) + V_{\mathrm{Z}}\big(\mathrm{r}+\mathbf{r}_0(t),t\big) - \mathbf{v}_0\cdot\mathbf{p}. \tag{2.12}$$

Following Golovach et al.[43], a canonical transformation is performed, $H \to e^S H e^{-S} \simeq (1+S)H(1-S) = H + [S,H]$, where S is anti-Hermitian and chosen to eliminate the original spin-orbit term. Splitting $S = S_0 + S_1$ such that

$$[S_0, H_{\mathrm{d}}] + H_{\mathrm{SO}} = 0, \tag{2.13}$$

$$[S_1, H_{\mathrm{d}}] + [S_0, H_{\mathrm{Z}}] = 0, \tag{2.14}$$

and choosing

$$S_0 = \frac{im}{\hbar}\sigma^i C^{ij} r^j, \tag{2.15}$$

satisfies Eq. (2.13). This can be verified, noting that S_0 has no momentum dependence so it clearly commutes with the confining potential, and

$$\begin{aligned}\left[\frac{im}{\hbar}\sigma^i C^{ij} r^j, \frac{p^2}{2m}\right] &= \frac{i}{\hbar}\sigma^i C^{ij}[r^j, p^k]p^k \\ &= -\sigma^i C^{ij} p^j. \end{aligned} \tag{2.16}$$

With S_0 known, Eq. (2.14) is used to define S_1,

$$\begin{aligned}[H_d, S_1] &= [S_0, H_{\rm Z}] = \frac{img\mu_{\rm B}}{2\hbar} C^{ij} r^j [\sigma^i, \sigma^k] B_0^k \\ &= -\frac{mg\mu_{\rm B}}{\hbar} C^{ij} r^j \epsilon^{ikl} B_0^k \sigma^l \\ &= -\frac{g\mu_{\rm B}}{2}\mathbf{Q}\cdot(\mathbf{B}_0 \times \boldsymbol{\sigma}) \\ &= \frac{g\mu_{\rm B}}{2}(\mathbf{B}_0 \times \mathbf{Q})\cdot\boldsymbol{\sigma}, \end{aligned} \tag{2.17}$$

where $Q^i \equiv (2m/\hbar)C^{ij}r^j$. This equation can be written in terms of the electron's orbital states $|n\rangle$ in the dot potential,

$$\begin{aligned}\langle n|[H_d, S_1]|m\rangle &= (S_1)_{nm}(E_n - E_m) \\ &= \frac{g\mu_{\rm B}}{2}\boldsymbol{\sigma}\cdot(\mathbf{B}_0 \times \langle\mathbf{Q}\rangle_{nm}). \end{aligned} \tag{2.18}$$

As long as the relevant dot quantization energies are non-degenerate, $E_n \neq E_m$, this can be written as

$$\begin{aligned}(S_1)_{nm} &= \frac{g\mu_{\rm B}}{2}\boldsymbol{\sigma}\cdot\frac{(\mathbf{B}_0 \times \langle\mathbf{Q}\rangle_{nm})}{E_n - E_m} \\ &= \frac{g\mu_{\rm B}}{2}\boldsymbol{\sigma}\cdot\mathbf{W}_{nm}, \end{aligned} \tag{2.19}$$

where W_{nm} is defined as

$$\mathbf{W}_{nm} \equiv \frac{\mathbf{B}_0 \times \langle\mathbf{Q}\rangle_{nm}}{E_n - E_m}. \tag{2.20}$$

Expanding the canonical transformation to first order in the spin-orbit parameter contained in S_0 and S_1, the transformed Hamiltonian is

$$\bar{H} \simeq H + [S_0, V_{\rm Z}] - [S_0, \mathbf{v}_0\cdot\mathbf{p}] + [S_1, V] + [S_1, H_{\rm Z}] - [S_1, \mathbf{v}_0\cdot\mathbf{p}] + [S_1, V_{\rm Z}] \tag{2.21}$$

Using the definition of S_0, the first two commutators are

$$\frac{img\mu_{\rm B}}{2\hbar}C^{ij}r^j[\sigma^i,\sigma^k]B^k_{\rm N} = -\frac{mg\mu_{\rm B}}{\hbar}C^{ij}r^j(\epsilon^{ikl}B^k_{\rm N}\sigma^l)$$
$$= \frac{g\mu_{\rm B}}{2}(\mathbf{B}_{\rm N}\times\mathbf{Q})\cdot\boldsymbol{\sigma} \tag{2.22}$$

and

$$-\frac{im}{\hbar}\sigma^i C^{ij}[r^j,p^k]v^k_0 = m\sigma^i C^{ij}v^j_0 = \frac{1}{2}\mathbf{Q}_{\rm v}\cdot\boldsymbol{\sigma}, \tag{2.23}$$

respectively, with $Q^i_{\rm v}\equiv 2mC^{ij}v^j_0$ defined analogously to Q^i. One can now define the effective spin Hamiltonian by projecting onto the orbital ground state, $H_S\equiv\langle 0|\bar{H}|0\rangle$. This makes it possible to express the remaining commutators in the transformed Hamiltonian using the definition of S_1. The first commutator involving S_1 in Eq. (2.21) is simplified by explicitly writing out the commutator and inserting a complete set of states,

$$\begin{aligned}\langle 0|S_1V|0\rangle - \langle 0|VS_1|0\rangle &= \sum_{n>0}\langle 0|S_1|n\rangle\langle n|V|0\rangle - \langle 0|V|n\rangle\langle n|S_1|0\rangle \\ &= \sum_{n>0}(S_1)_{0n}V_{n0} - V_{0n}(S_1)_{n0}.\end{aligned} \tag{2.24}$$

Assuming V has no momentum dependence, it commutes with S_1 and the order of the second term above can be reserved. Since S_1 is anti-Hermitian, while V is Hermitian,

$$\begin{aligned}\langle 0|S_1V|0\rangle - \langle 0|VS_1|0\rangle &= 2\sum_{n>0}(S_1)_{0n}V_{n0} \\ &= g\mu_{\rm B}\boldsymbol{\sigma}\cdot\sum_{n>0}\mathbf{W}_{0n}V_{n0}.\end{aligned} \tag{2.25}$$

This technique can be used on the remaining terms in Eq. (2.21). The second term involving S_1 in Eq. (2.21) vanishes because

$$(H_{\rm Z})_{n0} = \frac{g\mu_{\rm B}}{2}\boldsymbol{\sigma}\cdot\langle n|\mathbf{B}_0|0\rangle = 0. \tag{2.26}$$

However, the final commutator in Eq. (2.21) contains the terms

$$(S_1)_{0n}(V_{\rm Z})_{n0} - (V_{\rm Z})_{0n}(S_1)_{n0}, \tag{2.27}$$

which do not commute because they contain two spin terms, but can be treated using the spin identity

$$(\boldsymbol{\sigma}\cdot\mathbf{a})(\boldsymbol{\sigma}\cdot\mathbf{b}) = \mathbf{a}\cdot\mathbf{b} + i\boldsymbol{\sigma}\cdot(\mathbf{a}\times\mathbf{b}). \tag{2.28}$$

The first term in Eq. (2.27) becomes

$$\left(\frac{g\mu_{\mathrm{B}}}{2}\right)^2 \left\{\mathbf{W}_{0n}\cdot(\mathbf{B}_{\mathrm{N}})_{n0} + i\boldsymbol{\sigma}\cdot[\mathbf{W}_{0n}\times(\mathbf{B}_{\mathrm{N}})_{n0}]\right\}, \tag{2.29}$$

and the second term looks quite similar, except the anti-commutator of the cross product causes the spin dependent term to cancel with the one above, while the spin-indepedent terms is doubled. The Hamiltonian contains several of these spin-independent terms that can be taken as constants. The effective spin Hamiltonian, up to a constant, can now be written simply as

$$H_S = \frac{1}{2}\hbar\left[\boldsymbol{\omega}_0 + \boldsymbol{\omega}_1(t)\right]\cdot\boldsymbol{\sigma}, \tag{2.30}$$

where

$$\boldsymbol{\omega}_0 = \frac{g\mu_{\mathrm{B}}}{\hbar}\mathbf{B}_0, \tag{2.31}$$

is the Larmor frequency, and the time-dependent term $\omega_1 \equiv \omega_1(t)$ is

$$\begin{aligned}\boldsymbol{\omega}_1 = {} & \frac{g\mu_{\mathrm{B}}}{\hbar}\left[\mathbf{B}_{\mathrm{N}} + (\mathbf{B}_{\mathrm{N}})_{0n}\times(\mathbf{Q})_{n0}\right] + \frac{4g\mu_{\mathrm{B}}}{\hbar}\mathbf{W}_{0n}V_{n0} \\ & + \frac{1}{2\hbar}\mathbf{Q}_{\mathrm{v}} - \frac{2g\mu_{\mathrm{B}}}{\hbar}\mathbf{W}_{0n}(\mathbf{v}_0\cdot\mathbf{p}_{n0}),\end{aligned} \tag{2.32}$$

where the index n is implicitly summed over all the excited states of the dot. If the phonons are ignored and the Overhauser fields are approximated as static, the time dependence of the terms in the first line of Eq. (2.32) comes only from the position, parameterized by the known trajectory of the dot, $\mathbf{r}_0(t)$. Similarly, the time-dependence of the terms in the second line comes from both the position $\mathbf{r}_0(t)$ and the dot velocity $\mathbf{v}_0(t)$, [in fact, these terms are all linear in components of $\mathbf{v}_0(t)$]. This simple spin

Hamiltonian is the key result of this derivation; it is correct to linear order in the spin-orbit couplings. It should be noted that including cubic spin-orbit terms in the original Hamiltonian introduces additional terms proportional to $\mathbf{v}_0^2(t)$ and $\mathbf{v}_0^3(t)$ in Eq. (2.32), but these terms are smaller by at least an order of magnitude [64] and the general spin Hamiltonian form in Eq. (2.30) is preserved.

2.4 Average Fidelity

2.4.1 General Expression

In order to analyze the implications of the additional terms in the effective spin Hamiltonian (Eq. 2.30), one needs to take into account that the qubit is actually formed by two electron spins. It will be convenient to assume that the dots' velocities are small compared to the speed of sound, $v_0 \ll s \sim 5 \times 10^3\,\mathrm{m/s}$, meaning the phonon effects should decouple from the dots' motion and can be approximated as contributing to the same "intrinsic" decoherence times as one would have without the motion. The effect of such decoherence terms on dynamical decoupling has been considered in detail in Ref. [91]; in the following we assume that these decoherence times are large compared to the characteristic period T of the dots' motion and therefore can be ignored.

In the absence of phonons, and approximating the Overhauser field as classical, the time evolution of the two-spin wavefunction with $N = 4$ components can be characterized by a unitary matrix $U(t)$. The qubit subspace $\mathcal{Q}$ is formed by the $m_z = 0$ component of the two-spin wavefunction; it has $M = 2$ dimensions. The standard assumption is that, at the beginning of the experiment, the two spins are initiated in a pure state $|\psi\rangle$ which belongs to the qubit subspace, $|\psi\rangle \in \mathcal{Q}$. Therefore, when com-

puting the average fidelity, one only needs to average over the original wavefunctions in $\mathcal{Q}$.

More generally, consider an M-dimensional subspace $\mathcal{Q}$ of an N-dimensional Hilbert space $\mathcal{H}$. Introducing an $N \times M$ matrix T whose columns are formed by the components of orthonormal vectors that form a basis of $\mathcal{Q}$, then the components of an arbitrary wavefunction $|\psi\rangle \in \mathcal{Q}$ are a linear combination of the columns of T; namely $\psi = T\varphi$, where φ is an M-dimensional column vector, $\|\varphi\| = 1$. The corresponding density matrix can be written in this basis as $\rho_0 \equiv T\varphi\,\varphi^\dagger T^\dagger$. The fidelity corresponding to the evolution matrix U is

$$F = \mathrm{Tr}(\rho_0 U \rho_0 U^\dagger) = (\varphi^\dagger W \varphi)(\varphi^\dagger W^\dagger \varphi), \tag{2.33}$$

where $W = T^\dagger U T$ can be thought of as the projection of U onto the subspace $\mathcal{Q}$. The average fidelity in the subspace can now be calculated using the averaging identities for components φ_i of the normalized wavefunction $|\varphi\rangle$

$$\langle \varphi_i \varphi_j^* \rangle = \delta_{ij}/M, \tag{2.34}$$

$$\langle \varphi_i \varphi_j^* \varphi_k \varphi_l^* \rangle = \frac{\delta_{ij}\delta_{kl} + \delta_{il}\delta_{jk}}{M^2 + M}. \tag{2.35}$$

This leads to the average fidelity

$$\langle F \rangle = \frac{|\,\mathrm{Tr}\, W|^2 + \mathrm{Tr}(WW^\dagger)}{M^2 + M}. \tag{2.36}$$

For the special case of the qubit formed by the singlet and $m = 0$ triplet states of two spins, assuming no interdot tunneling, the net evolution matrix is just the Kronecker product of evolution matrices corresponding to the two qubits, $U = U_1 \otimes U_2$. Further, it will be convenient to decompose each single-spin matrix in the interaction representation with respect to the precession in the net effective magnetic field along

the z-axis,

$$U_i = U_{0i} S_i, \quad U_{0i} \equiv e^{-i\sigma^z \varphi_i(t)/2}, \tag{2.37}$$

where

$$\varphi_i(t) \equiv \omega_0 t + \int_0^t dt' \, \omega_{1i}^z(t'), \tag{2.38}$$

ω_0 is the Larmor frequency, $\omega_{1i}^z(t)$, with $i = 1, 2$ [cf. Eq. (2.30)], are the two dot's effective perturbing fields in the z-direction, and the matrices

$$S_i \equiv e^{-i\gamma_i - i\boldsymbol{\phi}_i \cdot \boldsymbol{\sigma}/2}, \;\; i = 1, 2, \tag{2.39}$$

are parametrized as rotations by angle $\phi_i \equiv |\boldsymbol{\phi}_i|$ around the unit vectors $\hat{\boldsymbol{\phi}}_i$, with extra phases γ_i. These rotations come entirely from transverse, $\mu = x, y$, components of $\boldsymbol{\omega}_{1i}^{\mu}$ in the rotating frame, largely due to the Larmor frequency. Since the Larmor frequency is large, the additional rotation angles are expected to be small; we expand the average fidelity (2.36) to quadratic order in components of $\boldsymbol{\phi}_i$,

$$\langle F \rangle \;\; = \;\; 1 - f_0 - f_1 - f_2^z - f_2^{\perp} + \ldots, \tag{2.40}$$

with the infidelity terms

$$f_0 \;\; = \;\; \frac{2}{3} \sin^2(\Delta\varphi/2), \tag{2.41}$$

$$f_1 \;\; = \;\; \frac{1}{3} (\phi_2^z - \phi_1^z) \sin(\Delta\varphi), \tag{2.42}$$

$$f_2^z \;\; = \;\; \frac{1}{6} (\phi_2^z - \phi_1^z)^2 \cos(\Delta\varphi), \tag{2.43}$$

$$f_2^{\perp} \;\; = \;\; \frac{2 + \cos(\Delta\varphi)}{12} \left[(\phi_1^{\perp})^2 + (\phi_2^{\perp})^2 \right]. \tag{2.44}$$

Note that, as expected, the fidelity only depends on the differences $\Delta\varphi \equiv \varphi_2 - \varphi_1$ and $\phi_2^z - \phi_1^z$ of the two precession angles around the z-axis. To the same quadratic accuracy in the small angles, one can also write

$$f_0 + f_1 + f_2^z = \frac{1}{3} [1 - \cos(\Delta\varphi + \phi_2^z - \phi_1^z)], \tag{2.45}$$

which only depends on the total phase difference, and is exact in the limiting case when $\phi_i^{\perp} = 0$.

2.4.2 Rotating-Frame Approximation

Returning to the analysis of the single-spin Hamiltonian (Eq. 2.30), the Larmor frequency, $\omega_0 \gtrsim 4 \times 10^9\,\mathrm{rad/s}$, is the dominant term, $\omega_1 \ll \omega_0$. Given that the phonons have been excluded, one can also assume that the dot trajectory is such that the time dependence in $\omega_1(t)$ is slow on the scale of ω_0. This implies that the interaction representation Eq. (2.37) is valid, where $S(t)$ is the slow part of the unitary evolution operator; it obeys the equation

$$i\hbar\dot{S} = H_{\mathrm{int}}(t)S, \quad S(0) = \mathbb{1}, \tag{2.46}$$

where the dot index is temporarily omitted, and

$$\begin{aligned} \frac{1}{\hbar}H_{\mathrm{int}}(t) &\equiv \frac{1}{2}\boldsymbol{\omega}_1(t)\cdot U_0^{\dagger}(t)\boldsymbol{\sigma}U_0(t) && (2.47)\\ &= \frac{1}{2}\Big\{\sigma^x[\omega_1^x(t)\cos\varphi(t) + \omega_1^y(t)\sin\varphi(t)] \\ &\qquad + \sigma^y[\omega_1^y(t)\cos\varphi(t) - \omega_1^x(t)\sin\varphi(t)]\Big\} && (2.48)\end{aligned}$$

is the perturbing Hamiltonian in the interaction representation.

The remaining analysis is performed perturbatively with the Magnus (cumulant) expansion,

$$S(t) = \exp(\mathcal{C}^{(1)} + \mathcal{C}^{(2)} + \ldots), \tag{2.49}$$

where the first two cumulants are

$$\begin{aligned} \mathcal{C}^{(1)}(t) &= -\frac{i}{\hbar}\int_0^t dt_1 H_{\mathrm{int}}(t_1), && (2.50)\\ \mathcal{C}^{(2)}(t) &= -\frac{1}{\hbar^2}\int_{0<t_1<t_2<t} dt_1\,dt_2\,[H_{\mathrm{int}}(t_2), H_{\mathrm{int}}(t_1)]. && (2.51)\end{aligned}$$

The integration is performed explicitly to leading order in ω_1/ω_0, also assuming that the time-dependence of ω_1 is slow on the scale of ω_0:

$$\begin{aligned} i\mathcal{C}^{(1)}(t) &= \frac{1}{2\omega_0}\Big\{\sigma^x[a\sin\varphi(t)+b_0-b\cos\varphi(t)] \\ &\quad + \sigma^y[b\sin\varphi(t)-a_0+a\cos\varphi(t)]\Big\}, \qquad (2.52) \\ i\mathcal{C}^{(2)}(t) &= \frac{1}{4\omega_0}\sigma^z\int_0^t dt_1\,[a^2(t_1)+b^2(t_1)], \qquad (2.53) \end{aligned}$$

denoting $a \equiv \omega_1^x(t)$, $b \equiv \omega_1^y(t)$, and a_0, b_0 as the corresponding values at $t=0$ [$\varphi(0)\equiv 0$ by definition]. Eq. 2.40 gives the expression for the average qubit infidelity

$$\begin{aligned} 1-\langle F\rangle &= f_z + f_1^\perp + f_2^\perp, \qquad (2.54) \\ f^z &= \frac{1}{3}[1-\cos(\Delta\varphi+\delta_2-\delta_1)], \qquad (2.55) \\ f_i^\perp &= \frac{2+\cos(\Delta\varphi)}{12\omega_0^2}\left[(B_i-b_{0i})^2+(A_i-a_{0i})^2\right], \qquad (2.56) \end{aligned}$$

where the index $i=1,2$ refers to the two spins, $A_i \equiv a_i\cos\varphi(t)+b_i\sin\varphi(t)$ and $B_i \equiv b_i\cos\varphi(t)-a_i\sin\varphi(t)$ are the rotated components of the transverse angular-velocity vectors, (ω_1^x,ω_1^y), for the corresponding spins, and the additional phases δ_i are given by the integrals.

$$\delta_i \equiv \int_0^t \frac{dt'}{2\omega_0}\,[\omega_{1i}^\perp(t')]^2. \qquad (2.57)$$

One immediately recognizes the additional phases in Eqs. (2.55) and (2.57) as the effect of level repulsion, or equivalently as the gap for the spins, driven adiabatically by the total instantaneous magnetic field $\propto[(\omega_0+\omega_{1i}^z)^2+(\omega_{1i}^\perp)^2]^{1/2}$. Then, Eq. (2.56) can be interpreted as the effect of the basis change between the original quantization z-axis and the direction of the instantaneous magnetic field; it is similar in nature to the initial decoherence associated with any periodic decoupling sequence, see, e.g. Ref. [92].

The terms of the next order in $1/\omega_0$ expansion, omitted in Eqs. (2.52) and (2.53), include the trivial correction to Eq. (2.53) $\propto\int dt\,\omega_1^z[\omega_1^\perp]^2/\omega_0^2$, as well as the ge-

ometrical phase $\propto \int dt' W[\omega_1^x, \omega_1^y]$, where $W[x,y] \equiv W[x(t), y(t)] \equiv x(t)y'(t) - x'(t)y(t)$ is the Wronskian.

Since $\Delta\varphi = \int_0^t dt' \left[\omega_{12}^z(t') - \omega_{11}^z(t')\right]$, the term f^z does not contain any rapid oscillations at the Larmor precession frequency ω_0, while the terms $f_i^\perp$ can be averaged over the period of Larmor precession by replacing $A_i^2 + B_i^2$ with $(\omega_{1i}^\perp)^2$ and dropping all of the terms linear in A_i, B_i,

$$\overline{f}_i^\perp = \frac{2 + \cos(\phi_2^z - \phi_1^z)}{3\omega_0^2} \left\{ [\omega_{1i}^\perp(t)]^2 + [\omega_{1i}^\perp(0)]^2 \right\}. \tag{2.58}$$

2.4.3 Sequences

Overall, the dynamical decoupling should be designed to null the difference between the accumulated phases $\varphi_i + \delta_i$ of the two spins which suppresses the main contribution to the infidelity, see Eq. (2.55). For a static Overhauser field, this can be achieved just by ensuring that each spin spends the same amount of time at each position, e.g., via the solid adiabatic trajectory in Fig. [2.5]. This is also sufficient to suppress the effect of the velocity-dependent terms in the second line of Eq. (2.32). A more complicated set of dot rotation involving motion in both directions, e.g., see the dashed trajectory of Fig. [2.5], are required to suppress a time-dependent Overhauser field.

To analyze the effect of a sequence of π-rotations in the presence of a time and position-dependent Overhauser field, consider its Fourier expansion at a position on the trajectory parametrized by the rotation angle θ,

$$B^z(\theta, t) = A_0(t) + \sum_m A_m(t) \cos m\theta + B_m(t) \sin m\theta. \tag{2.59}$$

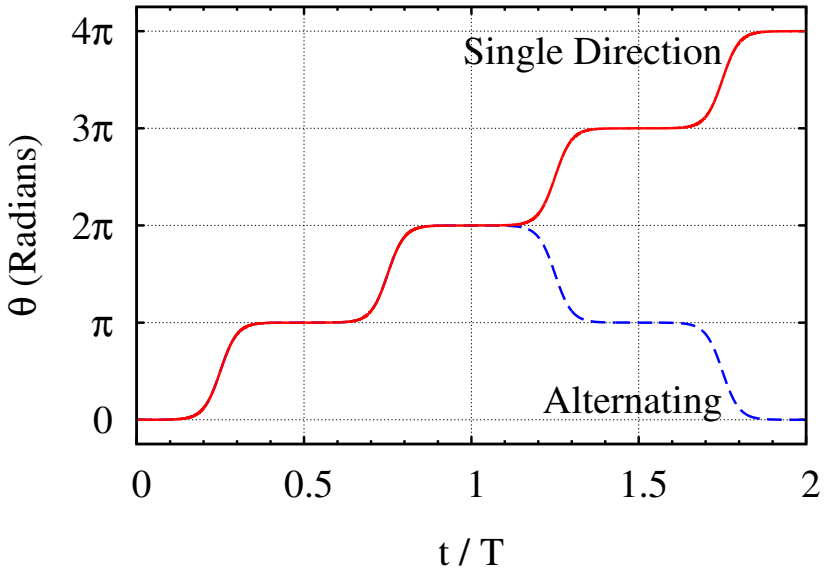

Figure 2.5: Time dependent adiabatic trajectories used in the simulations. Plotted is the position of the first dot parametrized in terms of the angle θ as a function of time t. The angle-dependent positions were defined as a sum of properly scaled and shifted hyperbolic tangents. Single direction rotations suppress the effect of a static Overhauser field but not of a time-varying one. Longer rotation sequences like alternating forward-and-back suppress the effect of a linear in time Overhauser field. Arbitary rotations on the Bloch sphere may be accomplished using additional operations when the dot is stationary, corresponding to the flat segments in the sequence.

Only the difference between the fields corresponding to the two dots (located at θ and $\theta + \pi$) is relevant for the infidelity Eq. (2.55). This leaves only the terms with m odd in the Fourier expansion (2.59).

For a term with $\cos m\theta$ (an even function of θ), a sequence of π rotations acts the same way, independent of the direction. It is easy to check that with an equidistant sequence of rotations centered at $T_0/2$, $3T_0/2$, ..., $(2s+1)T_0/2$, where the number of rotations s is even; any time-independent and linear in t contributions to $A_m(t)$ are suppressed, but a quadratic term would generally remain. Unlike the usual dynamical decoupling problem[60, 116], it is not generally possible to suppress the quadratic term of $A_m(t)$.

The rotation direction starts to matter for a term with $\sin m\theta$ which is an odd function of θ. Here a sequence of π rotations in the same direction picks up a sum of contributions from consecutive time intervals with alternating signs, suppressing the time-independent contribution to $B_m(t)$ but not the linear contribution. As an alternative prescription, a symmetrized sequence of two forward rotations by angle π, followed by two rotations in the opposite direction can be used to suppress the effect of the linear in t term in $B_m(t)$. Generally, it is possible to design more complicated sequences analogous to concatenated or Uhrig's sequences to suppress the effect of any fixed-degree polynomial in time $B_m(t)$[60, 116]. However, this is not expected to be useful since the quadratic time contribution of $A_m(t)$ would still remain.

2.5 Simulations

The above analytical results were corroborated by simulating the two-spin unitary evolution with the effective Hamiltonian (Eq. 2.30). Specifically, the dot trajectory

was parametrized by the rotation angle $\theta = \theta_1(t)$; the other dot is assumed to have the symmetric position, $\theta_2(t) = \theta_1(t) + \pi$ [see Fig. 2.5 for samples of actual trajectories.] The position-dependent terms in the first line of Eq. 2.32 were simulated in terms of a three-component correlated magnetic field $\mathbf{B}(\theta)$ drawn from the Gaussian distribution with zero average and the correlation function $\langle B_\mu(\theta) B_\nu(\theta')\rangle = \sigma_\mu^2 \delta_{\mu\nu} \vartheta(\theta - \theta')$ (no implicit summation in $\mu, \nu = x, y, z$), where

$$\vartheta(\theta) \equiv \sum_{m=-\infty}^{\infty} e^{-(\theta - 2\pi m)^2/\ell^2}$$

is an infinite sum of Gaussian functions (which can also be represented in terms of the Jacobi theta function). These were obtained by applying a Gaussian filter to a discrete set of uncorrelated random numbers drawn from the Gaussian distribution, and using the standard cubic spline interpolation with the result. To simulate the components of the time-dependent magnetic field $B(\mathbf{r}, t)$, explicit order-r spline interpolation are used between several such angle-dependent functions, where $r = 1, 2$.

For all simulations, the time units were chosen to correspond to the Larmor precession period, $\tau_0 \equiv 2\pi/\omega_0$, and the correlation time of the Overhauser field, $4 \cdot 10^4 \tau_0$, with each component of its r.m.s. value corresponding to rotation frequency, $\langle |\boldsymbol{\omega}_1|^2 \rangle = 0.025/\tau_0^2$. The adiabatic trajectories of the dots were simulated using a sum of appropriately shifted hyperbolic tangents, scaled so that the dot is in motion for approximately half of the protocol. The leading velocity-dependent term in Eq. 2.32 was simulated using the corresponding derivatives and the parameter $Q_v/\hbar = 0.075/\tau_0$, assuming equal contributions from the Rashba and Dresselhaus parameters.

For the case of the static Overhauser field (see Fig. 2.6) the average infidelity is dominated by the contribution from the z-component of the field, and the

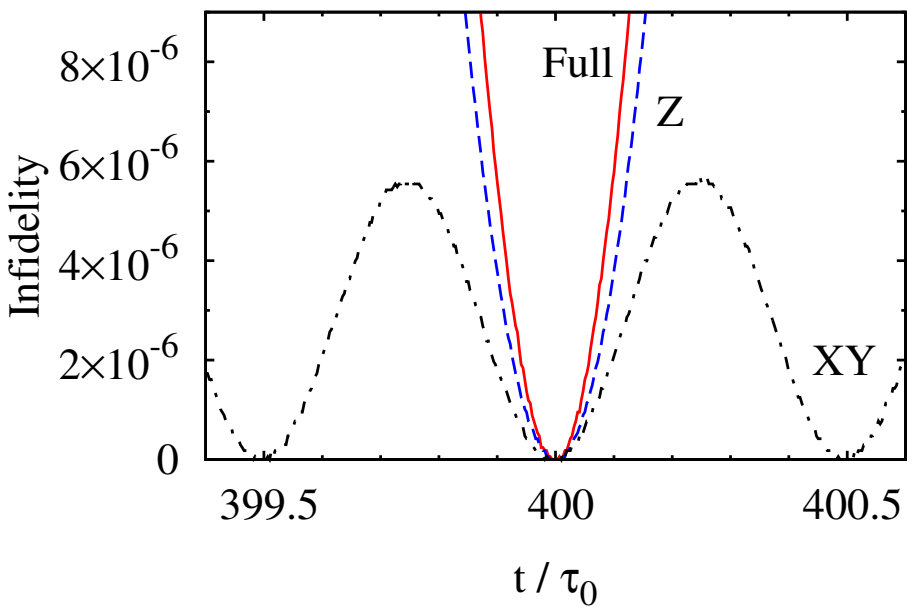

Figure 2.6: Simulated qubit infidelity $1 - \langle F \rangle$ (Eq. 2.36) in the vicinity of the first full rotation period of the double-dot qubit at $t = T$. Position-dependent magnetic field $B_\mu(\theta)$ is assumed static, and the rotation period T is chosen commensurate with the Larmor frequency, $\omega_0 T/2\pi = 400$. Dashed line: only $B_z(\theta)$ is included; the infidelity minimum is exactly at $t = T$, in agreement with Eq. 2.56 which is exact in this situation. Dotted line: only the transverse components $B_\mu(\theta)$, $\mu = x, y$ are included. The infidelity minimum is slightly off the commensurate time $t = T$ due to the terms not included in Eq. 2.54. All three components of the field $B_\mu(\theta)$ are included for the red solid line.

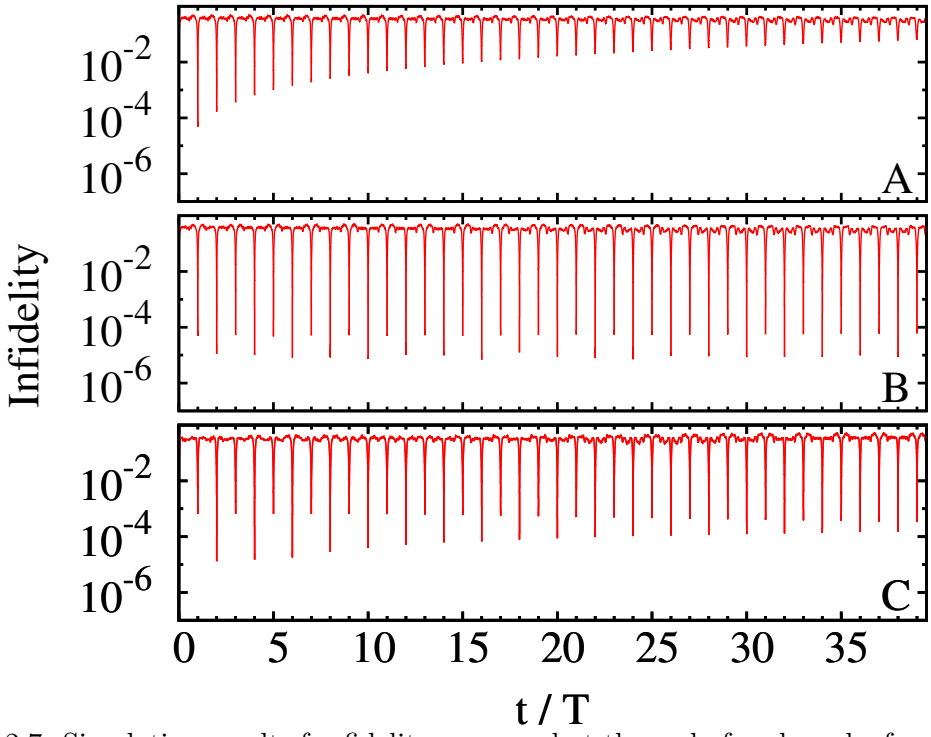

Figure 2.7: Simulation results for fidelity measured at the end of each cycle, for a linear time-dependence of B_{1i}^{μ} with (a) forward, (b) forward-back dot rotations (cf. Fig. 2.5), as well as (c) forward-back for quadratic time-dependence.

infidelity nearly vanishes at the end of the spatial exchange protocol. For a linearly time-interpolated Overhauser field, the infidelity increases over several cycles of the single-direction π pulses(see Fig. 2.7a), but maintains a low value $\sim 10^{-5}$ after alternating between a sequence of two forward rotations, followed by two rotations in the opposite direction (see Fig. 2.7b). However, for the quadratic interpolation (see Fig. 2.7c) the infidelity gradually increases even for the alternating protocol, though it stays below 10^{-4} for several cycles. The code for the above simulation is provided in Appendix A.

2.6 Possible Experimental Setup

Aspects of this proposal, such as the precise construction of few-electron quantum dots in III-V semiconductor heterostructures, have already been demonstrated in experiments [87, 97, 117, 65]. However, the precise adiabatic rotation required in our

proposal may be quite difficult to accomplish experimentally. This section discusses possible design implementations for an experimental realization, as well as physical constraints.

As discussed above, our proposal is suitable in materials with relatively weak spin-orbit coupling such as GaAs/AlGaAs heterostructures. It should be possible for the electrons in a 2DEG to be confined in the radial direction by creating a circular depletion layer by placing electrostatic gates with a constant voltage in the center and outer edge of the circle as sketched in Figure 2.4. Since the tunneling should be suppressed, the normal interdot spacing should be much larger than the dot size, a, so the circular trajectory can have a radius $r_0 \sim 15a \sim 1$ μm. Confinement and rotation in the angular direction could be accomplished by placing appropriately chosen time-dependent voltages on the "wedge-gates" on both sides of the electron (Fig. 2.4). Several of these wedges will be needed to accomplish the smooth and adiabatic trajectory needed. The combined use of wedge and circular gates may require gates on both the top and bottom of the sample. The typical confining potential is approximated by $U \sim m\omega_{\mathrm{d}}^2 a^2$, which only requires reasonable gate voltages on the order of 100 mV.

The averaging of the hyperfine interaction is only valid in the quasi-static approximation of the Overhauser field. In general, the hyperfine interaction between the electron and the nuclei leads to a Knight shift. However, in the presence of the magnetic field, fluctuating corrections to the quasi-static approximation are inversely proportional to B_0 and can be neglected [114]. Thus, the most significant effect in this case is due to the dipole-dipole interaction of nearest neighboring nuclei which requires that $T \ll t_{\mathrm{nuc}} \sim 10^{-4}$s [77]. This places a lower bound on the velocity, while there is also an upper bound necessary to ensure that the Lorentz force from the dot rotation only deforms the actual path of the electron by a distance much smaller than the correlation

length of the Overhauser field. This results in the restriction, $10^{-1} \ll v \ll 10^5$ in m/s. This rather lenient constraint is due to the assumption that the confinement potential be large compared to the other potentials in our system. This allows one to neglect trajectory deviations from perturbations such as charge noise. In these estimates, v is assumed to be small compared to the speed of sound, $s \sim 5 \times 10^3$m/s. With $a/r_0 \sim 15$ and $v \sim 10\,\mathrm{m/s}$, a rotation period of $T \sim 1\ \mu$s, which easily satisfies the above conditions and, according to the simulations, should result in the infidelities lower than 10^{-4} for significant time-scales. The rotation period could potentially be decreased to allow more operations to be performed before the states decohere.

2.7 Quantum Dot Proposal Summary

The real-space exchange of quantum dots was analyzed as a possible substitute for the tunneling exchange. Ideally, exchange eliminates the hyperfine dephasing from the Overhauser field parallel to the applied field, leaving only the smaller effects from the in-plane field. The real-space exchange accomplishes the same suppression of the hyperfine interaction, but avoids the problematic sensitivity to charge noise present in exchange via tunneling. While spatial exchange does introduce additional effects such as spin-orbit coupling, simple tricks like using pairs of π rotations in alternating directions can be used to suppress these so that the decoherence is still dominated by the hyperfine interaction. In particular, this field only enters as the small ratio of the average in-plane Overhauser field to the externally applied field. Perhaps the simplest way to suppress the hyperfine interaction in this approach is to reduce this ratio by increasing the externally applied field.

In addition, this spatial exchange is compatible with some of the methods already being attempted such as nuclear polarization via pumping. If the hyperfine interaction can be further suppressed, the next largest contribution from our spatial exchange approach would come from the disorder of the sample or the electron-phonon coupling. The analysis of this spatial exchange also remains valid in systems that use additional quantum dots [65, 117], with universal quantum operations in mind, as long as each operation is applied to only two dots at a time.

While the movement of quantum dots requires the precise control of the confining potential, which may be difficult to realize experimentally, the analysis shows that the construction of such a system is viable. With realistic parameter values from current experiments, the analysis produces infidelities smaller than 10^{-4} after ten decoupling cycles. This setup could also be a productive step towards the experimental realization of more complicated exchange systems, with many more interesting applications.

Chapter 3

Majorana Bound States

Background

3.1 Historical Review: Topological Quantum Information

3.1.1 Majorana Fermions

The experimental observation of a self-conjugate fermionic particle with equal creation and annihilation operators, $\hat{\gamma}^\dagger = \hat{\gamma}$, known as a Majorana fermion, has been a goal in physics since it was first theorized by Ettore Majorana over 75 years ago as a real solution to the relativistic Dirac equation [74]. It should be noted that the creation ($\hat{c}^\dagger$) and annihilation ($\hat{c}$) operators of any conventional fermion can be trivialy decomposed into two self-conjugate forms

$$\hat{\gamma}^A = -i(\hat{c} - \hat{c}^\dagger) \tag{3.1}$$

$$\hat{\gamma}^B = \hat{c} + \hat{c}^\dagger. \tag{3.2}$$

While these operators are mathematically valid, they do not neccesarily correspond to physically meaningful particles that occur in nature.

While the existence of a self-conjugate, elementary particle is still an unsolved problem [98, 58, 115], there has been recent evidence, and subsequent interest, that a self-conjugate quasiparticle may also exist in condensed matter systems [37]. Besides being self-conjugate, these particles are novel because they arise from topological principles rather than broken symmetries alone. The rest of this section briefly reviews some of these systems; refer to the articles by Nayak et. al. and Hasan et. al. [83, 48] for a more thorough review.

3.1.2 Two-Dimensional Spinless p-wave Superconductor

One of the first systems theorized to host a self-conjugate fermion is a two-dimensional p-wave superconductor with spinless, or spin-polarized, particles as noted by Read and Green [95], given by the effective Hamiltonian

$$H = \sum_k \left[t_k \hat{c}_k^\dagger \hat{c}_k + \frac{1}{2} \left(\Delta_k^* \hat{c}_{-k} \hat{c}_k + \Delta_k \hat{c}_k^\dagger \hat{c}_{-k}^\dagger \right) \right], \tag{3.3}$$

where $t_k = \epsilon_k - \mu$, with single-particle kinetic energy ϵ_k, and Δ_k has p-wave symmetry (e.g. $\Delta_k \propto k_x - ik_y$). This system can be diagonalized by making a Bogoliubov-de Gennes transformation

$$\hat{a}_k = u_k \hat{c}_k - v_k \hat{c}_{-k}^\dagger \tag{3.4}$$

$$\hat{a}_k^\dagger = u_k^* \hat{c}_k^\dagger - v_k^* \hat{c}_{-k} \tag{3.5}$$

with the requirement that $|u_k|^2 + |v_k|^2 = 1$. The Bogoliubov coefficients u_k and v_k can be found by requiring that $[\hat{a}_k, H] = E_k \hat{a}_k$, which is consistent with

$$H = E_0 + \sum_k E_k \hat{a}_k^\dagger \hat{a}_k, \tag{3.6}$$

for some constant E_0 and positive quasiparticle energies, E_k. Together with the transformation, the commutation relation leads to the Bogoliubov-de Gennes(BdG) equations

$$t_k u_k - \Delta_k^* v_k = E_k u_k, \tag{3.7}$$

$$-\Delta_k u_k - t_k^* v_k = E_k v_k, \tag{3.8}$$

or equivalently in matrix form,

$$\begin{pmatrix} t_k & -\Delta_k^* \\ -\Delta_k & -t_k \end{pmatrix} \begin{pmatrix} u_k \\ v_k \end{pmatrix} = E_k \begin{pmatrix} u_k \\ v_k \end{pmatrix}, \tag{3.9}$$

which lead to the equations

$$E_k^2 = t_k^2 + |\Delta_k|^2, \tag{3.10}$$

$$\frac{v_k}{u_k} = \frac{t_k - E_k}{\Delta_k^*}, \tag{3.11}$$

$$|u_k|^2 = \frac{1}{2}\left(1 + \frac{t_k}{E_k}\right), \tag{3.12}$$

$$|v_k|^2 = \frac{1}{2}\left(1 - \frac{t_k}{E_k}\right). \tag{3.13}$$

The topological nature of this system can be seen by noting the asymptotic behavior of k. For large values of $|k|$, $\Delta_k \simeq 0$ and $\epsilon_k \gg \mu$, which implies $t_k \simeq E_k$. Thus as $|k| \to \infty$, $|u_k| \to 1$ and $|v_k| \to 0$ [see Eq.(3.13)]. For small k, $\epsilon_k \simeq k^2/(2m^*)$ and $\Delta_k \simeq \Delta(k_x - ik_y)$, so as $k \to 0$ then $t_k \to -\mu$ and $\Delta_k \to 0$, which together lead to $E_k^2 \to t_k^2$, yielding two distinct cases.

3.1.2.1 Topological Phases

When $\mu < 0$, $t_k > 0$ and $t_k \to E_k$ implies $|u_k| \to 1$ and $|v_k| \to 0$, while when $\mu > 0$, $t_k < 0$ and $t_k \to -E_k$ implies $|u_k| \to 0$ and $|v_k| \to 1$. These two cases have physically distinct phases: the first case, known as strong-pairing, has short-range wavefunctions, while the second case, known as weak-pairing, has long-range

wavefunctions [95]. More so, these two phases are topologically distinct, which can be seen by looking at the relationship between k and the Bogoliubov coefficients u_k and v_k.

Specifically, values of the Bogoliubov coefficients u_k and v_k, which satisfy $|u_k|^2 + |v_k|^2 = 1$, correspond to points on the surface of a sphere, S^2, with the point $|u_k| = 1$, $|v_k| = 0$ defined as the North pole, while the opposite case, $|u_k| = 0$, $|v_k| = 1$, is defined as the South pole. By adding the point at $|k| \to \infty$ as the North pole, the two-dimensional plane spanned by k_x and k_y corresponds topologically to S^2 as well. This defines a map from the k-space S^2 to u,v-space S^2 that forms the homotopy group $\pi_2(S^2)$, which is equivalent to the group of integers Z. This mapping has an integer topological invariant, known as the degree, that counts the minimum number of times a mapping must pass through any point other than the North pole [118, 95].

In the $\mu < 0$ case, both $|k| \to \infty$ and $k \to 0$ can be mapped to the point with $|u_k| = 1$, $|v_k| = 0$ at the North pole, meaning that this mapping can be smoothly deformed to the trivial mapping of all points to the North pole. Thus, the mapping does not need to pass through any point other than the North pole, and the degree of this phase is 0. For the $\mu > 0$ case, on the other hand, since $k \to 0$ corresponds to $|u_k| = 0$, $|v_k| = 1$, this mapping must include the South pole and cannot be smoothly deformed to the trivial mapping. Thus, this phase has degree 1 and these two phases are topologically distinct; the strong and weak pairing phases cannot be smoothly deformed into each other without changing the sign of μ.

3.1.2.2 Majorana Bound States and Non-Abelian Statistics

Since these two phases are topologically distinct, any system that includes both phases must contain a boundary where the topological parameter changes sign, i.e., $\mu = 0$. At this boundary value, the $k = 0$ solution has $t_0 = -\mu = 0$, meaning

$E_0 = 0$ since $\Delta_0 = 0$. Solving the BdG equations when $\mu = 0$ also yields $u_k = v_k^*$, meaning the $k = 0$ quasiparticle operator satisfies $\hat{a}_0^\dagger = \hat{a}_0$ [see Eq. (3.5)] and is thus a self-conjugate Majorana fermion. As a consequence, a system in the topologically non-trivial weak-pairing phase that contains a defect in the trivial phase, such as the core of a superconducting vortex, must have a zero-energy Majorana state bound to that defect.

While the guarantee of Majorana bound states(MBS) pinned to defects is interesting in itself, the real promise of these states come from their non-trivial exchange statistics. Unlike the wavefunctions of conventional bosons or fermions that only gain a phase of 1 or -1 when they are exchanged, these topological bound states can take on "any" phase upon exchange, and are known as "anyons" for this reason [95, 83].

Just as any conventional fermion can be decomposed into two self-conjugate fermions [see Eq. (3.2)], a pair of Majorana bound states $\hat{\gamma}_A$ and $\hat{\gamma}_B$, bound to two different defects, can be combined to form a conventional, albeit non-local, fermion by writing

$$\hat{d} = \frac{1}{2}(\hat{\gamma}_A + i\hat{\gamma}_B). \tag{3.14}$$

Since the MBS has zero-energy, the presence or absence of this fermion does not change the energy, meaning there are two degenerate ground states. More generally, $2n$ Majorana bound states form n conventional fermions, which can each be present or absent, forming a 2^n-fold ground state degeneracy. Physically exchanging any two of the bound states, known as braiding, applies a non-trivial unitary transformation to the degenerate manifold of ground states and can be used to perform quantum operations. Furthermore, when exchanging multiple pairs of Majorana bound states, the order of the braiding matters; the braiding operations do not commute, also known as "non-Abelian" in

the group theory context [83]. Since the MBS bound to topological defects are inherently non-local, they are immune to local pertubations, and thus, excellent candidates for fault-tolerant qubits.

3.1.3 Additional Topological Systems

Many other physical systems have been proposed to advance towards a more feasible experimental setup, though the underlying topological arguments often remain the same. For example, the non-Abelian statistics of Majorana bound states, also known as Ising anyons, were also discussed in the Moore-Read Pfaffian state of the $\nu = 5/2$ fractional quantum Hall state [81]. In addition, it is worth noting that the $\nu = 12/5$ Read-Reyazi state [96] may host more powerful topological states, known as Fibonacci anyons, which are capable of universal quantum computation [38, 83], unlike Ising anyons which require entangled states via magic state distillation [11].

Ivanov showed that equivalent MBS exist in the cores of half-quantum vortices (vortices where the superconducting phase shifts by π, rather than 2π, as it circles the core) of p-wave superconductors with spin [54], rather than the spinless or spin-polarized case discussed previously. Similarly, Kane and Mele found that similar states should exist in quantum spin Hall states of graphene [56].

Another important step came when Fu and Kane found that the required superconducting Cooper pairs with the rare p-wave symmetry could be replaced by Cooper pairs that tunnel from a conventional s-wave superconductor into a topological insulator via the proximity effect [39]. This proposal was advanced even further by Sau et. al. when they replaced the topological insulator with a ferromagnetic insulator and semiconductor with strong Rashba spin-orbit coupling [102]. Alicea simplified that proposal

even more by showing that the ferromagnetic insulator could be omitted if the semiconductor also had Dresselhaus spin-orbit coupling and a magnetic field was applied [2].

All of these gradual improvements culminated in the proposals by Lutchyn et. al. [73] and Oreg et al. [86] of a one-dimensional semiconductor nano-wire with strong spin-orbit coupling, on top of an s-wave superconductor in the presence of a magnetic field. Since it only involves relatively conventional materials and a one-dimensional system, their proposal was experimentally attempted by several groups. These groups reported zero-bias conductance peaks when connecting the nano-wire to a normal lead [82, 24, 20, 36, 18], indicating a single level at zero-energy that is consistent with a MBS at the wire-ends. This exciting development has spurred a great amount of interest in this system, and is the subject of the proposal in Ch. 4.

The remainder of this chapter will introduce and discuss some details of this "Majorana wire" system. However, it is helpful to first understand a simplified model analyzed by Kitaev [61], as well as Lieb et. al. in the context of the transverse Ising model [70].

3.2 Review of Kitaev's Toy Model

Kitaev's toy model is a 1D chain of N spinless electrons, with tight-binding hopping parameter t_0, chemical potential μ, and superconducting pairing $\Delta e^{i\theta}$ with phase θ on bonds (i.e. p-wave).

$$H = \mu \sum_{1\leq j\leq N} \left(c_j^\dagger c_j - \frac{1}{2} \right) + \sum_{1\leq j\leq N-1} \left[-t_0 (c_j^\dagger c_{j+1} + c_{j+1}^\dagger c_j) + \Delta e^{i\theta} c_j c_{j+1} + \Delta e^{-i\theta} c_{j+1}^\dagger c_j^\dagger \right] \tag{3.15}$$

This Hamiltonian is not meant to represent a physical system, but is useful in easily demonstrating the existence of localized unpaired Majorana modes at the ends of the

wire. With that said, one can think of a spinless model as a real system where one spin species is energetically favored, effectively eliminating spin as a true degree of freedom in the low-energy Hamiltonian. Indeed, the semiconductor wire system discussed in the next section can be directly mapped to the Kitaev model in certain parameter regimes.

Consider the self-conjugate Majorana operators (i.e. those that satisfy $\gamma^\dagger = \gamma$)

$$\gamma_j^A = -i(e^{i\frac{\theta}{2}}c_j - e^{-i\frac{\theta}{2}}c_j^\dagger) \qquad \gamma_j^B = e^{i\frac{\theta}{2}}c_j + e^{-i\frac{\theta}{2}}c_j^\dagger. \tag{3.16}$$

. One can easily check the anti-commutation relations

$$\{\gamma_j^A, \gamma_{j'}^A\} = -(\{c_j, -c_{j'}^\dagger\} + \{-c_j^\dagger, c_{j'}\}) = 2\delta_{jj'} \tag{3.17}$$

$$\{\gamma_j^B, \gamma_{j'}^B\} = \{c_j, c_{j'}^\dagger\} + \{c_j^\dagger, c_{j'}\} = 2\delta_{jj'} \tag{3.18}$$

$$\{\gamma_j^A, \gamma_{j'}^B\} = -i(\{c_j, c_{j'}^\dagger\} + \{-c_j^\dagger, c_{j'}\}) = 0 \tag{3.19}$$

to give the general $\{\gamma_j^\alpha, \gamma_{j'}^{\alpha'}\} = 2\delta_{jj'}\delta_{\alpha\alpha'}$. Note that there are two "species" of Majorana fermions that commute with each other, but which one is A or B is an arbitrary distinction. Equivalently, one can solve for the electron operators

$$c_j = \frac{1}{2}e^{-i\frac{\theta}{2}}(\gamma_j^B + i\gamma_j^A) \qquad c_j^\dagger = \frac{1}{2}e^{i\frac{\theta}{2}}(\gamma_j^B - i\gamma_j^A) \tag{3.20}$$

and substitute these into the Hamiltonian. Before doing this, note the useful property

$$\gamma_j^\alpha\gamma_j^\alpha = \frac{1}{2}(\gamma_j^\alpha\gamma_j^\alpha + \gamma_j^\alpha\gamma_j^\alpha) = \frac{1}{2}\{\gamma_j^\alpha, \gamma_j^\alpha\} = 1. \tag{3.21}$$

Thus the Hamiltonian in terms of the Majorana operators is given by

$$\begin{aligned}
H &= \frac{\mu}{4}\sum_{1\le j\le N}\left[(\gamma_j^B - i\gamma_j^A)(\gamma_j^B + i\gamma_j^A) - 2\right] \\
&\quad +\frac{1}{4}\sum_{1\le j\le N-1}\Big\{ - t_0\Big[(\gamma_j^B - i\gamma_j^A)(\gamma_{j+1}^B + i\gamma_{j+1}^A) + (\gamma_{j+1}^B - i\gamma_{j+1}^A)(\gamma_j^B + i\gamma_j^A)\Big] \\
&\qquad\qquad +\Delta\Big[(\gamma_j^B + i\gamma_j^A)(\gamma_{j+1}^B + i\gamma_{j+1}^A) + (\gamma_{j+1}^B - i\gamma_{j+1}^A)(\gamma_j^B - i\gamma_j^A)\Big]\Big\} \\
&= \frac{\mu}{4}\sum_{1\le j\le N}\left[1 + 1 + i(\gamma_j^B\gamma_j^A - \gamma_j^A\gamma_j^B) - 2\right] \\
&\quad +\frac{1}{4}\sum_{1\le j\le N-1}\Big\{ - t_0\Big[\gamma_j^B\gamma_{j+1}^B + \gamma_j^A\gamma_{j+1}^A + i(\gamma_j^B\gamma_{j+1}^A - \gamma_j^A\gamma_{j+1}^B)\Big] \\
&\qquad\qquad -t_0\Big[\gamma_{j+1}^B\gamma_j^B + \gamma_{j+1}^A\gamma_j^A + i(\gamma_{j+1}^B\gamma_j^A - \gamma_{j+1}^A\gamma_j^B)\Big] \\
&\qquad\qquad +\Delta\Big[\gamma_j^B\gamma_{j+1}^B - \gamma_j^A\gamma_{j+1}^A + i(\gamma_j^B\gamma_{j+1}^A + \gamma_j^A\gamma_{j+1}^B)\Big] \\
&\qquad\qquad +\Delta\Big[\gamma_{j+1}^B\gamma_j^B - \gamma_{j+1}^A\gamma_j^A - i(\gamma_{j+1}^B\gamma_j^A + \gamma_{j+1}^A\gamma_j^B)\Big]\Big\} \\
&= \frac{i}{2}\Big\{ - \mu\sum_{1\le j\le N}\gamma_j^A\gamma_j^B + \sum_{1\le j\le N-1}\Big[(t_0+\Delta)\gamma_j^A\gamma_{j+1}^B + (t_0-\Delta)\gamma_{j+1}^A\gamma_j^B\Big]\Big\}.
\end{aligned}$$

It is useful to analyze this Hamiltonian in two extreme parameters regimes. First, if $t_0 = \Delta = 0$ while $\mu \neq 0$ we have

$$\begin{aligned}
H &= -\frac{i}{2}\mu\sum_{1\le j\le N}\gamma_j^A\gamma_j^B = -\frac{1}{2}\mu\sum_{1\le j\le N}(e^{i\frac{\theta}{2}}c_j - e^{-i\frac{\theta}{2}}c_j^\dagger)(e^{i\frac{\theta}{2}}c_j + e^{-i\frac{\theta}{2}}c_j^\dagger) \quad (3.22)\\
&= \mu\sum_{1\le j\le N}c_j^\dagger c_j, \quad (3.23)
\end{aligned}$$

which is, not surprisingly, just a trivial 1D chain of fermions on the usual sites, with a ground state with no fermions present. This is a reminder that the substitution of Majorana operators is always mathematically possible, but in most cases doesn't lead to any physical significance. However, if $\mu = 0$ and $\Delta = t_0$ then

$$H = it_0\sum_{1\le j\le N-1}\gamma_j^A\gamma_{j+1}^B, \quad (3.24)$$

which importantly lacks the operators γ_N^A and γ_1^B. One can introduce new fermion operators that are localized between two adjacent sites

$$d_j = \frac{1}{2}(\gamma_j^A + i\gamma_{j+1}^B), \quad (3.25)$$

which are conventional in the sense that they are not self-conjugate and they satisfy the usual anti-commutation relations

$$\{d_j, d_{j'}\} = \frac{1}{4}\{\gamma_j^A + i\gamma_{j+1}^B, \gamma_{j'}^A + i\gamma_{j'+1}^B\} = \frac{1}{2}(\delta_{jj'} - \delta_{jj'}) = 0, \tag{3.26}$$

$$\{d_j^\dagger, d_{j'}^\dagger\} = \frac{1}{4}\{\gamma_j^A - i\gamma_{j+1}^B, \gamma_{j'}^A - i\gamma_{j'+1}^B\} = \frac{1}{2}(\delta_{jj'} - \delta_{jj'}) = 0, \tag{3.27}$$

$$\{d_j^\dagger, d_{j'}\} = \frac{1}{4}\{\gamma_j^A - i\gamma_{j+1}^B, \gamma_{j'}^A + i\gamma_{j'+1}^B\} = \frac{1}{2}(\delta_{jj'} + \delta_{jj'}) = \delta_{jj'}. \tag{3.28}$$

Using

$$d_j^\dagger d_j = \frac{1}{4}(\gamma_j^A - i\gamma_{j+1}^B)(\gamma_j^A + i\gamma_{j+1}^B) = \frac{1}{4}\Big[2 + i(\gamma_j^A\gamma_{j+1}^B - \gamma_{j+1}^B\gamma_j^A)\Big] = \frac{1}{2} + \frac{i}{2}\gamma_j^A\gamma_{j+1}^B \tag{3.29}$$

the Hamiltonian can be written in terms of the "conventional" fermions operators

$$H = 2t_0 \sum_{1\le j\le N-1} \Big(d_j^\dagger d_j - \frac{1}{2}\Big). \tag{3.30}$$

The significant point is that the missing end Majorana operators can be combined to form a conventional, but non-local, fermion operator

$$d_{\text{end}} = \frac{1}{2}(\gamma_N^A + i\gamma_1^B), \tag{3.31}$$

with

$$d_{\text{end}}^\dagger d_{\text{end}} = \frac{1}{4}(\gamma_N^A - i\gamma_1^B)(\gamma_N^A + i\gamma_1^B) = \frac{1}{4}(2 + 2i\gamma_N^A\gamma_1^B) \tag{3.32}$$

$$= \frac{1}{2} + \frac{i}{2}\gamma_N^A\gamma_1^B, \tag{3.33}$$

or equivalently,

$$i\gamma_N^A\gamma_1^B = 2d_{\text{end}}^\dagger d_{\text{end}} - 1. \tag{3.34}$$

In order to include every operator in the Hamiltonian, one could write

$$H = E_{\text{g}} + \epsilon_0 d_{\text{end}}^\dagger d_{\text{end}} + 2t_0 \sum_{1\le j\le N-1} d_j^\dagger d_j \tag{3.35}$$

with $E_g = -2t_0(N-1)$ and $\epsilon_0 = 0$. Thus, Majorana operators at the ends of the chain form a zero-energy fermion, while the conventional localized fermions in the bulk of the chain require energy $2t_0$. Thus the system has a ground-state degeneracy consisting of the states with, and without, the non-local fermion. The operator that measures the presence of the non-local fermion is

$$-i\gamma_N^A \gamma_1^B = 1 - 2d_{\text{end}}^\dagger d_{\text{end}}, \tag{3.36}$$

with eigenvalues -1 and $+1$ corresponding to the presence and absence of the non-local fermion, respectively. Thus, the Kitaev toy model also has two distinct phases, with and without the presence of a zero-energy Majorana mode, sometimes known as a Majorana zero mode.

While the case of general parameters requires more complicated calculations, the above result generalizes for all parameter values. However, instead of the MBS existing only at the end sites, the distribution of these two states decays exponentially into the wire, and thus have a small overlap that slightly breaks the degeneracy of the two ground states.

3.3 Majorana Wire System Band Structure

While the more realistic Majorana wire proposal by Lutchyn et. al. [73] and Oreg et al. [86] is more complicated then the Kitaev model, the band structure in the uniform case shows that similar states emerge. The full Hamiltonian of the system is given by

$$H_S = H_{TB} + H_{SO} + H_Z + H_{SC}, \tag{3.37}$$

where

$$H_{\rm TB} = \sum_{j\sigma}\left[(2t_0-\mu)\hat{c}^\dagger_{j\sigma}\hat{c}_{j\sigma} - t_0\hat{c}^\dagger_{j\pm1,\sigma}\hat{c}_{j\sigma}\right], \tag{3.38}$$

$$H_{\rm SO} = \sum_{j\sigma}\left[\frac{\alpha}{2}\,s(\sigma)\left(\hat{c}^\dagger_{j\bar{\sigma}}\hat{c}_{j+1,\sigma} - \hat{c}^\dagger_{j+1,\bar{\sigma}}\hat{c}_{j\sigma}\right)\right], \tag{3.39}$$

$$H_{\rm Z} = \sum_{j\sigma}\left[s(\sigma)V^z\hat{c}^\dagger_{j\sigma}\hat{c}_{j\sigma} + V^{s(\sigma)}\hat{c}^\dagger_{j\bar{\sigma}}\hat{c}_{j\sigma}\right], \tag{3.40}$$

$$H_{\rm SC} = \sum_{j}\left(\Delta\hat{c}^\dagger_{j\uparrow}\hat{c}^\dagger_{j\downarrow} + \Delta^*\hat{c}_{j\downarrow}\hat{c}_{j\uparrow}\right), \tag{3.41}$$

are the tight-binding, spin-orbit, Zeeman, and proximity-effect superconducting terms, respectively. Here $\hat{c}_{j\sigma}$ is the electron annihilation operator for spin σ at site j, $t_0 = \hbar^2/(2m^*a^2)$ is the tight-binding coefficient with effective mass m^* and lattice size a, μ is the chemical potential, $\alpha/2$ is the Rashba coupling, $\mathbf{V} = g\mu_{\rm B}\mathbf{B}/2$ is the Zeeman coupling with $V^\pm = V^x \pm iV^y$ used for the terms perpendicular to the wire axis, and Δ is the s-wave pairing potential. The coefficient $s(\sigma)$ stands for $+$ and $-$ when σ is $\uparrow$ and $\downarrow$, respectively, and $\bar{\sigma}$ denotes the opposite spin.

Consider the tight-binding terms in the lattice Hamiltonian, transferring to momentum space by using

$$c_{j\sigma} = \sum_k \langle j\sigma|k\rangle c_{k\sigma} = \sum_k e^{ikja}c_{k\sigma} \tag{3.42}$$

where $|j\sigma\rangle$ is the state at site j, at $x = ja$, and $c_{k\sigma}$ annihilates an electron of momentum k and spin σ. Thus, the tight-binding term in k-space can be written

$$\begin{aligned}
H &= \sum_{j\sigma kk'}\left[-t_0(e^{-ik(j+1)a}e^{ik'ja} + e^{-ik(j-1)a}e^{ik'ja}) + (2t_0-\mu)e^{-ikja}e^{ik'ja}\right]c^\dagger_{k\sigma}c_{k'\sigma} \\
&= \sum_{j\sigma kk'}\left[-t_0(e^{-ika}e^{-i(k-k')ja} + e^{ika}e^{-i(k-k')ja}) + (2t_0-\mu)e^{-i(k-k')ja}\right]c^\dagger_{k\sigma}c_{k'\sigma} \\
&= \sum_{\sigma k}\left[-t_0(e^{-ika}+e^{ika}) + (2t_0-\mu)\right]c^\dagger_{k\sigma}c_{k\sigma} \\
&= \sum_{\sigma k}\left[-2t_0\cos(ka) + (2t_0-\mu)\right]c^\dagger_{k\sigma}c_{k\sigma} \\
&= \sum_{\sigma k}\left\{2t_0[1-\cos(ka)] - \mu\right\}c^\dagger_{k\sigma}c_{k\sigma},
\end{aligned} \tag{3.43}$$

while the spin-orbit term is written

$$\sum_{j\sigma kk'} \frac{\alpha}{2} s(\sigma)[e^{-ikja}e^{ik'(j+1)a}c^\dagger_{k\bar{\sigma}}c_{k'\sigma} - e^{-ik(j+1)a}e^{ik'ja}c^\dagger_{k\bar{\sigma}}c_{k'\sigma}] \tag{3.44}$$

$$= \sum_{j\sigma kk'} \frac{\alpha}{2} s(\sigma)[e^{ik'a} - e^{-ika}]e^{-i(k-k')ja}c^\dagger_{k\bar{\sigma}}c_{k'\sigma} \tag{3.45}$$

$$= \sum_{\sigma k} \frac{\alpha}{2} s(\sigma)[e^{ika} - e^{-ika}]c^\dagger_{k\bar{\sigma}}c_{k\sigma} = \sum_{\sigma k} i\alpha s(\sigma)\sin(ka)c^\dagger_{k\bar{\sigma}}c_{k\sigma}, \tag{3.46}$$

and the superconducting terms are

$$\sum_{jkk'} (\Delta e^{-ikja}e^{-ik'ja}c^\dagger_{k\uparrow}c^\dagger_{k'\downarrow} + \Delta^* e^{ikja}e^{ik'ja}c_{k\downarrow}c_{k'\uparrow}) \tag{3.47}$$

$$= \sum_{jkk'} (\Delta e^{-i(k+k')ja}c^\dagger_{k\uparrow}c^\dagger_{k'\downarrow} + \Delta^* e^{i(k+k')ja}c_{k\downarrow}c_{k'\uparrow}) \tag{3.48}$$

$$= \sum_{k} (\Delta c^\dagger_{k\uparrow}c^\dagger_{-k\downarrow} + \Delta^* c_{-k\downarrow}c_{k\uparrow}). \tag{3.49}$$

This can be repeated for the remaining terms in the lattice Hamiltonian to give

$$H = \sum_{\sigma k} \{2t_0[1-\cos(ka)] - \mu + s(\sigma)V^z\} c^\dagger_{k\sigma}c_{k\sigma} \tag{3.50}$$

$$+ \sum_{\sigma k} \left[i\alpha s(\sigma)\sin(ka) + V^{s(\sigma)}\right] c^\dagger_{k\bar{\sigma}}c_{k\sigma} \tag{3.51}$$

$$+ \sum_{k} (\Delta c^\dagger_{k\uparrow}c^\dagger_{-k\downarrow} + \Delta^* c_{-k\downarrow}c_{k\uparrow}). \tag{3.52}$$

Separating the terms with negative-momentum and reordering gives

$$H = \sum_{k\geq 0,\sigma} \{2t_0[1-\cos(ka)] - \mu + s(\sigma)V^z\} (c^\dagger_{k\sigma}c_{k\sigma} + c^\dagger_{-k\sigma}c_{-k\sigma}) \tag{3.53}$$

$$+ \sum_{k\geq 0,\sigma} \left[i\alpha s(\sigma)\sin(ka) + V^{s(\sigma)}\right] c^\dagger_{k\bar{\sigma}}c_{k\sigma} \tag{3.54}$$

$$+ \sum_{k\geq 0,\sigma} \left[-i\alpha s(\sigma)\sin(ka) + V^{s(\sigma)}\right] c^\dagger_{-k\bar{\sigma}}c_{-k\sigma} \tag{3.55}$$

$$+ \sum_{k\geq 0} \Delta(c^\dagger_{k\uparrow}c^\dagger_{-k\downarrow} + c^\dagger_{-k\uparrow}c^\dagger_{k\downarrow}) + \Delta^*(c_{-k\downarrow}c_{k\uparrow} + c_{k\downarrow}c_{-k\uparrow}) \tag{3.56}$$

$$= \sum_{k\geq 0,\sigma} \{2t_0[1-\cos(ka)] - \mu + s(\sigma)V^z\} [c^\dagger_{k\sigma}c_{k\sigma} + (1 - c_{-k\sigma}c^\dagger_{-k\sigma})] \tag{3.57}$$

$$+ \sum_{k\geq 0,\sigma} \left[i\alpha s(\sigma)\sin(ka) + V^{s(\sigma)}\right] c^\dagger_{k\bar{\sigma}}c_{k\sigma} \tag{3.58}$$

$$+ \sum_{k\geq 0,\sigma} \left[i\alpha s(\sigma)\sin(ka) - V^{s(\sigma)}\right] c_{-k\sigma}c^\dagger_{-k\bar{\sigma}} \tag{3.59}$$

$$+ \sum_{k\geq 0} \Delta(c^\dagger_{k\uparrow}c^\dagger_{-k\downarrow} - c^\dagger_{k\downarrow}c^\dagger_{-k\uparrow}) + \Delta^*(c_{-k\downarrow}c_{k\uparrow} - c_{-k\uparrow}c_{k\downarrow}) \tag{3.60}$$

By defining $h(k) = 2t_0[1-\cos(ka)]-\mu$, one can write the momentum-space Hamiltonian concisely in matrix form as $H = E_{\rm g}^k + \sum_{k\geq 0} K^\dagger H_k K$ with $E_{\rm g}^k = 2\sum_{k\geq 0} h(k)$,

$$K^\dagger = \begin{pmatrix} c_{k\uparrow}^\dagger & c_{k\downarrow}^\dagger & c_{-k\uparrow} & c_{-k\downarrow} \end{pmatrix} \tag{3.61}$$

and

$$H_k = \begin{pmatrix} h(k)+V^z & -i\alpha\sin(ka)+V^- & 0 & \Delta \\ i\alpha\sin(ka)+V^+ & h(k)-V^z & -\Delta & 0 \\ 0 & -\Delta^* & -h(k)-V^z & i\alpha\sin(ka)-V^+ \\ \Delta^* & 0 & -i\alpha\sin(ka)-V^- & -h(k)+V^z \end{pmatrix}, \tag{3.62}$$

which has the expected block form

$$\begin{pmatrix} H(k) & \Delta \\ -\Delta^* & -H^*(-k) \end{pmatrix}. \tag{3.63}$$

The eigenvalues can be found, but are generally quite complicated. Instead, consider the simpler case of just the tight-binding, Zeeman, and spin-orbit terms. Without the superconducting terms, the eigenvalues of the top-left 2×2 block are

$$\epsilon_\pm(k) = h(k) \pm \sqrt{V_\perp^2 + [V_y + \alpha\sin(ka)]^2}, \tag{3.64}$$

where $V_\perp^2 = V_x^2 + V_z^2$ is the Zeeman field perpendicular to the spin-quantization axis caused by the spin-orbit coupling. The relevant physics occurs near $k = 0$, so consider the $k \ll a$ limit. The tight-binding and spin-orbit terms give

$$\epsilon_\pm(k) = t_0(ka)^2 \pm \alpha ka - \mu \tag{3.65}$$

$$\iff \epsilon/t_0 = x^2 \pm \frac{\alpha}{t_0}x - \frac{\mu}{t_0} = x\left(x \pm \frac{\alpha}{t_0}\right) - \frac{\mu}{t_0} \tag{3.66}$$

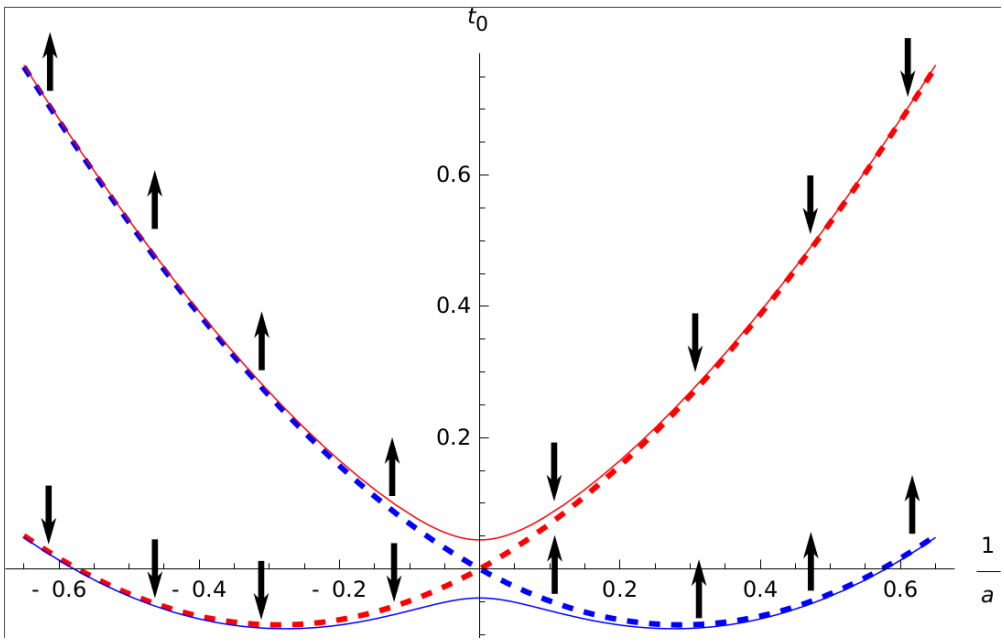

Figure 3.1: Band structure with $\alpha = 6.666$, $t_0 = 11.3$, and $V_\perp = 0.5$: Dashed is tight-binding and spin-orbit only, while solid also includes perpendicular Zeeman field, plotting with $\mu = 0$ and implicitly understanding that μ is the $y = 0$ axis. The units for the y and x axis are t_0 and $1/a$ respectively.

with $x \equiv ka$, so the spin-orbit shifts the spin bands to be centered at $\pm\frac{\alpha}{2t_0}$ with $k_\mathrm{F} = \pm(\alpha + \sqrt{\alpha^2 + 4\mu t_0})/2t_0 a$. The introduction of the perpendicular Zeeman terms gives

$$\epsilon_\pm/t_0 = x^2 \pm \frac{1}{t_0}\sqrt{V_\perp^2 + \alpha^2 x^2} - \frac{\mu}{t_0}, \tag{3.67}$$

which opens up a gap of $2V_\perp/t_0$ at $k = 0$.

For the introduction of the superconducting pairing terms, consider the whole 4×4 matrix, which has the particle-hole symmetry that results in symmetric eigenvalues about $E = 0$. For this reason, consider the square of the excitation spectrum

$$\epsilon_\pm^2(k) = h^2 + V_\perp^2 + \Delta^2 + \alpha^2\sin^2(ka) \pm 2\sqrt{V_\perp^2(h^2+\Delta^2) + h^2\alpha^2\sin^2(ka)}, \tag{3.68}$$

which at $k = 0$ is

$$\epsilon_\pm^2(0) = \mu^2 + V_\perp^2 + \Delta^2 \pm 2V_\perp\sqrt{\mu^2+\Delta^2} \tag{3.69}$$

$$= (V_\perp \pm \sqrt{\mu^2+\Delta^2})^2. \tag{3.70}$$

So $\epsilon_-(0) = |V_\perp - \sqrt{\mu^2 + \Delta^2}|$. Thus the gap closes when $V_\perp = \sqrt{\mu^2 + \Delta^2}$ and, just like the Kitaev toy model, this marks the transition between the two distinct phases.

To see which side of this transition yields localized modes, consider the band structure in the figure above. With just the tight-binding and spin-orbit terms present, the Kramer's degeneracy is split, forming two spin-bands parallel and anti-parallel to the spin-orbit quantization axis (shown in the figure as up and down arrows). The introduction of the Zeeman term perpendicular to that axis results in a gap which creates two bands with different "spin-momentum" species. In the upper band, the electrons with positive momentum have one spin (down in the figure), while the electrons with the opposite momentum have the opposite spin (up in the figure). Similarly in the lower band, the momentum and spin are tied together. This relationship between spin and momentum, which effectively reduces the degrees of freedom of the system, is known as a helical liquid. If the chemical potential lies in the gap (and $k_\mathrm{B}T \ll (V_\perp - \mu)/t_0$) the upper band becomes energetically unavailable and the system can be projected to the lower band. While the lower band still contains opposite spins for opposite momentum (as required for the s-wave pairing which will be introduced) the spin is not a true degree of freedom; in some sense the system is "spinless" like Kitaev's toy model.

As mentioned above, the energy spectrum for more general parameters in this system can be found analytically, but are quite complicated and not particularly helpful. Instead, the lowest energy level and the gap to next highest energy level was simulated using the code in App. B, and are plotted below with $V_z = 0.9$, $V_y = V_x = 0$, $\Delta = 0.5$, $\mu = 0$ and $\alpha = 3$.

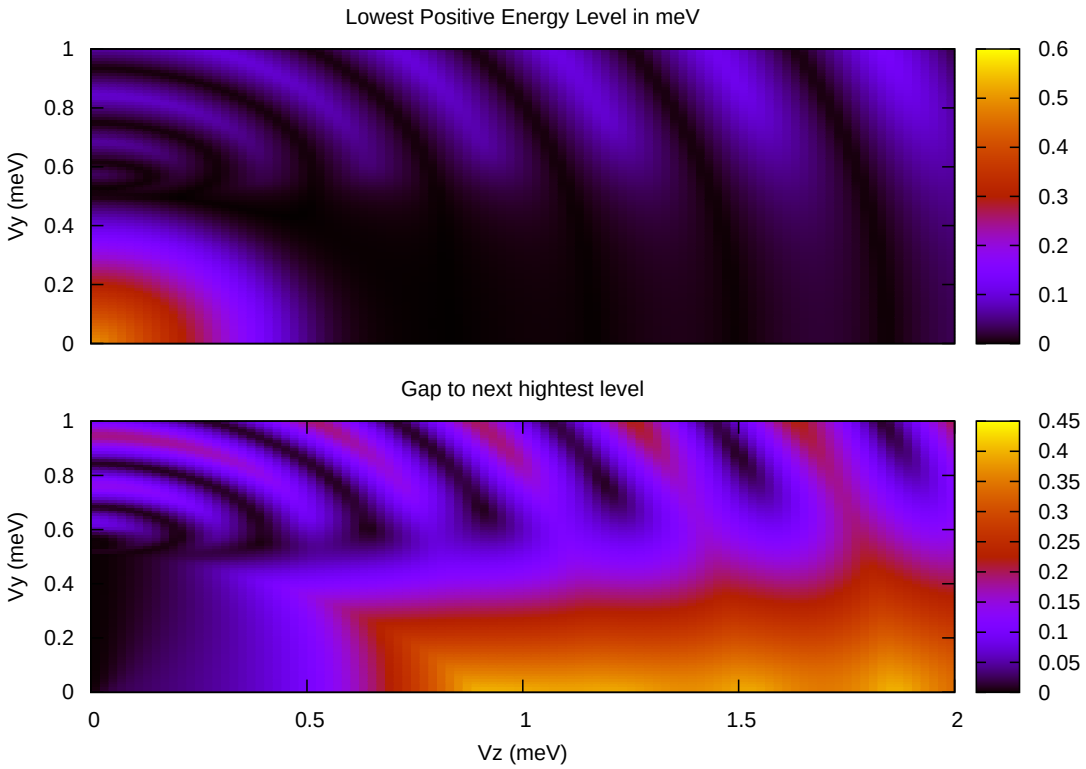

Figure 3.2: Modes near zero energy are robust near $V_z \sim 0.5$, and small $V_y < 0.3$. It should be noted that V_x could be used equivalently since it is also perpendicular to the spin-orbit quantization axis.

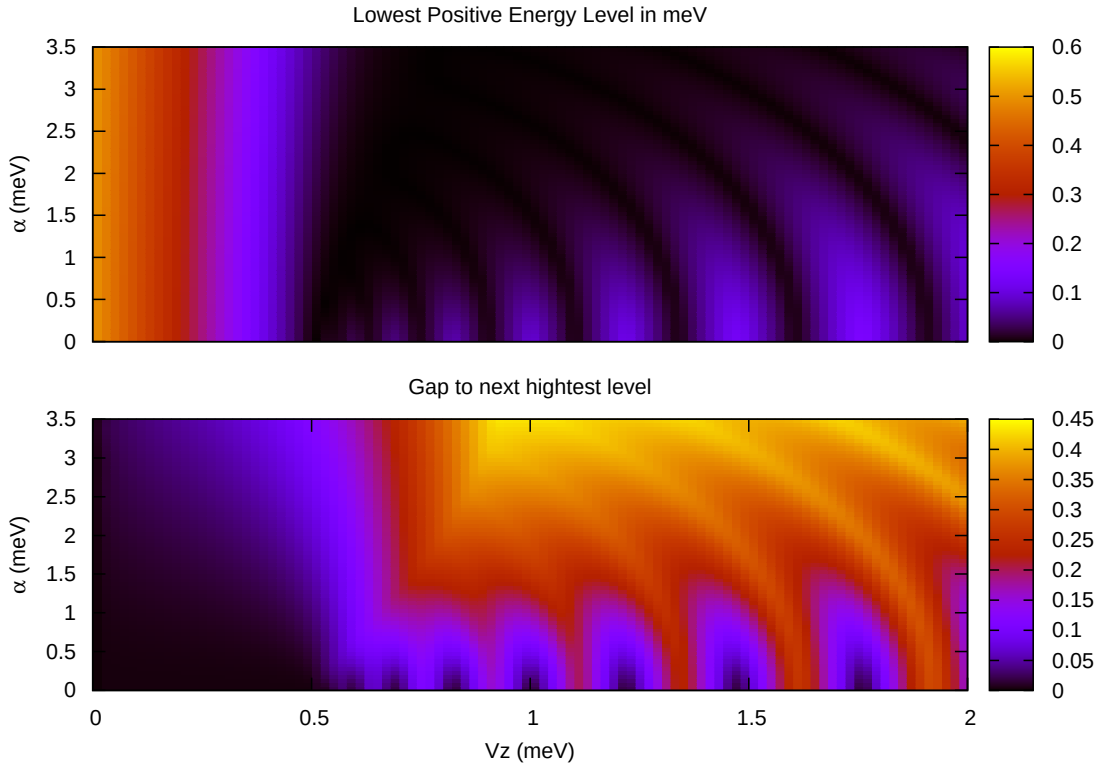

Figure 3.3: Modes near zero energy are robust near $V_z \sim 0.9$, and large $\alpha > 1.5$. There seems to be little benefit to increasing α beyond 3.

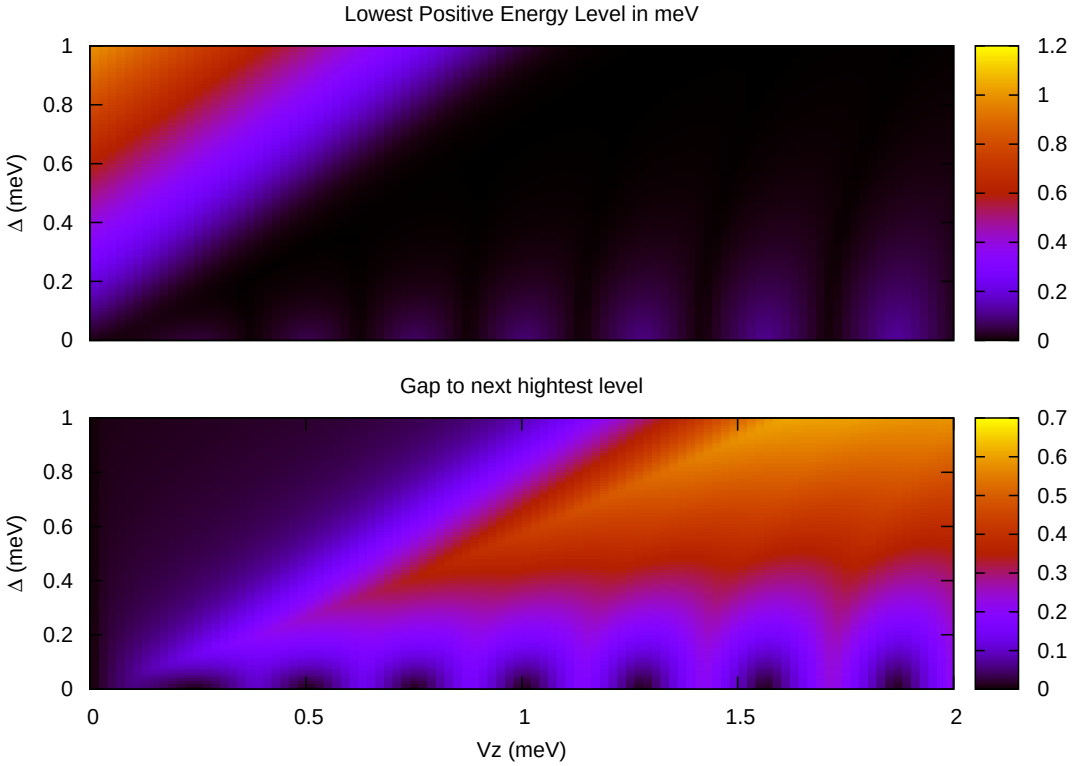

Figure 3.4: The results are in general agreement that the topological threshold is given by $\mu^2 < V_z^2 - \Delta^2$, which for $\mu = 0$ is below the line of slope 1. On the other hand, a Δ value that is too small ($\Delta < 0.3$) doesn't create a sufficiently large gap, so this equation doesn't apply for small Δ. Thus, robust modes only appear for $V_z \geq 0.5$, and decent window of feasible Δ doesn't begin until $V_z \geq 0.9$.

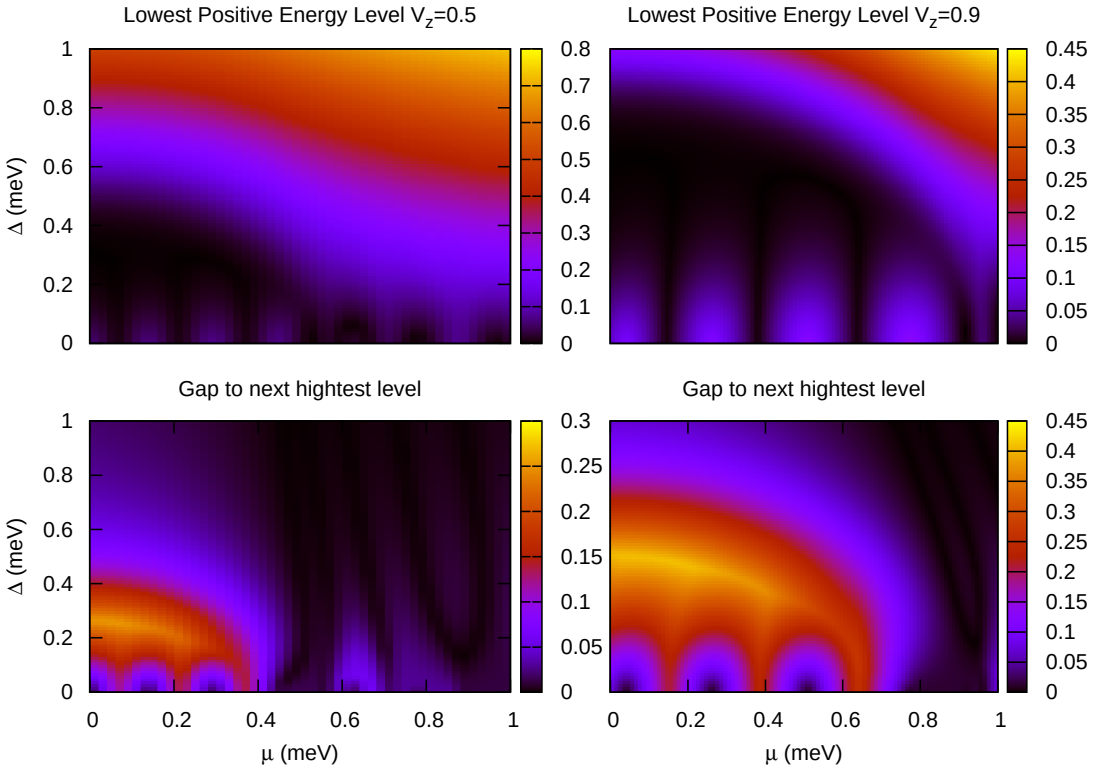

Figure 3.5: The results are in general agreement that the topological threshold is given by $\mu^2 < V_z^2 - \Delta^2$, which is a circle of radius V_z. On the other hand, a Δ value that is too small ($\Delta < 0.2$) doesn't create a sufficiently large gap, so this equation doesn't apply for small Δ. Thus, robust modes are best achieved with higher $V_z \sim 0.9$, where $\Delta \sim 0.5$ is ideal for small μ values.

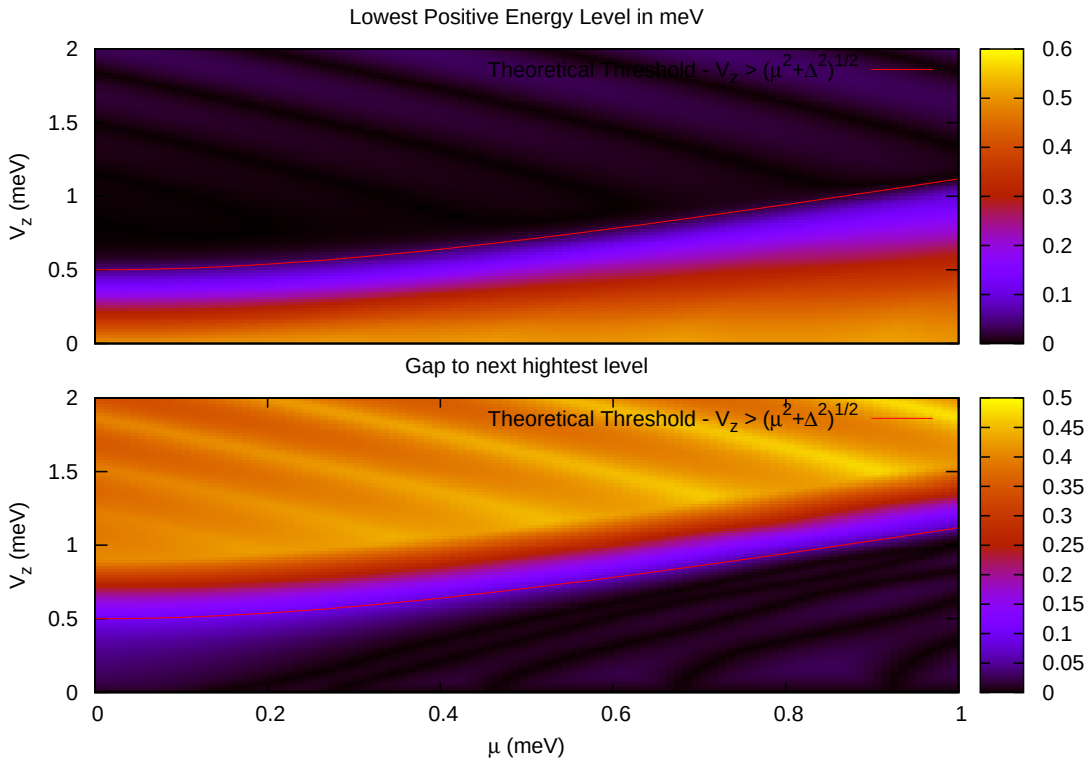

Figure 3.6: The results are in general agreement that the topological threshold is given by $\mu^2 < V_z^2 - \Delta^2$, which is plotted as a red line. Robust modes are best achieved with higher $V_z > 0.75$ for small μ values.

Chapter 4

Majorana Wire

4.1 Introduction

While Majorana zero modes bound to the ends of semiconductor nanowires as discussed in the previous chapter are theoretically supported by models, further evidence is needed to rule out alternative explanations [71, 101, 68]. Perhaps the most definitive signature of Majorana bound states in these "Majorana wires" would be the demonstration of their non-trivial braiding statistics. While braiding is ultimately needed for topological quantum computation, it remains an ambitious experimental task. With this in mind, simpler experiments are desired to provide insight and direct further research before braiding is attempted. Though there have been feasible tests proposed and performed on several aspects of the system [50, 49, 15, 21, 99, 72, 103], such as qubit measurement, there is still no clear consensus on the presence of Majorana bound states [37].

Observing entanglement of these states in Majorana wires would not only be a significant step towards their verification, but would also demonstrate their potential utility for topological quantum computation. While tests of quantum entanglement

with Ising anyons have been discussed formally [13, 16], the goal of this chapter is to devise and analyze a more concrete protocol motivated by the recent experimental developments discussed in the previous chapter.

Thus we propose a procedure for demonstrating Bell's theorem with three pairs of Majorana bound states in semiconductor nanowire systems [see Fig. 4.1]. Specifically, our procedure can be used to test the Bell [7] and Clauser-Horne-Shimony-Holt (CHSH) [19] inequalities using only two operations on maximally entangled states, which can be prepared using the same operations and projective measurement [see Fig. 4.2]. These operations are accomplished by moving the domain walls along the axis of the wire using "keyboard" gates already needed for braiding [1]. Hence, our proposal may also serve as a step towards experiments that perform topological operations.

The remainder of this chapter will proceed as follows: Sec. 4.2 introduces a simplified model for the Majorana wire and defines a qubit basis. Sec. 4.3 summarizes the entanglement inequalities and lay out the procedure for testing them. Sec. 4.4 introduces a more realistic description of the semiconductor nanowire system, discuss corresponding simulation results, and introduces a simpler version of the CHSH experiment. Sec. 4.5 discusses experimental considerations and Sec. 4.7 summarizes the proposal. Modifications to this procedure for different measurement outcomes are discussed in Section 4.6.

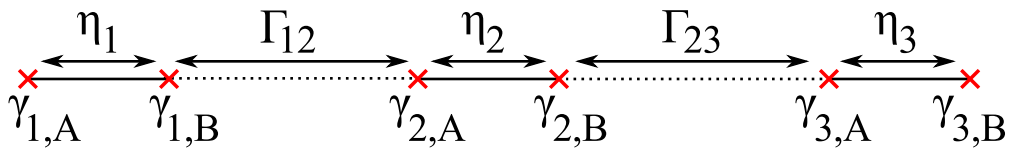

Figure 4.1: The wires are segmented into three regions where the wires are in the topological phase (solid lines). Majorana bound states, represented by red x's, are localized at the ends of these regions. Majorana bound states at the ends of the same topological region are coupled by η, while neighboring topological regions are coupled by Γ.

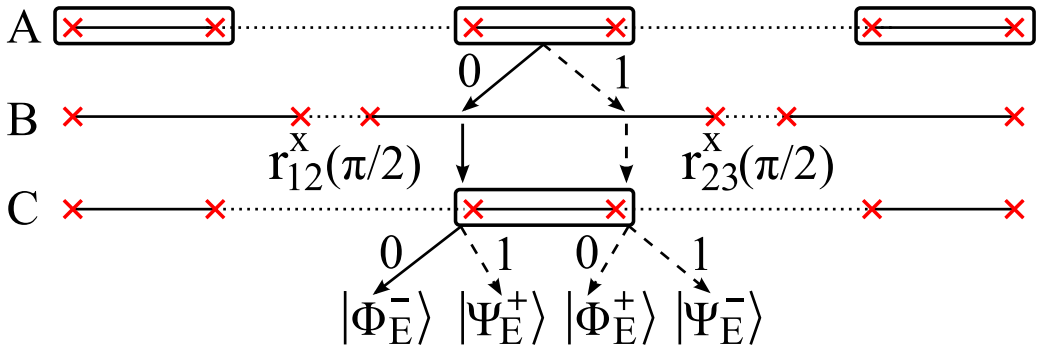

Figure 4.2: Preparation of the maximally entangled states of even total parity. A) The occupation of all three topological regions is measured, represented by rectangles around each region. B) The topological regions (solid lines) are expanded to perform $\pi/2$ rotations about the x-axis for the left and right logical qubits. C) The middle qubit is measured, projecting to one of the four maximally entangled states of even total parity. Different measurement outcomes are shown; when both middle measurements are 0, the $|\Phi^-_E\rangle$ state is prepared as discussed in Sec. 4.3 (solid arrows), while other outcomes (dashed arrows) are discussed in Section 4.6.

4.2 Majorana Model Hamiltonian

To discuss the salient features of the Majorana wire system, begin by considering a description similar to the toy model analyzed by Kitaev [61]. With appropriate parameters, the wire is driven into a topological phase with an unpaired Majorana fermion at each end [73, 86, 1]. If the parameters vary spatially (e.g., non-uniform chemical potential) there may be multiple topological regions separated by non-topological

regions, with a Majorana fermion localized at each domain wall separating the two regions. For this proposal, consider the case with three topological regions separated by two non-topological regions [see Fig. 4.1] with the Majorana Hamiltonian

$$H = i\eta_1\hat{\gamma}_{1,A}\hat{\gamma}_{1,B} + i\eta_2\hat{\gamma}_{2,A}\hat{\gamma}_{2,B} + i\eta_3\hat{\gamma}_{3,A}\hat{\gamma}_{3,B} + i\Gamma_{12}\hat{\gamma}_{1,B}\hat{\gamma}_{2,A} + i\Gamma_{23}\hat{\gamma}_{2,B}\hat{\gamma}_{3,A}, \qquad (4.1)$$

where η describes the coupling between Majorana bound states at the ends of a single topological region, while Γ describes the coupling of neighboring topological regions, assuming that all couplings decay exponentially as the Majorana bound states separate from their nearest neighbors. Each topological region has two types of Majorana operators, denoted by index A or B, that form a conventional fermion operator $\hat{d}_n = \frac{1}{2}(\hat{\gamma}_{n,A} + i\hat{\gamma}_{n,B})$, and satisfy $\{\hat{\gamma}_i, \hat{\gamma}_j\} = 2\delta_{ij}$, where i, j specifies both the region and type.

The parity of the occupation number for the conventional fermions, (i.e., the eigenstate of $\hat{N}_n \equiv \hat{d}_n^\dagger \hat{d}_n$), will serve as the degree of freedom for our qubits. A computational basis is specified with the conventional fermions by defining the state $|000\rangle$ such that $\hat{d}_n|000\rangle = 0$ for all n and using the ordering conventions given by

$$\begin{aligned}
|000\rangle &&& |010\rangle = \hat{d}_2^\dagger|000\rangle \\
|011\rangle &= \hat{d}_2^\dagger\hat{d}_3^\dagger|000\rangle &&|001\rangle = \hat{d}_3^\dagger|000\rangle \\
|110\rangle &= \hat{d}_1^\dagger\hat{d}_2^\dagger|000\rangle &&|100\rangle = \hat{d}_1^\dagger|000\rangle \\
|101\rangle &= \hat{d}_1^\dagger\hat{d}_3^\dagger|000\rangle &&|111\rangle = \hat{d}_1^\dagger\hat{d}_2^\dagger\hat{d}_3^\dagger|000\rangle.
\end{aligned} \qquad (4.2)$$

Since this model describes a system with superconductivity, the total number of particles is conserved modulo 2. This restriction splits the basis into two sub-bases, S_{E} and S_{O}, with an even and odd number of total particles (i.e., total parity), which are the left and right columns of Eqs. (4.2), respectively. A state from one basis cannot evolve into

a state from the other basis since they differ by a single particle. Strictly speaking, the two bases can interact if we account for quasi-particle poisoning in our model [94], but this occurs on a much longer time-scale than our proposed operations as discussed in Sec. 4.5. The middle occupation number is used to preserve the total parity rather than storing unique quantum information. Thus, two logical qubits are encoded in the left and right topological regions while the occupation of the middle region is forfetied as the "parity qubit".

By writing the Majorana operators in terms of the conventional fermions with $\hat{\gamma}_{n,A} = \hat{d}_n + \hat{d}_n^\dagger$ and $i\hat{\gamma}_{n,B} = \hat{d}_n - \hat{d}_n^\dagger$, the Hamiltonian in this basis is

$$H = -\eta_1(\sigma^z \otimes \sigma^0 \otimes \sigma^0) - \eta_2(\sigma^0 \otimes \sigma^z \otimes \sigma^0) - \eta_3(\sigma^0 \otimes \sigma^0 \otimes \sigma^z) \\ - \Gamma_{12}(\sigma^x \otimes \sigma^x \otimes \sigma^0) - \Gamma_{23}(\sigma^0 \otimes \sigma^x \otimes \sigma^x). \quad (4.3)$$

The η terms for each topological region perform the σ^z operation for their corresponding qubits, while the σ^x operation is performed on the neighboring qubits involved in the Γ terms. Thus rotations on the Bloch spheres of the qubits can be made by adjusting the parameters of the wire to suppress the couplings of all but one term in the Hamiltonian. For example, if all the couplings other than Γ_{12} are negligible, the evolution operator after time T is

$$r_{12}^x(\theta) \equiv \exp\left[i\frac{\theta}{2}(\sigma^x \otimes \sigma^x \otimes \sigma^0)\right] \quad (4.4)$$

$$= \cos\frac{\theta}{2}(\sigma^0 \otimes \sigma^0 \otimes \sigma^0) + i\sin\frac{\theta}{2}(\sigma^x \otimes \sigma^x \otimes \sigma^0), \quad (4.5)$$

where $\theta = 2\Gamma_{12}T/\hbar$ is the angle that qubits 1 and 2 rotate about the x-axis of their respective Bloch spheres. By adjusting the parameters appropriately, one can perform all the Bloch sphere rotations necessary for this proposal.

4.3 Entanglement Inequalities

Before testing the Bell and CHSH inequalities, this section discusses the preparation of one of the four maximally entangled states of even parity,

$$|\Phi_{\mathrm{E}}^{\pm}\rangle = \frac{|000\rangle \pm |101\rangle}{\sqrt{2}}, \qquad |\Psi_{\mathrm{E}}^{\pm}\rangle = \frac{|011\rangle \pm |110\rangle}{\sqrt{2}} \tag{4.6}$$

using the operations already discussed and projective measurement. To begin the preparation, the parity of each topological region is measured, fixing the total parity and projecting to one of the basis states. The inequalities can be tested equivalently with any of the maximally entangled states from either parity, but for conciseness, assume the total parity is even for the rest of the body of this paper, and consider only the inequalities with $|\Phi_{\mathrm{E}}^{-}\rangle$, assuming the initially measured state is $|000\rangle$. This proposal can be accomplished for general initial conditions by altering the procedure slightly as described in Section 4.6. If a $\pi/2$ rotation about the x-axis is performed for both logical qubits the resulting state is $r_{12}^{x}(\pi/2)r_{23}^{x}(\pi/2)|000\rangle$, or

$$\frac{|000\rangle - |101\rangle + i|011\rangle + i|110\rangle}{2} = \frac{|\Phi_{\mathrm{E}}^{-}\rangle + i|\Psi_{\mathrm{E}}^{+}\rangle}{\sqrt{2}}, \tag{4.7}$$

which will project to $|\Phi_{\mathrm{E}}^{-}\rangle$ if the middle parity qubit is measured to be 0. Note that the r_{12}^{x} and r_{23}^{x} operations commute since they involve different γ operators, making the operation order irrelevant (as well as allowing simultaneous operations). In general, each maximally entangled state can be prepared by measuring all three qubits to project to a single basis state, extending the outer topological regions towards the middle topological region for a small time, returning them to their original position, then projectively measuring the middle qubit.

Once the state $|\Phi_{\mathrm{E}}^{-}\rangle$ is prepared, one can test the version of Bell's inequality given in Section 1.3,

$$P_{=}(a,b) + P_{=}(b,c) + P_{=}(a,c) \geq 1, \tag{4.8}$$

where $P_{=}(L,R)$ is the probability that the left and right qubits are equal after being rotated by angles L and R, respectively. The left side of the inequality, which will be called the "Bell quantity", can be interpreted as the probability that at least one of the rotation combinations will make the left and right qubits equal.

According to quantum mechanics the probability that the qubits are equal after rotations L and R is $\cos^2\left(\frac{L-R}{2}\right)$. Only the relative angles between rotations are physically relevant, so one can set $A \equiv a - c$ and $B \equiv b - c$ to write the Bell quantity as

$$\cos^2\left(\frac{A-B}{2}\right) + \cos^2\left(\frac{A}{2}\right) + \cos^2\left(\frac{B}{2}\right), \tag{4.9}$$

which is plotted in Fig. 4.3. Quantum mechanics predicts the Bell quantity can be as low as 3/4 (for the relative angles $A = 2\pi/3$ and $B = 4\pi/3$, or vice-versa) and is inconsistent with local hidden variable theories, which require the Bell quantity to be greater than or equal to 1. In principle, Bell's inequality could be experimentally tested in our proposal by repeatedly preparing maximally entangled states, performing the three rotation combinations in Eq. (4.8), and measuring the qubits to find the probability of each state.

In practice however, almost every experiment that tests Bell's theorem uses the CHSH inequality discussed in Appendix 1.3,

$$|\langle L_1, R_1\rangle + \langle L_2, R_1\rangle + \langle L_1, R_2\rangle - \langle L_2, R_2\rangle| \leq 2, \tag{4.10}$$

where $\langle L, R\rangle = P_{=}(L,R) - P_{\neq}(L,R)$ is the expectation value of the combined parity of the left and right qubits after being rotated by angles L and R, respectively. The left

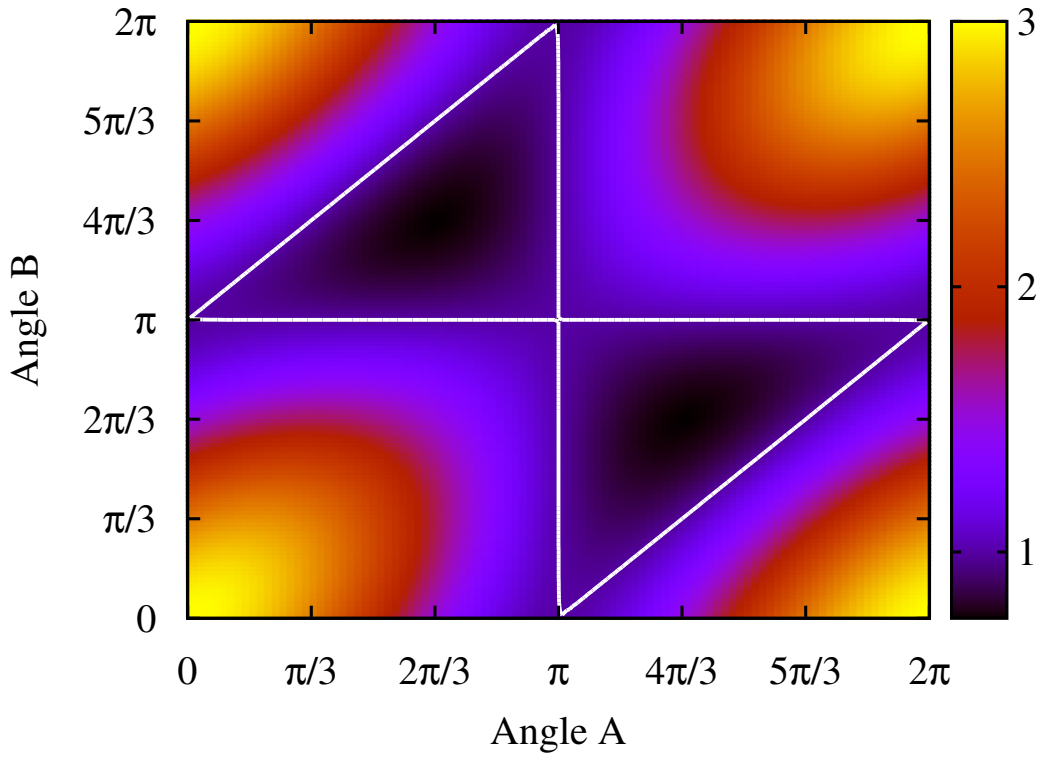

Figure 4.3: Contour plot for the quantum mechanical prediction of the Bell quantity for the state $|\Phi_E^-\rangle$. Local hidden variable theories require that the Bell quantity be greater than or equal to 1, but it is predicted to be less than 1 for relative rotation angles inside the white triangles.

side of the inequality, which will be called the "CHSH quantity", must be less than or equal to 2 in local hidden variable theories.

According to quantum mechanics, the expectation value discussed above for general rotation angles L and R is simply $\cos(L-R)$. Again, only the relative angles of rotation are physically significant, so we introduce angles $A \equiv L_1 - R_2$, $B \equiv R_1 - L_1$, and $C \equiv L_2 - R_1$, [see Fig. 4.4], making the CHSH quantity

$$|\cos(A) + \cos(B) + \cos(C) - \cos(A+B+C)|\,, \tag{4.11}$$

which has a maximum of $2\sqrt{2}$ when $A = B = C = \pi/4$, contradicting the local hidden variable prediction. The inequality can be tested experimentally by repeatedly preparing the state $|\Phi_{\mathrm{E}}^{-}\rangle$, extending the topological regions to perform one of the four rotation combinations involved in Eq. (4.10), then returning the topological regions to their original position to measure the qubits.

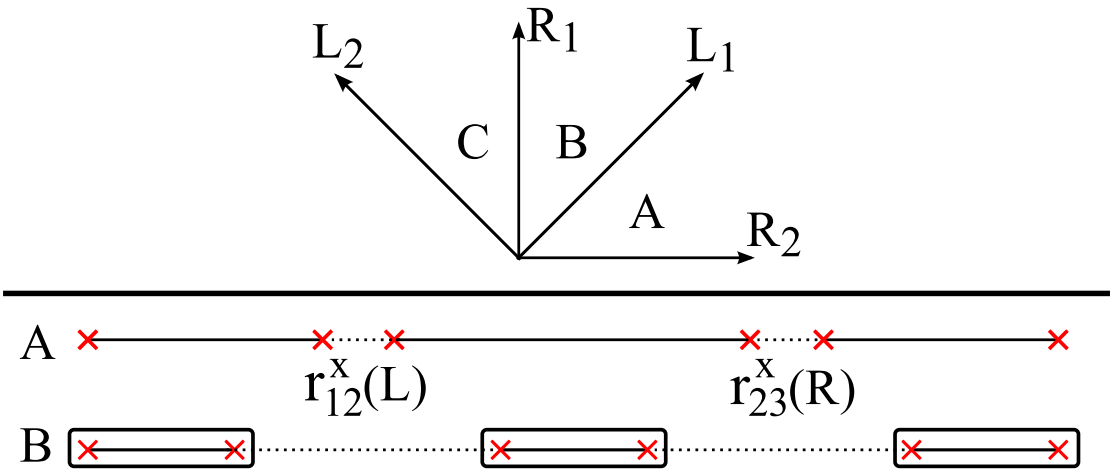

Figure 4.4: Top: Angles of rotation in CHSH inequality. The left qubit is rotated by either angle L_1 or L_2, while the right qubit is rotated by either angle R_1 or R_2. Bottom: A) One of the four rotation combinations is performed by extending the outer topological regions, B) then returned for measurement.

4.4 Semiconductor Hamiltonian and Simulation

Consider a more realistic model of the semiconductor system by re-introducing the one-dimensional lattice Hamiltonian

$$H_{\mathrm{S}} = H_{\mathrm{TB}} + H_{\mathrm{SO}} + H_{\mathrm{Z}} + H_{\mathrm{SC}}, \tag{4.12}$$

where

$$\begin{aligned}
H_{\mathrm{TB}} &= \sum_{j\sigma} \left[(2t_0 - \mu_j) \hat{c}^\dagger_{j\sigma} \hat{c}_{j\sigma} - t_0 \hat{c}^\dagger_{j\pm1,\sigma} \hat{c}_{j\sigma} \right], && (4.13)\\
H_{\mathrm{SO}} &= \sum_{j\sigma} \left[\alpha\, s(\sigma) \left(\hat{c}^\dagger_{j\bar{\sigma}} \hat{c}_{j+1,\sigma} - \hat{c}^\dagger_{j+1,\bar{\sigma}} \hat{c}_{j\sigma} \right) \right], && (4.14)\\
H_{\mathrm{Z}} &= \sum_{j\sigma} \left[s(\sigma) V^z \hat{c}^\dagger_{j\sigma} \hat{c}_{j\sigma} + V^{s(\sigma)} \hat{c}^\dagger_{j\bar{\sigma}} \hat{c}_{j\sigma} \right], && (4.15)\\
H_{\mathrm{SC}} &= \sum_{j} \left(\Delta \hat{c}^\dagger_{j\uparrow} \hat{c}^\dagger_{j\downarrow} + \Delta^* \hat{c}_{j\downarrow} \hat{c}_{j\uparrow} \right), && (4.16)
\end{aligned}$$

where all the terms are the same as defined in the uniform case of Sec. 3.3, except that the chemical potential, μ_i, is allowed to vary at each site.

When $\mu > \mu_{\mathrm{T}} \equiv \sqrt{V_\perp^2 - \Delta^2}$, where $V_\perp^2 = (V^z)^2 + (V^x)^2$ is the Zeeman field perpendicular to the spin-orbit quantization axis, the wire is a normal superconductor. In the other case, $\mu < \mu_{\mathrm{T}}$, a topologically distinct state emerges with Majorana bound states localized at the ends of the wire. If the chemical potential varies spatially and crosses the topological limit at multiple locations, then multiple Majorana bound states will be present and the setup discussed in Sec. 4.2 is possible.

Specifically, this proposal separates the wire into three topological regions, leading to six Majorana bound states, one at the end of each region, which are sufficiently separated to prevent them from fusing together. The Majorana bound states from each topological region can be paired together to form conventional fermions [e.g., $\hat{d}_n = (\hat{\gamma}_A + i\hat{\gamma}_B)/2$] that correspond to three zero-energy (in the limit of an infinite wire) Bogoliubov excitations, separated from the higher-energy bulk states by a topological

gap Δ_{T}[see Fig. 4.5]. Alternatively, these excitations can be thought of as the zero-energy eigenstate solutions to the Bogoliubov-de Gennes equations for the system. Just as in the simpler model, the occupation number of the eigenstates localized to the left and right serve as the logical qubits, while the occupation number of the middle eigenstate does not contain unique quantum information since the total parity is conserved.

The spatial distribution of these excitations is contained in the coefficients u and v from the Bogoliubov transformation

$$\hat{d}_n^\dagger = \sum_{j\sigma} (u_{j\sigma}^n \hat{c}_{j\sigma}^\dagger + v_{j\sigma}^n \hat{c}_{j\sigma}), \tag{4.17}$$

which can be used to form the parity operators of the wire segments, $\hat{P}_n \equiv 1 - 2\hat{d}_n^\dagger \hat{d}_n$.

The simulation begins by finding the coefficients for the lowest three eigenstates of the Hamiltonian in Eqs. (4.12)-(4.16) with parameters corresponding to $m^* = 0.015 m_{\mathrm{e}}$, $a = 15$nm leading to $t_0 = 11.3$meV, $g = 50$, $B = B^z = 0.625$T leading to $V_\perp = 0.9$meV, Rashba parameter $\alpha_{\mathrm{R}} = 225meV\cdot$Å corresponding to $\alpha \equiv \alpha_{\mathrm{R}}/a = 1.5$meV, and $\Delta = 0.5$meV. Thus the chemical potential marking the threshold between topological phases is $\mu_{\mathrm{T}} = 1.06$meV. The wire has 600 sites leading to length $l = 9\mu$m, with non-periodic boundary conditions. At the domain walls the chemical potential smoothly alternates between 0 and $2\mu_{\mathrm{T}}$ over a length of approximately $4\lambda = 0.04l$ with the profile function $\pm\mu_{\mathrm{T}} \tanh(x/\lambda)$, as shown in Fig. 4.5.

Each of the lowest three eigenstates has two peaks localized at the ends of the topological region, indicating the location of the Majorana bound states. The spatial distribution of the left and right bound states of each wire segment can be found numerically by considering the self-conjugate combinations of the three eigenstates, corresponding to $\hat{d}_n + \hat{d}_n^\dagger$ and $-i(\hat{d}_n - \hat{d}_n^\dagger)$, shown as solid and dashed curves in Fig. 4.5, respectively. Though the peaks decrease exponentially, their small, but non-zero over-

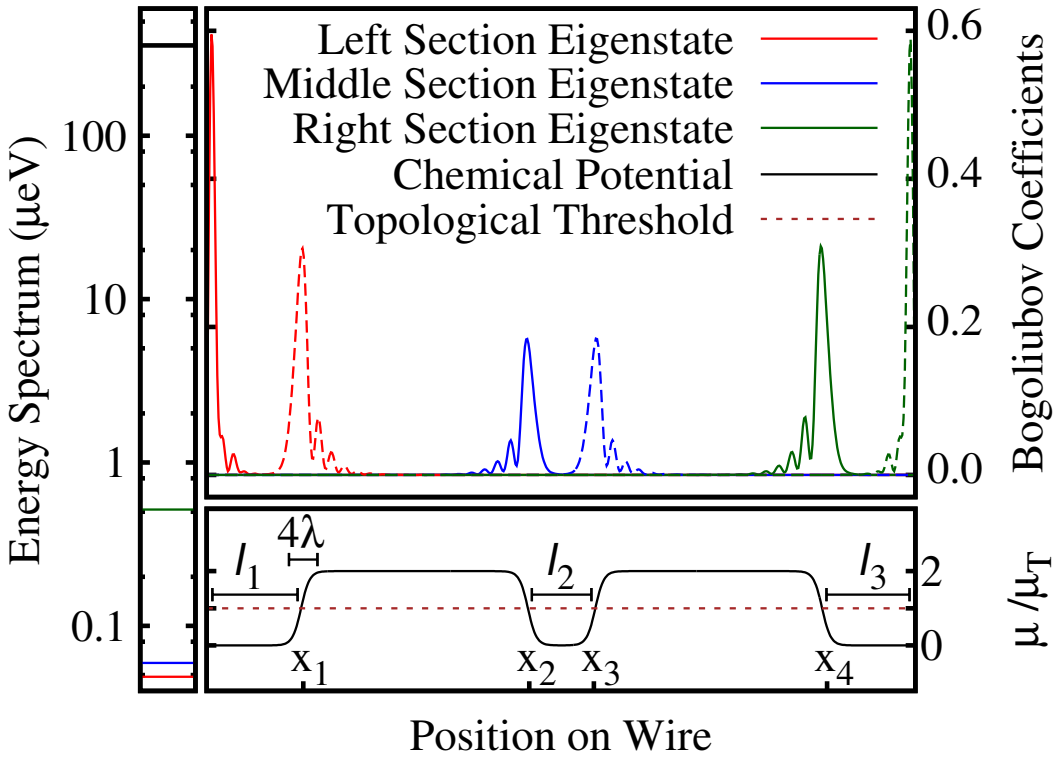

Figure 4.5: Bottom Right: A spatially varying chemical potential with three regions below the topological threshold of $\sqrt{V_\perp^2 - \Delta^2}$, with domain wall lengths of $\sim 4\lambda$. Top Right: This leads to six Majorana bound states, one at the end of each region, that form three conventional eigenstates. The simulated spatial distribution of the Bogoliubov coefficients $\sum_\sigma(|u_\sigma|^2+|v_\sigma|^2)$ along the length of the wire for the left and right bound states of each region s plotted in solid and dashed, respectively. Left: The energy spectrum of these eigenstates is plotted in log scale, as well as the lowest-energy bulk state separated by a topological gap of 360μeV. The splitting of the "zero"-energy states is due to the exponentially small overlap in peaks, which is larger for the shorter topological region of the middle segment. The topological region lengths are 13.3% and 9.5% of the wire length for the outer and middle regions, respectively.

laps cause the eigenstates to split from zero-energy. Thus the topological regions must be long enough to prevent these overlaps from splitting the excitations and coupling them to the bulk states. The lengths between the domain walls are set as $l_1 = x_1 = 0.133l$, $l_2 = x_3 - x_2 = 0.095l$, and $l_3 = l - x_4 = 0.133l$. These lengths were chosen to minimize the overlaps between the Majorana bound states of the same region, as well as the overlap between neighboring regions.

Once the chemical potential is tuned as described above, it can be varied dynamically to perform operations on the qubits. The only operation needed to test entanglement inequalities are r^x_{12} and r^x_{23}, which can be performed simultaneously by extending the outer topological regions towards the middle region. Specifically the domain wall positions x_i alternate back and forth according to the function

$$\pm \Lambda \left[\tanh\left(\frac{t}{\tau}\right) - \tanh\left(\frac{t-D}{\tau}\right) \right], \tag{4.18}$$

which smoothly shifts the domain walls 2Λ over an approximate transition time of 4τ for a duration D between the center of the two transitions as shown in Fig. 4.6.

One only needs to consider the dynamics of the zero-energy states, which won't mix with the bulk states above the topological gap as long as the domain wall trajectories are adiabatic. This constraint can be treated with the Landau-Zener condition [66, 122]: the rate the chemical potential changes must satisfy $\hbar|d\mu/dt| \ll 2\pi\Delta_{\mathrm{T}}^2$. To test this in the simulation, it finds the probability that the basis states of Eqs. (4.2) remain in the zero-energy subspace of the same total parity after evolution by using the following procedure.

The initial state $|\phi\rangle$ is assumed to be in the set of even parity basis states given in Eqs. (4.2), $S_{\mathrm{E}} = \{|000\rangle, |011\rangle, |110\rangle, |101\rangle\}$, where $|000\rangle$ is defined as the state such that $\hat{d}_n|000\rangle = 0$ for all n, including those corresponding to bulk states. The zero-energy

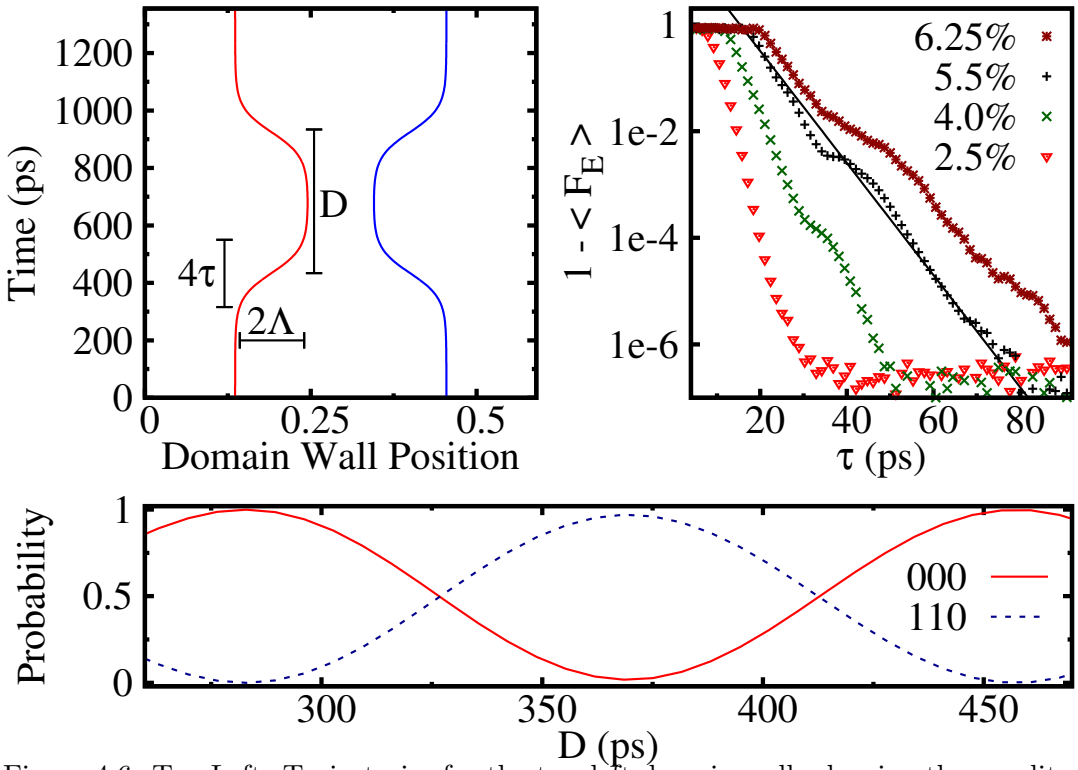

Figure 4.6: Top Left: Trajectories for the two left domain walls showing the amplitude, transition time, and duration for the $r^x_{12}(\pi/4)$ operation. Top Right: Average infidelity of even states after performing $r^x_{12}(\pi/4)$, plotted against transition time for various amplitudes (labeled as percentages of the wire length l) showing exponential behavior in general agreement with the Landau-Zener formula until limited by the Runge-Kutta step-size. The $\Lambda = 0.055l$ data is fit with a line that scales as $\exp(-\beta\tau)$ with $\beta = 240$ Ghz, reasonably close to the predicted value of 214 GHz. Bottom: Probabilities that the initial state $|000\rangle$ remains in $|000\rangle$ or transitions to $|110\rangle$ when acted on by r^x_{12} with various duration times. The simulated operation agrees well with the expected rotation, with a minimum probability of $\sim 0.3\%$ for the $|000\rangle$, due to the very small unintended overlaps of the bound states.

eigenstates are evolved using fourth-order Runge-Kutta in the eigenstate basis, possibly leaking into the bulk states if the transition time is too short, then acted on by the zero-energy eigenstate projector $\hat{P}_0$ to find the sub-matrix $\hat{U}_0 = \hat{P}_0\hat{U}\hat{P}_0$ of the full evolution matrix $\hat{U}$. Using $\hat{U}_0$, the time-evolved occupation operators in the Heisenberg picture are,

$$\hat{N}_n(t) = \hat{U}_0^\dagger(t)\hat{d}_n^\dagger(0)\hat{d}_n(0)\hat{U}_0(t), \tag{4.19}$$

for $n = 1, 2, 3$, which are bilinear combinations of the original $\hat{d}_n(0)$, including anomalous terms (e.g., $\hat{d}_1\hat{d}_2$) since the Hamiltonian contains superconductor pairing. The three $\hat{N}_n(t)$ are then used to form the projector for each multi-particle state in the three-qubit basis, allowing one to calculate the probability that the corresponding state is occupied. For example, the state $|110\rangle$ has the projector $\hat{N}_1\hat{N}_2(\hat{1} - \hat{N}_3)$, which yields 0 when acting on any other basis state. Since the projector's eigenvalue for $|110\rangle$ is 1, the expectation value is equal to the probability, and the probability that the initial state will be measured in the state $|110\rangle$ after time t is

$$P_{110}(\phi) = \langle\phi|\hat{N}_1(t)\hat{N}_2(t)[\hat{1} - \hat{N}_3(t)]|\phi\rangle, \tag{4.20}$$

which can be easily calculated for any $|\phi\rangle \in S_{\mathrm{E}}$. Similarly, the probabilities that $|\phi\rangle$ evolves into the other states in S_{E} are found using their respective multi-particle projectors, which are summed to give the fidelity from $|\phi\rangle$ to S_{E},

$$F_{\mathrm{E}}(\phi) = P_{000}(\phi) + P_{011}(\phi) + P_{110}(\phi) + P_{101}(\phi). \tag{4.21}$$

This is calculated for all $|\phi\rangle \in S_{\mathrm{E}}$ after simulating the operation $r_{12}^x(\pi/4)$, and plot the average infidelity for even states, $1 - \langle F_{\mathrm{E}}\rangle$, versus the transition time for various amplitudes in Fig. 4.6. The results are compared with the Landau-Zener formula[66, 122] by using the maximum of the chemical potential rate

$$\frac{d\mu}{dt} = \frac{\partial\mu}{\partial x_1}\frac{\partial x_1}{\partial t}, \tag{4.22}$$

which occurs halfway through the transitions when $\mathrm{sech}^2(0) = 1$, giving

$$\left(\frac{d\mu}{dt}\right)_{\max} = \frac{\mu_{\mathrm{T}}\Lambda}{\lambda\tau}. \tag{4.23}$$

Thus $1 - \langle F_{\mathrm{E}}\rangle$ should scale as $\exp(-\beta\tau)$ with

$$\beta = \frac{2\pi\lambda\Delta_{\mathrm{T}}^2}{\hbar\mu_{\mathrm{T}}\Lambda}, \tag{4.24}$$

in general agreement with the data. For example, the fitted line for $\Lambda = 0.055l$ in the logarithmic plot in Fig. 4.6 has a slope that corresponds to $\beta_{\rm fit} = 240$ GHz, while the value predicted from Eq. (4.24), using $\Delta_{\rm T} = 0.36$ meV found in the simulation, is $\beta \simeq 214$ GHz. All the amplitudes fit the expected exponential behavior reasonably well until limited by the Runge-Kutta step-size, with the exception of small plateaus that appear at different transition times for different amplitudes. This indicates that the coefficient for the average infidelity contains some amplitude-dependent factors, but these factors are insignificant compared to the exponential scaling and unimportant for this proposal. The adiabatic constraint is easily satisfied by proceeding with our simulation using $\Lambda \sim 0.06l$ and $\tau = 100$ ps.

In order to ensure the r^x_{12} operation is performed as expected, the probabilities for the basis states are found using the initial state $|000\rangle$ after the domain wall trajectory in Fig. 4.6 is simulated. As anticipated, the probabilities $P_{000}(000)$ and $P_{110}(000)$ oscillate, with negligible probabilities (on the order of our step-size limit of 10^{-6}) found in the states with incorrect total parity. However, the operation doesn't complete a full bit-flip for the duration expected to correspond to a π rotation, with $\sim 0.3\%$ of the probability found in the $|011\rangle$ and $|101\rangle$ instead of $|110\rangle$. This is consistent with a 0.3% shift of the rotation axis away from $\hat{x}$ due to small, undesired overlaps between Majorana bound states not involved in the operation. For example, a small overlap between the bound states of the middle and right regions would lead to an additional rotation r^x_{23} that effectively shifts the axis of rotation very slightly.

Using an amplitude $\Lambda = 0.06l$ for r^x_{12} and r^x_{23}, the simulated rotations have a period of ~ 0.2ns. Since the operations are achieved by bringing together exponentially decaying peaks, the overlap-dependent coupling between topological regions (e.g., Γ in the Majorana Hamiltonian) is exponentially sensitive to the trajectory amplitude. Thus,

longer rotation periods can be achieved by slightly decreasing the amplitude. On the other hand, greater amplitudes give shorter periods, but can also risk fusing the adjacent Majorana bound states if increased too much, which begins to occur in the simulation near $\Lambda \sim 0.07l$. Thus, the typical operation time (including adiabatic transitions) for the parameters is on the order of 0.4 ns.

Finally, the CHSH inequality is tested in the simulation by simultaneously performing the r^x_{12} and r^x_{23} rotations on the initial state $|\Phi^-_{\mathrm{E}}\rangle$ and finding the probabilities for each basis state. The CHSH quantity is a function of three relative angles, making it more difficult to visualize and compare to our simulation. Instead, consider one of the planes involving the maximum violation, namely when $R_2 = 0$ and $L_1 = \pi/4$ [i.e., $A = \pi/4$ in Eq. (4.11)]. The theoretical prediction and simulation are plotted in Fig. 4.7, showing very good agreement and a significant range of angles that violates the inequality. Thus the simulation indicates that the more realistic semiconductor Hamiltonian is consistent with the simpler Majorana model and the proposal is feasible for demonstrating entanglement in a Majorana wire. The code for this simulation is provided in Appendix B.

Before discussing experimental considerations, consider a simplification to the CHSH experiment that only requires projective measurement and the repeated use of two operations, namely $r^x_{12}(L)$ and $r^x_{23}(R)$ with specific values for L and R. Ideally $L = R = \pi/4$, but they remain unspecified here with the thought that the experiment could be attempted with angles that differ slightly from the ideal case.

The experiment begins by tuning the chemical potential to create three topological regions and measuring all of their occupation parities to project to one of the eight basis states. For concreteness, consider only the states that lead to the $|\Phi^-_{\mathrm{E}}\rangle$ state,

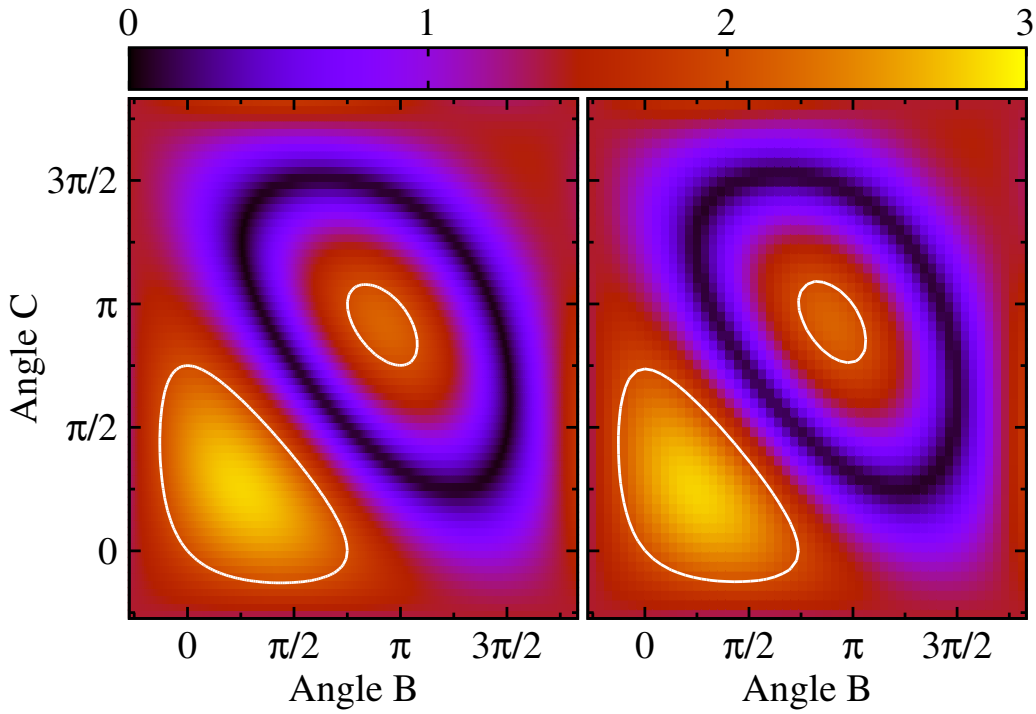

Figure 4.7: Left: Theoretical contour plot of the quantum mechanical prediction of the CHSH quantity for $|\Phi_{\mathrm{E}}^{-}\rangle$ for the $A = \pi/4$ plane. Local hidden variable theories are inconsistent with a CHSH quantity above 2, which occurs inside the white lines. Right: Simulated contour plot showing agreement with the global maximum at $B = C = \pi/4$ present, violating the CHSH inequality by approximately 40%.

so the procedure only continues if the measurement of the middle parity matches the total parity [see Table 4.1]. Alternatively, the full experiment is carried out regardless of the measurement outcomes, but the cases when the parities do not match are disregarded. Then the operations $r_{12}^{x}(L)$ and $r_{23}^{x}(R)$ are simultaneously performed twice before measuring the middle parity, proceeding only when this parity matches the initial result. For the ideal angles $L = R = \pi/4$, this procedure prepares maximally entangled states.

This preparation is followed by one of the four rotation combinations given in Eq. (4.10), with $L_1 = L$, $L_2 = 3L$, $R_1 = 0$, and $R_2 = 2R$. For example, the combination with L_2 and R_1 is performed by carrying out $r_{12}^{x}(L)$ three times but leaving the right domain walls stationary. After one of the rotation combinations is performed, all three

parities are measured and the results are recorded. This is repeated several times for each combination to find the corresponding probabilities and calculate the CHSH quantity in Eq. (4.10). The quantum mechanical prediction for the CHSH quantity using the above procedure is easily calculated (though not particularly illuminating in written form) and is plotted in Fig. 4.8. As expected, the CHSH quantity has a maximum at $L = R = \pi/4$, with a wide range of angles spanning from approximately $\pi/8$ to $3\pi/8$ confirming Bell's theorem. Thus this procedure can be used for a broad range of angles, demonstrating entanglement in Majorana wires, even with limited accuracy in the tuning operations.

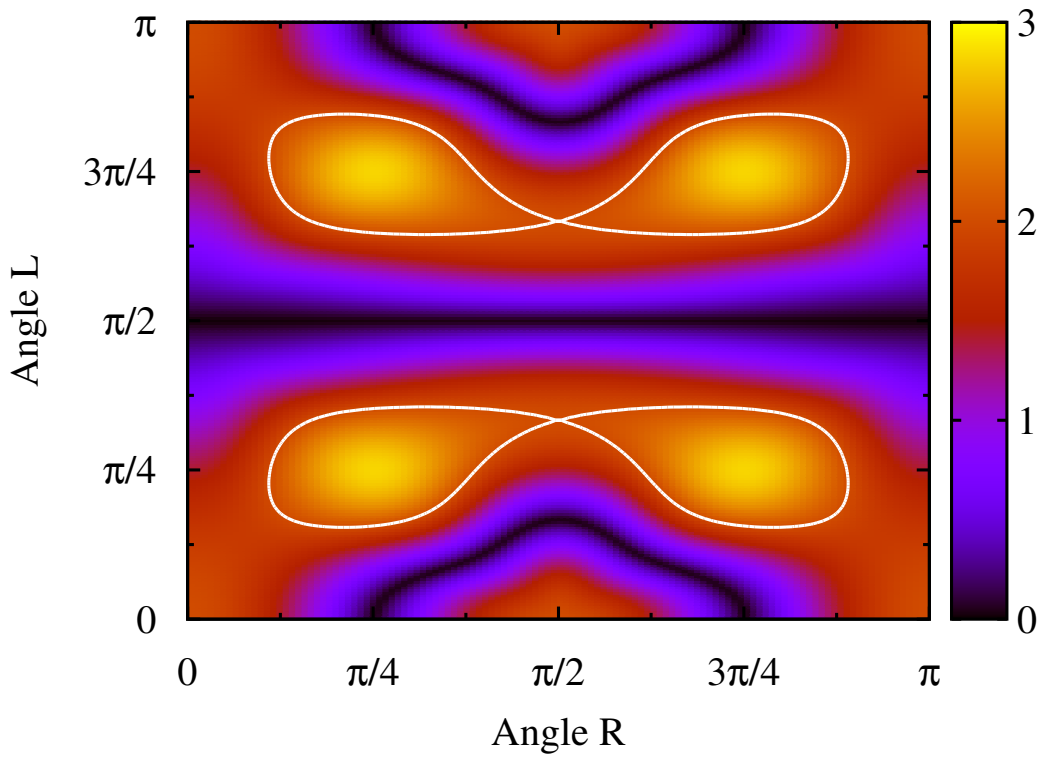

Figure 4.8: Contour plot of the quantum mechanical prediction of the CHSH quantity with $L_1 = L$, $L_2 = 3L$, $R_1 = 0$ and $R_2 = 2R$. Local hidden variable theories are inconsistent with a CHSH quantity above 2, which occurs inside the white lines. The plot repeats with a period of π for both L and R.

4.5 Experimental Considerations

This section discusses some aspects of this proposal that may be significant for an experimental realization. One of the first hurdles that must be overcome is the development of reliable projective measurement of the occupation. Aside from directly probing the wire with point contacts, there have been several proposals for observing the presence of Majorana bounds states such as using the Aharanov-Casher effect [50], transmons [49], and Shapiro step doubling in the AC-Josephson effect [99]. Without committing to a particular readout scheme, oe should note that this proposal requires measurement of a single topological region during the procedure in order to project to a maximally entangled state.

Another non-trivial experimental task is the fine-tuning of the chemical potential to minimize undesired Majorana peak overlaps. The simplest way to mitigate these overlaps is to use a longer wire, which exponentially reduces the overlaps. The simulation indicates that a wire length on the order of $5 - 10\mu$m is sufficient. Alternatively, a setup that links together several shorter wires may also be possible if longer wires are experimentally unavailable.

The adiabatic constraint that was found using the Landau-Zener formula is rather lenient, only requiring transition times on the order of 0.1ns. This is due to the generous topological gap of ~ 0.35 meV that separated the zero-energy and bulk states, due to the relatively large proximity effect and g-factor. In addition, the topological phase requires a relatively large spin-orbit coupling. The parameters are reasonable when compared to experiments [82, 24, 20, 36, 99], but the need for a robust topological gap should be considered when selecting materials, and further advances of the proximity effect and spin-orbit in relevant materials would be helpful.

The operation time for performing the ideal rotation angles can be found experimentally by reproducing the probability plot in Fig. 4.6. Indeed, calibration is needed to find the effective coupling that accounts for additional effects such as disorder, local tunings, and the small coupling between regions via the s-wave superconductor [124]. For example, the ideal $r_{12}^x(\pi/4)$ for the simplified CHSH experiment can be calibrated in the following way. First, all three parities are measured to project to a basis state, then gates are gradually tuned to shift the chemical potential in the left non-topological region for an operation time $\sim$ 0.5ns, followed by a final measurement of all three parities. Note that the operations can be achieved by moving a single domain wall if this is easier experimentally; the simulation moved both the outer and middle domain walls to suppress overlaps in the topological regions, but this may be unnecessary in longer wires. This procedure is repeated several times with the same operation time, tracking the percentage of times the state remains in the initially measured state. This is repeated with several slightly different operation times, until the percentage is near $\cos^2(\pi/8) \simeq 0.85$. Once the rotations that correspond to $L = R = \pi/4$ are roughly calibrated, the simplified CHSH experiment can be carried out.

As noted by Rainis et. al. [93], one must also consider the phenomena of quasi-particle poisoning in any system that uses superconductors to achieve the topological phase. While a superconductor at $T = 0$ (which our simulation assumed) will only form Cooper pairs, at finite temperatures less than $\sim$ 160mK a small, fixed population of quasi-particles seems to be present [23]. Quasi-particle poisoning occurs when a single quasi-particle tunnels between the superconductor and semiconductor, changing the total parity of the system and destroying the quantum information. Thus, both the measurement and operation times must be much shorter than the average time of quasi-particle tunneling, constraining the operation time in the opposite limit as the adiabatic

condition. Fortunately, estimates of the average time for quasi-particle tunneling in Majorana wire systems are 100ns or greater in typical experimental systems, depending on the specific parameters [93]. These calculations depend linearly on the resistance of the wire; thus longer wires will have shorter tunneling times. Nevertheless, with the adiabatic constraint only requiring operation times ~ 0.1ns, there is a wide window for the few operations needed in this proposal. Thus the constraint on the operation time of measurements is largely unaffected by the proposed operations.

Another non-trivial aspect inherent to the entanglement inequalities is the need to find probabilities rather than single measurements outcomes, requiring a high level of precision in the gate tuning. While this may make it difficult to reproduce the exact predictions of quantum mechanics, the large violation of the CHSH inequality by $> 40\%$, and the wide range of angles that violate hidden variable theories may still be sufficient for demonstrating entanglement.

One way to circumvent the precision requirement is to perform the test proposed by Greenberger, Horne, Zeilinger [45] and Mermin [78] (GHZM). The GHZM experiment requires three logical qubits (thus four topological regions), but tests hidden variable theories with a single measurement, rather than involving probabilities and inequalities. Indeed, there are many interesting experiments that demonstrate entanglement, such as quantum teleportation, that are possible with one additional qubit.

This and other relevant systems are still a new and emerging area of research for theorists and experimentalists alike. With that said, there are many recent developments which are not reflected in the above Hamiltonian. For example, it seems that the experiments on the Majorana wire systems are not strictly one-dimensional, and must be analyzed as multi-channel wires to explain some of the experimental findings [110]. Others note that phenomena like Andreev reflection, disorder, finite temperature and

the Kondo effect may need to be further understood in these systems [110, 71, 94]. Despite these complications, one should note that almost any convincing manifestation of Majorana bound states must demonstrate entanglement, which will likely be easier than full braiding. While this proposal discussed the specific setup of Majorana wires, the general idea of using non-topological, proximity-induced operations to test the entanglement inequalities as a stepping stone to braiding operations, as well as other aspects like separating logical qubits with a parity qubit, may be applied to many systems that potentially support Majorana bound states, such as topological insulators [39, 85, 40].

4.6 General Preparation Procedure

Sec. 4.3 discussed the Bell and CHSH inequalities with the state $|\Phi_{\mathrm{E}}^{-}\rangle$. This section considers more general procedures for preparing any maximally entangled state and testing the inequalities.

Entanglement can be demonstrated for any initial condition with very simple alterations to the proposal, rather than discarding data for the incorrect initial state or measurement outcome.

The procedure shown in Fig. 4.2 prepares one of the eight maximally entangled states,

$$|\Phi_{\mathrm{E}}^{\pm}\rangle = \frac{|000\rangle \pm |101\rangle}{\sqrt{2}}, \qquad |\Psi_{\mathrm{E}}^{\pm}\rangle = \frac{|011\rangle \pm |110\rangle}{\sqrt{2}}, \tag{4.25}$$

$$|\Phi_{\mathrm{O}}^{\pm}\rangle = \frac{|010\rangle \pm |111\rangle}{\sqrt{2}}, \qquad |\Psi_{\mathrm{O}}^{\pm}\rangle = \frac{|001\rangle \pm |100\rangle}{\sqrt{2}}, \tag{4.26}$$

by measuring all three parities to project to one basis state from Eqs. (4.2), performing the operations $r_{12}^{x}(\pi/2)$ and $r_{23}^{x}(\pi/2)$, then measuring the middle parity. The state that is prepared depends on the overall parity and middle parity measurements, as shown in

Table 4.1. Note that the results for the even and odd total parity are equivalent upon the exchange $0 \leftrightarrow 1$ for the middle parity qubit.

Table 4.1: Maximally entangled state prepared for various total parity and middle parity measurements. The even and odd total parities give the same results if upon exchanging $0 \leftrightarrow 1$ for the middle parity.

Total Parity	Even				Odd			
Initial Middle Parity	0		1		1		0	
Final Middle Parity	0	1	0	1	1	0	1	0
Resulting State	Φ_{E}^{-}	Ψ_{E}^{+}	Φ_{E}^{+}	Ψ_{E}^{-}	Φ_{O}^{-}	Ψ_{O}^{+}	Φ_{O}^{+}	Ψ_{O}^{-}

Any of the maximally entangled states can be used to demonstrate the violation of the Bell and CHSH inequalities, but with different rotation angles. For example, quantum mechanics predicts that $\langle L, R\rangle$, the expectation value of the combined parity of the left and right qubits after being rotated by angles L and R, respectively, for Φ_{E}^{+} is $\cos(L+R)$ rather than $\cos(L-R)$ for Φ_{E}^{-}. Clearly the CHSH quantity in Eq. (4.10) is the same except with $R \to -R$, which can be returned to the case in Sec. 4.3 by substituting $\{L_1, L_2, R_1, R_2\} \to \{L_1, L_2, -R_1, -R_2\}$. The relevant probabilities and angle transformations for the Bell and CHSH inequalities are listed in Table 4.2 for each maximally entangled state. These changes can be accounted for by designing the experiment to perform different rotations depending on the middle parity measurement outcomes found during the preparation of the maximally entangled states.

Table 4.2: Probabilities and expectation values predicted by quantum mechanics for the various maximally entangled states. The set of angles that corresponds to the case in the body of the paper for the CHSH violation is given as well. The results are the same for even and odd parity, so we suppress the corresponding subscript.

State	$P_{=}(L,R)$	$\langle L, R\rangle$	CHSH Angles
Φ^{-}	$\cos^2\left(\frac{L-R}{2}\right)$	$\cos(L-R)$	L_1, L_2, R_1, R_2
Φ^{+}	$\cos^2\left(\frac{L+R}{2}\right)$	$\cos(L+R)$	$L_1, L_2, -R_1, -R_2$
Ψ^{-}	$\sin^2\left(\frac{L-R}{2}\right)$	$-\cos(L-R)$	L_1, L_2, R_1, R_2
Ψ^{+}	$\sin^2\left(\frac{L+R}{2}\right)$	$-\cos(L+R)$	$L_1, L_2, -R_1, -R_2$

4.7 Majorana Wire Proposal Summary

The motion of domain walls in a Majorana wire system was analyzed with three separate topological regions using a simple Hamiltonian analogous to Kitaev's toy model [61]. Using the occupation number parity of the Majorana bound states in each region, a three qubit basis was defined with two logical qubits and one qubit forfeited to total parity conservation. In this basis, x-axis rotations are performed by extending the outer topological regions to isolate a single coupling between different topological regions. While these rotations are not topologically protected from local perturbations, they can demonstrate entanglement by testing the Bell and CHSH inequalities.

With the simpler model as a guide, the domain wall motion was simulated using a more realistic semiconductor Hamiltonian. Results indicate that the topological regions can be well separated in wires of length $\sim 5 - 10\,\mu$m with reasonable parameters compared to recent experiments. Adiabatic changes in the chemical potential were simulated with operation times on the order of 0.5ns, consistent with the Landau-Zener condition applied to excitations from zero-energy to the bulk. Extending the topological regions results in the rotations predicted by the simpler model. Finally, the CHSH experiment was simulated and the expected inconsistency with hidden variable theories predicted by Bell's theorem was found, indicating that the simpler Majorana Hamiltonian approximates the Majorana wire system well.

A simplified version of the CHSH experiment was introduced that only requires projective measurement and repeated use of two $\pi/4$ rotations, finding a wide range of operation angles from $\pi/8$ to $3\pi/8$ violate the entanglement inequalities. Thus a keyboard gating setup needs to be relatively precise, but moderate inaccuracy is tolerable. Calibrating methods were provided for the rotations and potential hurdles for an ex-

perimental realization were discussed. The analysis and simulation indicate that there is a large window of operation times, spanning three orders of magnitude, that satisfy the adiabatic and quasi-particle poisoning constraints. Hence this proposal is viable for demonstrating entanglement in Majorana wires if methods for projective measurement and precise gate tuning are available.

While there have been proposals to test entanglement [57, 17, 100, 6, 67], observing entanglement by directly testing Bell's theorem would be a novel opportunity in solid state systems in general, and an important advance for Majorana wire systems specifically. In addition to being a crucial ingredient for quantum information, entanglement may rule out alternative explanations inconsistent with non-local effects, providing a useful tool beyond local measurements. More so, experimental work on gate tuning is already required for braiding operations, and this proposal could serve as a useful benchmark towards that goal. Thus, the observation of entanglement would support current models of Majorana wires and provide a significant piece of evidence supporting the presence of Majorana bound states.

Chapter 5

Conclusion

This dissertation considered two proposals for fault-tolerant qubits: pairs of electron spins in semiconductor double quantum dots and Majorana zero modes bound to the ends of semiconductor nanowires coupled to conventional superconductors. Analysis and simulation of these systems showed that both proposals are viable and could enable further advances in the construction of quantum information systems.

In particular, the additional spin-orbit effects that result from the motion of double quantum dots do not outweigh the benefits of suppressing the hyperfine dephasing. Thus it is possible to incorporate adiabatic motion into quantum dot proposals, providing another avenue for decreasing the infidelity of quantum operations closer to the quantum error correction threshold. Similarly, the adiabatic motion of Majorana bound states in nanowires can be used to perform quantum operations within the constraints of quasiparticle poisoning. Aside from the potential for braiding operations, these operations are important since they allow one to both setup and demonstrate entanglement, a crucial ingredient to any quantum information system.

Rather than depending solely on experimental advances, these proposals aim to prevent decoherence by building on qubit schemes that are fundamentally immune to certain local errors. Furthermore, these proposals are constructed from conventional materials and are within reach of current experiments. While these fault-tolerant designs require additional complications such as adiabatic motion and novel particles, the included analysis and simulations indicate that these added complexities are not prohibitive. In summary, the use of adiabatic motion to implement and operate on qubits is a viable method for getting closer to the ambitious goal of realizing a quantum computer.

Bibliography

[1] J. Alicea, Y. Oreg, G. Refael, F. von Oppen, and M. P. A. Fisher. Non-abelian statistics and topological quantum information processing in 1d wire networks. *Nat. Phys.*, 7:412–417, May 2011.

[2] Jason Alicea. Majorana fermions in a tunable semiconductor device. *Phys. Rev. B*, 81:125318, Mar 2010.

[3] David D. Awschalom, Lee C. Bassett, Andrew S. Dzurak, Evelyn L. Hu, and Jason R. Petta. Quantum spintronics: Engineering and manipulating atom-like spins in semiconductors. *Science*, 339(6124):1174–1179, 2013.

[4] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O/'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and John M. Martinis. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014.

[5] C. Barthel, J. Medford, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Interlaced dynamical decoupling and coherent operation of a singlet-triplet qubit. *Phys. Rev. Lett.*, 105(26):266808, Dec 2010.

[6] C. W. J. Beenakker, C. Emary, M. Kindermann, and J. L. van Velsen. Proposal for production and detection of entangled electron-hole pairs in a degenerate electron gas. *Phys. Rev. Lett.*, 91:147901, Oct 2003.

[7] John S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1:195–200, 1964.

[8] P. Benioff. Quantum mechanical hamiltonian models of turing machines. *Journal of Statistical Physics*, 29:515–546, nov 1982.

[9] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of International Conference on Computers, Systems and Signal Processing*, volume 175, pages 175–179, 1984.

[10] Charles H. Bennett and Stephen J. Wiesner. Communication via one- and two-particle operators on einstein-podolsky-rosen states. *Phys. Rev. Lett.*, 69:2881–2884, Nov 1992.

[11] Sergey Bravyi. Universal quantum computation with the v=5/2 fractional quantum hall state. *Phys. Rev. A*, 73:042313, Apr 2006.

[12] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005.

[13] G K Brennen, S Iblisdir, J K Pachos, and J K Slingerland. Non-locality of non-abelian anyons. *New Journal of Physics*, 11(10):103023, 2009.

[14] Guido Burkard, Daniel Loss, and David P. DiVincenzo. Coupled quantum dots as quantum gates. *Phys. Rev. B*, 59(3):2070–2078, Jan 1999.

[15] F. J. Burnell, Alexander Shnirman, and Yuval Oreg. Measuring fermion parity correlations and relaxation rates in one-dimensional topological superconducting wires. *Phys. Rev. B*, 88:224507, Dec 2013.

[16] E. T. Campbell, M. J. Hoban, and J. Eisert. Majorana fermions and non-locality. *ArXiv e-prints*, May 2013.

[17] Nikolai M. Chtchelkatchev, Gianni Blatter, Gordey B. Lesovik, and Thierry Martin. Bell inequalities and entanglement in solid-state devices. *Phys. Rev. B*, 66:161320, Oct 2002.

[18] H. O. H. Churchill, V. Fatemi, K. Grove-Rasmussen, M. T. Deng, P. Caroff, H. Q. Xu, and C. M. Marcus. Superconductor-nanowire devices from tunneling to the multichannel regime: Zero-bias oscillations and magnetoconductance crossover. *Phys. Rev. B*, 87:241401, Jun 2013.

[19] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[20] Anindya Das, Yuval Ronen, Yonatan Most, Yuval Oreg, Moty Heiblum, and Hadas Shtrikman. Zero-bias peaks and splitting in an AlInAs nanowire topological superconductor as a signature of Majorana fermions. *Nat. Phys.*, 8:887–895, 2012.

[21] S. Das Sarma, Jay D. Sau, and Tudor D. Stanescu. Splitting of the zero-bias conductance peak as smoking gun evidence for the existence of the majorana mode in a superconductor-semiconductor nanowire. *Phys. Rev. B*, 86:220506, Dec 2012.

[22] C. M. Dawson and M. A. Nielsen. The solovay-kitaev algorithm. *eprint arXiv:quant-ph/0505030*, May 2005.

[23] P. J. de Visser, J. J. A. Baselmans, P. Diener, S. J. C. Yates, A. Endo, and T. M. Klapwijk. Number fluctuations of sparse quasiparticles in a superconductor. *Phys. Rev. Lett.*, 106:167004, Apr 2011.

[24] M. T. Deng, C. L. Yu, G. Y. Huang, M. Larsson, P. Caroff, and H. Q. Xu. Anomalous zero-bias conductance peak in a nb-insb nanowire-nb hybrid device. *Nano Lett.*, 12(12):6414–6419, 2012.

[25] D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Royal Society of London Proceedings Series A*, 400:97–117, July 1985.

[26] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 425(1868):73–90, 1989.

[27] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Royal Society of London Proceedings Series A*, 439:553–558, December 1992.

[28] M. H. Devoret and R. J. Schoelkopf. Superconducting circuits for quantum information: An outlook. *Science*, 339(6124):1169–1174, 2013.

[29] D. Dieks. Communication by {EPR} devices. *Physics Letters A*, 92(6):271 – 272, 1982.

[30] Y. Doi, T. Makino, H. Kato, D. Takeuchi, M. Ogura, H. Okushi, H. Morishita, T. Tashima, S. Miwa, S. Yamasaki, P. Neumann, J. Wrachtrup, Y. Suzuki, and N. Mizuochi. Deterministic electrical charge-state initialization of single nitrogen-vacancy center in diamond. *Phys. Rev. X*, 4:011057, Mar 2014.

[31] D. E. Drummond, A. A. Kovalev, C.-Y. Hou, K. Shtengel, and L. P. Pryadko. Demonstrating entanglement by testing bell's theorem in majorana wires. *ArXiv e-prints*, March 2014.

[32] David Drummond, Leonid P. Pryadko, and Kirill Shtengel. Suppression of hyperfine dephasing by spatial exchange of double quantum dots. *Phys. Rev. B*, 86:245307, Dec 2012.

[33] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[34] Artur K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.

[35] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.

[36] A. D. K. Finck, D. J. Van Harlingen, P. K. Mohseni, K. Jung, and X. Li. Anomalous modulation of a zero-bias peak in a hybrid nanowire-superconductor device. *Phys. Rev. Lett.*, 110:126406, Mar 2013.

[37] Marcel Franz. Majorana's wires. *Nat Nano*, 8(3):149–152, Mar 2013.

[38] M. H. Freedman, M. Larsen, and Z. Wang. Modular functor which is universal for quantum computation. *Communications in Mathematical Physics*, 227:605–622, 2002.

[39] L. Fu and C. Kane. Superconducting proximity effect and majorana fermions at the surface of a topological insulator. *Phys. Rev. Lett.*, 100:096407–, 2008.

[40] Liang Fu and C. L. Kane. Josephson current and noise at a superconductor/quantum-spin-hall-insulator/superconductor junction. *Phys. Rev. B*, 79:161408, Apr 2009.

[41] Alexei Gilchrist, Nathan K. Langford, and Michael A. Nielsen. Distance measures to compare real and ideal quantum processes. *Phys. Rev. A*, 71:062310, Jun 2005.

[42] Vitaly N. Golovach, Massoud Borhani, and Daniel Loss. Holonomic quantum computation with electron spins in quantum dots. *Phys. Rev. A*, 81(2):022315, Feb 2010.

[43] Vitaly N. Golovach, Alexander Khaetskii, and Daniel Loss. Phonon-induced decay of the electron spin in quantum dots. *Phys. Rev. Lett.*, 93:016601, 2004.

[44] D. Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.

[45] Daniel M. Greenberger, Michael A. Horne, Abner Shimony, and Anton Zeilinger. Bell's theorem without inequalities. *American Journal of Physics*, 58(12):1131–1143, 1990.

[46] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.

[47] M. Gullans, J. J. Krich, J. M. Taylor, H. Bluhm, B. I. Halperin, C. M. Marcus, M. Stopa, A. Yacoby, and M. D. Lukin. Dynamic nuclear polarization in double quantum dots. *Phys. Rev. Lett.*, 104(22):226807, Jun 2010.

[48] M. Z. Hasan and C. L. Kane. Colloquium. *Rev. Mod. Phys.*, 82:3045–3067, Nov 2010.

[49] F Hassler, A R Akhmerov, and C W J Beenakker. The top-transmon: A hybrid superconducting qubit for parity-protected quantum computation. *New J. Phys.*, 13(9):095004, 2011.

[50] F. Hassler, A. R. Akhmerov, C.-Y. Hou, and C. W. J. Beenakker. Anyonic interferometry without anyons: How a flux qubit can read out a topological qubit. *New J. Phys.*, 12:125002, 2010.

[51] Alexander S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problems of Information Transmission*, 9(3):177–183, 1973.

[52] A. Imamoğlu, E. Knill, L. Tian, and P. Zoller. Optical pumping of quantum-dot nuclear spins. *Phys. Rev. Lett.*, 91(1):017402, Jul 2003.

[53] Roman S. Ingarden. Quantum information theory. *Reports on Mathematical Physics*, 10(1):43 – 72, 1976.

[54] D. A. Ivanov. Non-Abelian statistics of half-quantum vortices in p-wave superconductors. *Phys. Rev. Lett.*, 86:268–271, 2001.

[55] Richard Jozsa and Noah Linden. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 459(2036):2011–2032, 2003.

[56] C. L. Kane and E. J. Mele. Z2 topological order and the quantum spin hall effect. *Phys. Rev. Lett.*, 95:146802, Sep 2005.

[57] S. Kawabata. Test of bell's inequality using the spin filter effect in ferromagnetic semiconductor microstructures. *Journal of the Physical Society of Japan*, 70:1210, May 2001.

[58] Wai-Yee Keung and Goran Senjanovic. Majorana neutrinos and the production of the right-handed charged gauge boson. *Phys. Rev. Lett.*, 50:1427–1430, May 1983.

[59] Alexander Khaetskii, Daniel Loss, and Leonid Glazman. Electron spin evolution induced by interaction with nuclei in a quantum dot. *Phys. Rev. B*, 67(19):195329, May 2003.

[60] K. Khodjasteh and D. A. Lidar. Fault-tolerant quantum dynamical decoupling. *Phys. Rev. Lett.*, 95:180501, 2005.

[61] A Yu Kitaev. Unpaired majorana fermions in quantum wires. *Physics-Uspekhi*, 44(10S):131, 2001.

[62] D. Klauser, W. A. Coish, and Daniel Loss. Nuclear spin state narrowing via gate-controlled rabi oscillations in a double quantum dot. *Phys. Rev. B*, 73(20):205302, 2006.

[63] Leo Kouwenhoven and Charles Marcus. Quantum dots. 1998.

[64] Jacob J. Krich and Bertrand I. Halperin. Cubic dresselhaus spin-orbit coupling in 2d electron quantum dots. *Phys. Rev. Lett.*, 98:226802, May 2007.

[65] E. A. Laird, J. M. Taylor, D. P. DiVincenzo, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Coherent spin manipulation in an exchange-only qubit. *Phys. Rev. B*, 82(7):075403, Aug 2010.

[66] Lev Landau. Zur theorie der energieubertragung. ii. *Physikalische Zeitschrift der Sowjetunion*, 2:46–51, 1932.

[67] A. V. Lebedev, G. B. Lesovik, and G. Blatter. Entanglement in a noninteracting mesoscopic structure. *Phys. Rev. B*, 71:045306, Jan 2005.

[68] Eduardo J. H. Lee, Xiaocheng Jiang, Ramón Aguado, Georgios Katsaros, Charles M. Lieber, and Silvano De Franceschi. Zero-bias anomaly in a nanowire quantum dot coupled to superconductors. *Phys. Rev. Lett.*, 109:186802, Oct 2012.

[69] Jeremy Levy. Universal quantum computation with spin-1/2 pairs and Heisenberg exchange. *Phys. Rev. Lett.*, 89:147902, 2002.

[70] Elliott Lieb, Theodore Schultz, and Daniel Mattis. Two soluble models of an antiferromagnetic chain. *Annals of Physics*, 16(3):407–466, 1961.

[71] Jie Liu, Andrew C. Potter, K. T. Law, and Patrick A. Lee. Zero-bias peaks in the tunneling conductance of spin-orbit-coupled superconducting wires with and without majorana end-states. *Phys. Rev. Lett.*, 109:267002, Dec 2012.

[72] Jie Liu, Fu-Chun Zhang, and K. T. Law. Majorana fermion induced nonlocal current correlations in spin-orbit coupled superconducting wires. *Phys. Rev. B*, 88:064509, Aug 2013.

[73] Roman M. Lutchyn, Jay D. Sau, and S. Das Sarma. Majorana fermions and a topological phase transition in semiconductor-superconductor heterostructures. *Phys. Rev. Lett.*, 105:077001, Aug 2010.

[74] E. Majorana. Teoria simmetrica dellelettronee del positrone. *Nuovo Cimento*, 14:171–, 1937.

[75] Yuri Manin. *Computable and Non-computable (in Russian).* Sovetskoye Radio, Moscow, 1980.

[76] M.P. Marder. *Condensed matter physics.* Wiley-Interscience. John Wiley, 2000.

[77] IA Merkulov, A.L. Efros, and M. Rosen. Electron spin relaxation by nuclei in semiconductor quantum dots. *Phys. Rev. B*, 65(20):205309, 2002.

[78] N. David Mermin. Quantum mysteries revisited. *American Journal of Physics*, 58(8):731–734, 1990.

[79] N.D. Mermin. *Quantum Computer Science: An Introduction.* Cambridge University Press, 2007.

[80] C. Monroe and J. Kim. Scaling the ion trap quantum processor. *Science*, 339(6124):1164–1169, 2013.

[81] Gregory Moore and Nicholas Read. Nonabelions in the fractional quantum hall effect. *Nuclear Physics B*, 360(2?3):362 – 396, 1991.

[82] V. Mourik, K. Zuo, S. M. Frolov, S. R. Plissard, E. P. A. M. Bakkers, and L. P. Kouwenhoven. Signatures of majorana fermions in hybrid superconductor-semiconductor nanowire devices. *Science*, 336(6084):1003–1007, 2012.

[83] Chetan Nayak, Steven H. Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Rev. Mod. Phys.*, 80:1083–1159, Sep 2008.

[84] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information.* Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000.

[85] Johan Nilsson, A. R. Akhmerov, and C. W. J. Beenakker. Splitting of a cooper pair by a pair of majorana bound states. *Phys. Rev. Lett.*, 101:120403, Sep 2008.

[86] Yuval Oreg, Gil Refael, and Felix von Oppen. Helical liquids and Majorana bound states in quantum wires. *Phys. Rev. Lett.*, 105:177002, Oct 2010.

[87] J. R. Petta, A. C. Johnson, J. M. Taylor, E. A. Laird, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Coherent manipulation of coupled electron spins in semiconductor quantum dots. *Science*, 309:2180, 2005.

[88] R.P. Poplavskii. Thermodynamical models of information processing (in russian). *Uspekhi Fizicheskikh Nauk*, 115(3):465–501, 1975.

[89] J. Preskill. Fault-tolerant quantum computation. *eprint arXiv:quant-ph/9712048*, December 1997.

[90] J. Preskill. Lecture notes for ph219/cs219: Quantum information and computation. `http://www.theory.caltech.edu/~preskill/ph219/index.html`, 2001.

[91] Leonid P. Pryadko and Gregory Quiroz. Soft-pulse dynamical decoupling with markovian decoherence. *Phys. Rev. A*, 80:042317, Oct 2009.

[92] Leonid P. Pryadko and Pinaki Sengupta. Quantum kinetics of an open system in the presence of periodic refocusing fields. *Phys. Rev. B*, 73:085321, Feb 2006.

[93] Diego Rainis and Daniel Loss. Majorana qubit decoherence by quasiparticle poisoning. *Phys. Rev. B*, 85:174533, May 2012.

[94] Diego Rainis, Luka Trifunovic, Jelena Klinovaja, and Daniel Loss. Towards a realistic transport modeling in a superconducting nanowire with majorana fermions. *Phys. Rev. B*, 87:024515, Jan 2013.

[95] N. Read and Dmitry Green. Paired states of fermions in two dimensions with breaking of parity and time-reversal symmetries and the fractional quantum hall effect. *Phys. Rev. B*, 61:10267–10297, Apr 2000.

[96] N. Read and E. Rezayi. Quasiholes and fermionic zero modes of paired fractional quantum hall states: The mechanism for non-abelian statistics. *Phys. Rev. B*, 54:16864–16887, Dec 1996.

[97] D. J. Reilly, J. M. Taylor, J. R. Petta, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Suppressing spin qubit dephasing by nuclear state preparation. *Science*, 321:817–821, 2008.

[98] Werner Rodejohann. Neutrino-less double beta decay and particle physics. *International Journal of Modern Physics E*, 20(09):1833–1930, 2011.

[99] Leonid P. Rokhinson, Xinyu Liu, and Jacek K. Furdyna. The fractional a.c. josephson effect in a semiconductor-superconductor nanowire as a signature of majorana particles. *Nat. Phys.*, 8(11):795–799, November 2012.

[100] P. Samuelsson, E. V. Sukhorukov, and M. Büttiker. Orbital entanglement and violation of bell inequalities in mesoscopic conductors. *Phys. Rev. Lett.*, 91:157002, Oct 2003.

[101] Jay D. Sau, Erez Berg, and Bertrand I. Halperin. On the possibility of the fractional ac josephson effect in non-topological conventional superconductor-normal-superconductor junctions. *arXiv:1206.4596*, 2012.

[102] Jay D. Sau, Roman M. Lutchyn, Sumanta Tewari, and S. Das Sarma. Generic new platform for topological quantum computation using semiconductor heterostructures. *Phys. Rev. Lett.*, 104:040502, Jan 2010.

[103] Jay D. Sau, Brian Swingle, and Sumanta Tewari. A proposal to probe quantum non-locality of majorana fermions in tunneling experiments. *arXiv:1210.5514*, 2013.

[104] F. Selleri and A. Van der Merwe. *Quantum Paradoxes and Physical Reality*. Kluwer Academic Publishers, 1990.

[105] Atsuo Shitade, Motohiko Ezawa, and Naoto Nagaosa. Manipulation of two spin qubits in a double quantum dot using an electric field. *Phys. Rev. B*, 82(19):195305, Nov 2010.

[106] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Sci. Statist. Comput.*, 26:1484, 1997.

[107] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[108] Robert H Silsbee. Spin–orbit induced coupling of charge current and spin polarization. *Journal of Physics: Condensed Matter*, 16(7):R179–R207, 2004.

[109] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26:116–123, 1994.

[110] T D Stanescu and S Tewari. Majorana fermions in semiconductor nanowires: fundamentals, modeling, and experiment. *Journal of Physics: Condensed Matter*, 25(23):233201, 2013.

[111] A. M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793–797, Jul 1996.

[112] Ady Stern and Netanel H. Lindner. Topological quantum computation?from basic concepts to first experiments. *Science*, 339(6124):1179–1184, 2013.

[113] J. M. Taylor, C. M. Marcus, and M. D. Lukin. Long-lived memory for mesoscopic quantum bits. *Phys. Rev. Lett.*, 90(20):206803, May 2003.

[114] J. M. Taylor, J. R. Petta, A. C. Johnson, A. Yacoby, C. M. Marcus, and M. D. Lukin. Relaxation, dephasing, and quantum control of electron spins in double quantum dots. *Phys. Rev. B*, 76(3):035315, Jul 2007.

[115] Vladimir Tello, Miha Nemevsek, Fabrizio Nesti, Goran Senjanovic, and Francesco Vissani. Left-right symmetry: From the lhc to neutrinoless double beta decay. *Phys. Rev. Lett.*, 106:151801, Apr 2011.

[116] Gotz S. Uhrig. Keeping a quantum bit alive by optimized pi-pulse sequences. *Phys. Rev. Lett.*, 98:100504, Mar 2007.

[117] I. van Weperen, B. D. Armstrong, E. A. Laird, J. Medford, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Charge-state conditional operation of a spin qubit. *Phys. Rev. Lett.*, 107(3):030506, Jul 2011.

[118] G.E. Volovik. An analog of the quantum hall effect in a superfluid-3 he film. *JETP*, 67(9):1804–1811.

[119] S.J. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78?88, 1983.

[120] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, October 1982.

[121] W Zawadzki and P Pfeffer. Spin splitting of subband energies due to inversion asymmetry in semiconductor heterostructures. *Semiconductor Science and Technology*, 19(1):R1–R17, 2004.

[122] Clarence Zener. Non-Adiabatic Crossing of Energy Levels. *Proceedings of the Royal Society of London. Series A*, 137(833):696–702, Sep 1932.

[123] Wenxian Zhang, V. V. Dobrovitski, Lea F. Santos, Lorenza Viola, and B. N. Harmon. Dynamical control of electron spin coherence in a quantum dot: A theoretical study. *Phys. Rev. B*, 75(20):201302, 2007.

[124] A. A. Zyuzin, Diego Rainis, Jelena Klinovaja, and Daniel Loss. Correlations between majorana fermions through a superconductor. *Phys. Rev. Lett.*, 111:056802, Jul 2013.

Appendix A

Double Quantum Dots Simulation Code

```
/*
 * RK4 program written by David Drummond
 */

#include <iostream>
#include <cstdlib>
#include <string>
#include <fstream>
#include <complex>
#include <Eigen/Dense>
#include "grev_rnd.h"

#define PI 3.1415926535897932

using namespace std;
using namespace Eigen;

//global variables
Matrix2cd U1, U2, B0, I, sx, sy, sz,
      H1, H2, k1, k2, k3, k4;
double chi=0.5;   //approximate percentage of time spent moving
double T=400.0;   //total time to complete 1 circle, including rest time
double tau=T*chi/2;  //time spent to make one semi-circle
int steps=400*3200;  //good convergence at 3200
int tt=0;
double TT=40000.0;
double stepSize=T/steps;
double beta = 0.075;
double b=2*PI/*+0.001*/;
double t1=0.0;
double t2=TT/2;
```

```
32 double t3=TT;
33
34 //prototypes
35 void plotOriginal();
36 double angle(double t, int dir, int dot);
37 double velx(double t, int dir, int dot);
38 double vely(double t, int dir, int dot);
39 double nonStatic(double t, int dir, int dot, int field);
40 Matrix2cd Ham(double t, int dir, int dot);
41 void plotAngle(int dir, int reps, int dot);
42 void plotVel(int dir, int reps, int dot);
43 void plotSpli();
44 void plotField(int dir, int reps, int dot);
45 void plotPhi(int dir, int reps);
46 void display(int dir, int reps);
47 double infid(Matrix2cd & u1, Matrix2cd & u2);
48 Matrix4cf kronProd(Matrix2cd & A, Matrix2cd & B);
49 void initMatrices();
50 void rk4(int dir, int reps);
51 void repeatSim(int cycles, int dir, int reps);
52
53 //plot original field from do_field
54 void plotOriginal()
55 {
56    cout<<endl<<endl;
57    cout<<"# Plotting original field"<<endl;
58    for(int i=0; i<256*2;i++)
59       cout<< i*0.5<<"\t"<<do_field(i*0.5,0)
60          <<"\t"<<do_field(i*0.5,1)
61          <<"\t"<<do_field(i*0.5,2)
62          <<"\t"<<do_field(i*0.5,3)
63          <<"\t"<<do_field(i*0.5,4)
64          <<"\t"<<do_field(i*0.5,5)
65          <<"\t"<<do_field(i*0.5,6)
66          <<"\t"<<do_field(i*0.5,7)
67          <<"\t"<<do_field(i*0.5,8)
68       <<endl;
69    cout<<endl<<endl;
70 }
71
72 //returns number between 0 and 256 representing angle between 0 and 2*pi
73 //for dot going in direction dir - allows t>T using fmod function
74 double angle(double t, int dir, int dot)
75 {
76    return 64*(2+ dir*tanh(5*(fmod(t+dot*T/2,T)-T/4)/tau)
77             + dir*tanh(5*(fmod(t+dot*T/2,T)-3*T/4)/tau));
78 }
79
80 double velx(double t, int dir, int dot)
81 {
82    return dir*sin(2*PI*fmod(t+dot*T/2,T)/T)*(pow(cosh(5*(fmod(t+dot*T/2,T↩
       )-T/4)/tau), -2)
83       + pow(cosh(5*(fmod(t+dot*T/2,T)-3*T/4)/tau),-2));
84 }
85
86 double vely(double t, int dir, int dot)
87 {
88    return 3.5*dir*cos(2*PI*fmod(t+dot*T/2,T)/T)*(pow(cosh(5*(fmod(t+dot*T↩
       /2,T)-T/4)/tau), -2)
89       + pow(cosh(5*(fmod(t+dot*T/2,T)-3*T/4)/tau),-2));
```

```
90 }
91
92 inline
93 double nonStatic(double t, int dir, int dot, int field)
94 {
95    /*return (t/TT)*do_field(angle(t, dir, dot), field)
96    +((TT-t)/TT)*do_field(angle(t, dir, dot), field+3);
97    */
98       return ( (t-t2)*(t-t3)/((t1-t2)*(t1-t3)) )*do_field(angle(t, dir, ↩
          dot), field)
99     + ( (t-t1)*(t-t3)/((t2-t1)*(t2-t3)) )*do_field(angle(t, dir, dot), ↩
         field+3)
100    + ( (t-t2)*(t-t1)/((t3-t2)*(t3-t1)) )*do_field(angle(t, dir, dot), ↩
         field+6);
101
102
103 }
104
105 inline
106 Matrix2cd Ham(double t, int dir, int dot)
107 {
108 /* return B0
109    + sx*do_field(angle(t, dir, dot),0)
110    + sy*do_field(angle(t, dir, dot),1);
111    + sz*do_field(angle(t, dir, dot),2);
112    // + sx*beta*(-1.0*velx(t, dir, dot)+vely(t, dir, dot))
113    // + sy*beta*(-1.0*velx(t, dir, dot)+vely(t, dir, dot));
114  */
115       return B0
116       + sx*beta*(-1.0*velx(t, dir, dot)+vely(t, dir, dot))
117       + sy*beta*(-1.0*velx(t, dir, dot)+vely(t, dir, dot))
118       + sx*nonStatic(t, dir, dot,0)
119       + sy*nonStatic(t, dir, dot,1)
120       + sz*nonStatic(t, dir, dot,2);
121
122 }
123
124 void plotAngle(int dir, int reps, int dot)
125 {
126    cout<<endl<<endl;
127    cout<<"# Plotting angle for dot "<<dot+1<<" with direction "<<dir<<↩
          endl;
128    tt=0;
129    int tempDir=1; //used to reverse direction if dir=-1
130    for(int j=0; j<reps; j++)
131    {
132       for(int i=0; i<steps; i++)
133       {
134          tt++;
135          cout<< tt*stepSize<<"\t"<<angle(tt*stepSize, tempDir, dot)<<endl↩
              ;
136       }
137       tempDir*=dir;  //changes direction if dir=-1
138    }
139 }
140
141 void plotVel(int dir, int reps, int dot)
142 {
143    cout<<endl<<endl;
```

```
144     cout<<"# Plotting velocity for "<<dot+1<<" with direction "<<dir<<endl↩
            ;
145     tt=0;
146     int tempDir=1; //used to reverse direction if dir=-1
147     for(int j=0; j<reps; j++)
148     {
149        for(int i=0; i<steps; i++)
150        {
151           tt++;
152           cout<< tt*stepSize<<"\t"<<velx(tt*stepSize, tempDir, dot)
153                       <<"\t"<<vely(tt*stepSize, tempDir, dot)<<endl;
154        }
155        tempDir*=dir;  //changes direction if dir=-1
156     }
157 }
158 
159 void plotSpli()
160 {
161     cout<<endl<<endl;
162     for(int t=0;t<TT;t++)
163        cout << t << '\t' <<( (t-t2)*(t-t3)/((t1-t2)*(t1-t3)) )
164           << '\t'<<( (t-t1)*(t-t3)/((t2-t1)*(t2-t3)) )
165        << '\t' <<( (t-t2)*(t-t1)/((t3-t2)*(t3-t1)) )<<endl;
166 }
167 
168 void plotField(int dir, int reps, int dot)
169 {
170     cout<<endl<<endl;
171     cout<<"# Plotting angle for dot "<<dot+1<<" with direction "<<dir<<↩
            endl;
172     tt=0;
173     int tempDir=1; //used to reverse direction if dir=-1
174     for(int j=0; j<reps; j++)
175     {
176        for(int i=0; i<steps; i++)
177        {
178           tt++;
179           //cout<< tt*stepSize<<"\t"<<do_field(angle(tt*stepSize, tempDir,↩
                   dot),2)<<endl;
180           cout<< tt*stepSize<<"\t"<<nonStatic(tt*stepSize,tempDir,dot,2)<<↩
                  endl;
181        }
182        tempDir*=dir;  //changes direction if dir=-1
183     }
184 }
185 
186 void plotPhi(int dir, int reps)
187 {
188     cout<<endl<<endl;
189     cout<<"# Plotting phi with direction "<<dir<<endl;
190     tt=0;
191     double phi1=0.0;
192     double phi2=0.0;
193     int tempDir=1; //used to reverse direction if dir=-1
194     for(int j=0; j<reps; j++)
195     {
196        for(int i=0; i<steps; i++)
197        {
198           tt++;
199           phi1+=do_field(angle(tt*stepSize,tempDir, 0) ,2);
```

```
200         phi2+=do_field(angle(tt*stepSize,tempDir, 1) ,2);
201         cout<< tt*stepSize<<"\t"<<phi1<<"\t"<<phi2<<endl;
202       }
203       tempDir*=dir;  //changes direction if dir=-1
204    }
205 }
206
207 void display(int dir, int reps)
208 {
209    plotOriginal();
210    plotAngle(dir, reps, 0);
211    plotAngle(dir, reps, 1);
212    plotField(dir, reps, 0);
213    plotField(dir, reps, 1);
214    plotSpli();
215    plotVel(dir, reps, 1);
216 // plotPhi(dir, reps);
217 }
218
219 //initialize matrices before rk4 iterations
220 void initMatrices()
221 {
222    U1 << 1.0,0.0,0.0,1.0;  //Initialize U to 1 for dot 1
223    U2 << 1.0,0.0,0.0,1.0;  //Initialize U to 1 for dot 2
224    I(0,0)=complex<double>(0.0,-stepSize); //Eigen doesn't like to mix ↩
       complex numbers with matrices
225    I(1,1)=complex<double>(0.0,-stepSize); //so an i=sqrt(-1) matrix is ↩
       constructed
226                                  //also include -step from the ↩
                                        differential equation for efficiency
227    sx << 0.0,1.0,1.0,0.0;
228    sz << 1.0,0.0,0.0,-1.0;
229    B0 << b,0.0,0.0,-b;  //Applied B field
230    sy(0,1)=complex<double>(0.0,-1.0);
231    sy(1,0)=complex<double>(0.0,1.0);
232 }
233
234 //returns the kronecker product of 2x2 matrices A and B
235 Matrix4cf kronProd(Matrix2cd & A, Matrix2cd & B)
236 {
237    Matrix4cf K;
238    for(int i=0; i<4; i++)
239    {
240       for(int j=0; j<4; j++)
241          K(i,j)=A(floor(i/2),floor(j/2))*B(i%2,j%2);
242    }
243    return K;
244 }
245
246 //outputs the average infidelity of u1 and u2
247 double infid(Matrix2cd & u1, Matrix2cd & u2)
248 {
249    Matrix4cf U=kronProd(u1,u2);  //forms kronecker product of the ↩
       matrices
250    U(0,0)=0; U(3,0)=0; U(0,3)=0; U(3,3)=0;   //project onto logical ↩
       subspace
251    Matrix4cf UUdag=U*U.adjoint();
252    return (1-real(abs(U.trace())*abs(U.trace()) + UUdag.trace())/6);
253 }
254
```

```
255 //performs RK4
256 void rk4(int dir, int reps)
257 {
258    cout<<endl<<endl;
259    tt=0;
260    int tempDir=1; //used to reverse direction if dir=-1
261    H1 = Ham(0,tempDir,0);  //initial H value
262    H2 = Ham(0,tempDir,1);
263    for(int j=0; j<reps; j++)
264    {
265       for(int i=0; i<steps; i++)
266       {
267          k1 = I*H1*U1;  //evaluate k1 using the previous H value
268          H1 = Ham((tt+tempDir*0.5)*stepSize, tempDir, 0);   //update for ↩
               half step
269          k2 = I*H1*(U1+0.5*k1);
270          k3 = I*H1*(U1+0.5*k2);
271          H1 = Ham((tt+tempDir)*stepSize, tempDir, 0); //update H for next↩
                step
272          k4 = I*H1*(U1+k3);
273          U1+=(1/6.0)*(k1+2*k2+2*k3+k4);
274
275          k1 = I*H2*U2;  //same for dot 2
276          H2 = Ham((tt+tempDir*0.5)*stepSize, tempDir, 1);
277          k2 = I*H2*(U2+0.5*k1);
278          k3 = I*H2*(U2+0.5*k2);
279          H2 = Ham((tt+tempDir)*stepSize, tempDir, 1);
280          k4 = I*H2*(U2+k3);
281          U2+=(1/6.0)*(k1+2*k2+2*k3+k4);
282
283          tt++;
284          if( (i<16000)||(i>(steps-16000)) )  //if 16000 steps within ↩
               commensurate time
285             {if(tt%10==1)
286                printf("%10.9g \t %10.9g \n",tt*stepSize, infid(U1,U2));
287             }
288          else
289             if(tt%3200==1)
290                printf("%10.9g \t %10.9g \n",tt*stepSize, infid(U1,U2));
291       }
292       tempDir*=dir;  //changes direction if dir=-1
293    }
294 }
295
296 //repeats sim cycles times, each with different steps, or TT
297 void repeatSim(int cycles, int dir, int reps)
298 {
299    for(int i=0;i<cycles;i++)
300    {
301       cout<<"#TT="<<TT<<endl;
302       rk4(dir, reps);
303 //     steps*=2;
304       TT-=2000;
305       t2=TT/2;
306       t3=TT;
307       tt=0;
308       stepSize=T/steps;
309       initMatrices();
310    }
311 }
```

```
312
313 int main (int argc, const char *argv[])
314 {
315    initMatrices();
316    do_init(argc,argv);
317    int reps=40;
318    int dir=-1;
319    cout<<"# Running rk4 with step size: "<<stepSize<<endl
320       <<"# "<<reps<<" Reps with direction="<<dir<<" with TT="<<TT<<endl;
321    rk4(dir, reps);
322 // repeatSim(5, dir, reps);
323    kill_spline();
324    return 0;
325 }
```

Appendix B

Majorana Wire Simulation Code

```
/* Author - David Drummond
 * Majorana program to simulate Hamiltonian of semiconductor wire
 * Note that unit tests are commented out, rather than deleted
 *
 * verbosity key
 * v=0 nothing
 * v=1 parameter comment
 * v=2 matrices comment
 * v=4 domain walls
 * v=8 chemical potential - animation compatible if TT!=0
 * v=16 eigenvalues for zero-energy and lowest bulk state
 * v=32 Zero-energy spatial distribution - animation compatible if TT!=0
 * v=64 non-adiabatic leakage
 */
#include "c-major.h"
/*         t     Vx   Vy   Vz  delta mu alpha lambda  AL   AR     durL   durR TauL TauR   L1    L2   L3   orig1 orig2 orig3 tSteps TT dispRate nx  ny  nz step  per mN  v */

params prm={11.3, 0.0, 0.0, 0.9, 0.5, 2.12, 2,  0.06, 0.11, 0.115, 496.6, 442, 200, 200, 0.13, 0.09, 0.1355, 0.0, 0.5, 1.0,  100,  800,  4,  400, 0,  0,  3,   0, 3, 56};
int NMAX=1000; /* matrix is initialized with size 4*NMAX, but reduced to actual size
   * after prm are initialized 4nn by 4nn hamiltonian of dynamic size X by X of
   * complex double coefficients initially full of zeros.  4nn for creation (c*) and
   * annihilation (c), as well as spin,
   * operators basis is c1u, c1d, c2u, c2d,...c(nn)u, c(nn)d, c*1u, c*1d, c*2u, c*2d,...c*(nn)u, c*(nn)d
   */
MatrixXcd ham, evec, maj, Usub, Iden, Uinit, Td, projMat1L, projMat2L, projMat3L, projMat1R, projMat2R, projMat3R;
MatrixXcd addMat1, addMat2, addMat3, /*addMat1R, addMat2R, addMat3R,*/ locL, locR, symL, symR, tempMaj;
MatrixXcd maj1L, maj2L, maj3L, maj1R, maj2R, maj3R, fixMat1L, fixMat2L, fixMat3L, fixMat1R, fixMat2R, fixMat3R;
VectorXd chem, Vx, Vy, Vz, eval;
VectorXcd del, initState, prob;
SelfAdjointEigenSolver<MatrixXcd> sol; //workspace to solve eigen equations
```

```
31 std::vector<T> tripletList;   //vector to hold triplets of row, col, ↩
       elements for sparse Matrix
32 SparseMatrix<complex<double> > sparseHam;
33 double latSize, stepSize, startL, startR;
34 int nn, mN, size;
35 IOFormat comMat(7, 0, "", "\n", "#  [ ", " ]");
36    //precision, alignRows, coeffSep, rowSep, rowPrefix, rowSuffix, ↩
         matPrefix, matSuffix
37
38 void init_params(int argc, char *argv[]){ /* read cmd line params */
39     int i,ok=1;
40     for(i=1;i<argc;++i){
41         if(
42          sscanf(argv[i],"t=%lg",&prm.t)   ||
43          sscanf(argv[i],"Vx=%lg",&prm.Vx)   ||
44          sscanf(argv[i],"Vy=%lg",&prm.Vy)   ||
45          sscanf(argv[i],"Vz=%lg",&prm.Vz)   ||
46          sscanf(argv[i],"delta=%lg",&prm.delta)   ||
47          sscanf(argv[i],"mu=%lg",&prm.mu)||
48          sscanf(argv[i],"alpha=%lg",&prm.alpha)||
49          sscanf(argv[i],"lambda=%lg",&prm.lambda)||
50          sscanf(argv[i],"AL=%lg",&prm.AL)||
51          sscanf(argv[i],"AR=%lg",&prm.AR)||
52          sscanf(argv[i],"durL=%lg",&prm.durL)||
53          sscanf(argv[i],"durR=%lg",&prm.durR)||
54          sscanf(argv[i],"TauL=%lg",&prm.TauL)||
55          sscanf(argv[i],"TauR=%lg",&prm.TauR)||
56          sscanf(argv[i],"L1=%lg",&prm.L1)||
57          sscanf(argv[i],"L2=%lg",&prm.L2)||
58          sscanf(argv[i],"L3=%lg",&prm.L3)||
59          sscanf(argv[i],"orig1=%lg",&prm.orig1)||
60          sscanf(argv[i],"orig2=%lg",&prm.orig2)||
61          sscanf(argv[i],"orig3=%lg",&prm.orig3)||
62          sscanf(argv[i],"tSteps=%d",&prm.tSteps)||
63          sscanf(argv[i],"TT=%d",&prm.TT)||
64          sscanf(argv[i],"dispRate=%d",&prm.dispRate)||
65          sscanf(argv[i],"nx=%d",&prm.nx)||
66          sscanf(argv[i],"ny=%d",&prm.ny)||
67          sscanf(argv[i],"nz=%d",&prm.nz)||
68          sscanf(argv[i],"step=%d",&prm.step)||
69          sscanf(argv[i],"per=%d",&prm.per)||
70          sscanf(argv[i],"mN=%d",&prm.mN)||
71          sscanf(argv[i],"v=%d",&prm.v)
72          )
73       {
74          if(prm.v&1)
75             printf("# main:  scanned arg[%d]=\"%s\"\n",i,argv[i]);
76          ok++;
77       }
78     }
79    if((prm.nx+prm.ny+prm.nz)>NMAX)
80       cout<<"ERROR: Number of lattice sites is larger than NMAX"<<endl;
81     if(ok !=argc)
82       cout<<"invalid input"<<endl
83          <<"USAGE: [t=#] [Vx=#] [Vy=#] [Vz=#] [delta=#] [mu=#] [alpha=#] ↩
               [lambda=#] [AL=#] [AR=#] [durL=#] [durR=#] [TauL=#] [TauR=#]↩
               [L1=#] [L2=#] [L3=#] [orig1=#] [orig2=#] [orig3=#] [tSteps↩
              =#] [TT=#] [dispRate=#] [n?=#] [per=#] [mN=#] [step=#] [v=#]↩
              "
84          <<endl<<"where ?=x,y,z"<<endl;
```

```
85    if(prm.v&1)
86    {
87       printf("# params: t=%g Vx=%g Vy=%g Vz=%g delta=%g mu=%g alpha=%g lambda=%g AL=%g AR=%g durL=%g durR=%g TauL=%g TauR=%g L1=%g L2=%g L3=%g orig1=%g orig2=%g orig3=%g tSteps=%d TT=%d dispRate=%d nx=%d ny=%d nz=%d step=%d per=%d mN=%d v=%d\n",
88          prm.t, prm.Vx, prm.Vy, prm.Vz, prm.delta, prm.mu, prm.alpha, prm.lambda, prm.AL, prm.AR, prm.durL, prm.durR, prm.TauL, prm.TauR, prm.L1, prm.L2, prm.L3, prm.orig1, prm.orig2, prm.orig3, prm.tSteps, prm.TT, prm.dispRate, prm.nx, prm.ny, prm.nz, prm.step, prm.per, prm.mN, prm.v);
89       cout<<"# verbosity key: v=0 nothing, v=1 parameter comment, v=2 matrices comment, v=4 domain walls, v=8 chemical potential, v=16 eigenvalues, v=32 Zero-energy spatial distribution, v=64 non-adiabatic leakage"<<endl;
90    }
91    nn=prm.nx+prm.ny+prm.nz;
92    mN=prm.mN;
93    size=powInt(2,mN);
94    latSize=1.0/nn;
95    stepSize=1.0/prm.tSteps;
96    chem = VectorXd::Zero(nn);
97    del = VectorXcd::Zero(nn);
98    Vx = VectorXd::Zero(nn);
99    Vy = VectorXd::Zero(nn);
100   Vz = VectorXd::Zero(nn);
101   prob = VectorXcd::Zero(size);
102   initState = VectorXcd::Zero(size);
103// initState(0)=0.5*sqrt(2);
104   initState(0)=1;
105   initState(1)=0;
106   initState(2)=0;
107   initState(3)=0;
108// initState(3)=-0.5*sqrt(2);
109   initState(4)=0;
110   initState(5)=0;
111   initState(6)=0;
112   initState(7)=0;
113   ham = MatrixXcd::Zero(4*nn,4*nn);
114   maj = MatrixXcd::Zero(4*nn,mN);
115   maj1L = MatrixXcd::Zero(4*nn,1); maj2L = MatrixXcd::Zero(4*nn,1); maj3L = MatrixXcd::Zero(4*nn,1);
116   maj1R = MatrixXcd::Zero(4*nn,1); maj2R = MatrixXcd::Zero(4*nn,1); maj3R = MatrixXcd::Zero(4*nn,1);
117   tempMaj = MatrixXcd::Zero(4*nn,1);
118   locL = MatrixXcd::Zero(4*nn,mN); locR = MatrixXcd::Zero(4*nn,mN);
119   symL = MatrixXcd::Zero(4*nn,mN); symR = MatrixXcd::Zero(4*nn,mN);
120
121//#include "savedMaj400.dat"
122   projMat1L = MatrixXcd::Zero(4*nn,4*nn); projMat2L = MatrixXcd::Zero(4*nn,4*nn);  projMat3L = MatrixXcd::Zero(4*nn,4*nn);
123   projMat1R = MatrixXcd::Zero(4*nn,4*nn); projMat2R = MatrixXcd::Zero(4*nn,4*nn);  projMat3R = MatrixXcd::Zero(4*nn,4*nn);
124   fixMat1L = MatrixXcd::Zero(4*nn,4*nn); fixMat2L = MatrixXcd::Zero(4*nn,4*nn); fixMat3L = MatrixXcd::Zero(4*nn,4*nn);
125   fixMat1R = MatrixXcd::Zero(4*nn,4*nn); fixMat2R = MatrixXcd::Zero(4*nn,4*nn); fixMat3R = MatrixXcd::Zero(4*nn,4*nn);
126   int cut = nn*0.2;
127   for(int i=0; i<nn; i++){
128      if(i <= cut){
```

```
129         projMat1L(2*i, 2*i) = 1;
130         projMat1L(2*i + 1, 2*i + 1) = 1;
131         projMat1L(2*i + 2*nn, 2*i + 2*nn) = 1;
132         projMat1L(2*i + 2*nn + 1, 2*i + 2*nn + 1) = 1;
133      }
134      if(i > cut && i <= 2*cut + 0.2*nn){
135         projMat2L(2*i, 2*i) = 1;
136         projMat2L(2*i + 1, 2*i + 1) = 1;
137         projMat2L(2*i + 2*nn, 2*i + 2*nn) = 1;
138         projMat2L(2*i + 2*nn + 1, 2*i + 2*nn + 1) = 1;
139      }
140      if(i > 2*cut + 0.2*nn){
141         projMat3L(2*i, 2*i) = 1;
142         projMat3L(2*i + 1, 2*i + 1) = 1;
143         projMat3L(2*i + 2*nn, 2*i + 2*nn) = 1;
144         projMat3L(2*i + 2*nn + 1, 2*i + 2*nn + 1) = 1;
145      }
146   }
147   cut = nn*0.4;
148   for(int i=0; i<nn; i++){
149      if(i <= cut){
150         projMat1R(2*i, 2*i) = 1;
151         projMat1R(2*i + 1, 2*i + 1) = 1;
152         projMat1R(2*i + 2*nn, 2*i + 2*nn) = 1;
153         projMat1R(2*i + 2*nn + 1, 2*i + 2*nn + 1) = 1;
154      }
155      if(i > cut && i <= 2*cut){
156         projMat2R(2*i, 2*i) = 1;
157         projMat2R(2*i + 1, 2*i + 1) = 1;
158         projMat2R(2*i + 2*nn, 2*i + 2*nn) = 1;
159         projMat2R(2*i + 2*nn + 1, 2*i + 2*nn + 1) = 1;
160      }
161      if(i > 2*cut){
162         projMat3R(2*i, 2*i) = 1;
163         projMat3R(2*i + 1, 2*i + 1) = 1;
164         projMat3R(2*i + 2*nn, 2*i + 2*nn) = 1;
165         projMat3R(2*i + 2*nn + 1, 2*i + 2*nn + 1) = 1;
166      }
167   }
168   addMat1 = MatrixXcd::Zero(mN, mN);  addMat2 = MatrixXcd::Zero(mN, mN);↩
          addMat3 = MatrixXcd::Zero(mN, mN);
169   for(int i=0; i<mN; i++){
170      addMat1(i,0) = 1;
171      addMat2(i,1) = 1;
172      addMat3(i,2) = 1;
173   }
174
175   Usub = MatrixXcd::Zero(4*nn,2*mN);
176   Uinit = MatrixXcd::Zero(4*nn,2*mN);
177   Td = MatrixXcd::Identity(2*mN,2*mN);
178   Iden = MatrixXcd::Identity(2*mN,2*mN);
179   sparseHam.resize(4*nn,4*nn);
180   tripletList.reserve(28*nn-16*prm.per);
181   /* reserves space for the number of non-zero elements in sparse Ham nn↩
          on-site h,
182    * nn on-site V, nn on-site Delta...*2 for spin plus nn-1 t, nn-1 ↩
          alpha...*2 for
183    * left and right movers and *2 for spin gives 6*nn+8*(nn-1)=14*nn↩
          -8...*2 for Nambu
```

```
184     * "symmetry" = 28*nn-16 if periodic all movers have nn terms instead of nn-1
185     */
186    prm.TauL=5000*max(prm.AL,prm.AR);
187    prm.TauR=prm.TauL;
188    startL=prm.TauL;
189    startR=prm.TauR;
190    prm.TT=2*max(prm.TauL, prm.TauR)+max(prm.durL,prm.durR);
191 // prm.dispRate=prm.TT/20; //display 20 times
192 }
193
194 void disp(double t)
195 {
196    if(prm.v&2) //Display hamiltonian matrix
197    {
198       cout<<"# Hamiltonian"<<endl<<ham.format(comMat)<<endl;
199       cout<<"# Multi-Particle Basis"<<endl;
200       multBasis();
201    }
202    if(prm.v&4) //plot domain walls with the time
203    {
204       cout<<t<<'\t';
205       if(prm.step==1)   //if 1 topological region, L2 and orig2 used for the middle
206          cout<<(prm.orig2-prm.L2/2)<<'\t'<<(prm.orig2+prm.L2/2+moveWallL(t))<<endl;
207       if(prm.step==2)   //if 2 regions, L1 and L3 are used tyipically toward the ends
208          cout<<(prm.orig1-prm.L1)<<'\t'<<(prm.orig1+prm.L1+moveWallL(t))<<'\t'
209          <<(prm.orig3-prm.L3-moveWallR(t))<<'\t'<<(prm.orig3+prm.L3)<<endl;
210       if(prm.step==3)   //if 3 regions, all are used
211          cout<<(prm.orig1-prm.L1)<<'\t'<<(prm.orig1+prm.L1+moveWallL(t))<<'\t'
212          <<(prm.orig2-prm.L2/2-moveWallL(t))<<'\t'<<(prm.orig2+prm.L2/2+moveWallR(t))<<'\t'
213          <<(prm.orig3-prm.L3-moveWallR(t))<<'\t'<<(prm.orig3+prm.L3)<<endl;
214    }
215    if(prm.v&8) //output chemical potential spatial distribution
216    {
217       for(int i=0;i<nn;i++)
218       {
219          cout<<i*latSize<<'\t'<<chem(i)<<endl;
220       }
221       cout<<endl<<endl; //new index
222    }
223    if(prm.v&16) //output mN eigenvalues + 1 bulk eigenvalue
224    {
225       cout<<"# Eigenvalues"<<endl;
226       for(int i=0;i<mN+1;i++)
227       {
228          cout<<eval(2*nn+i)<<'\t';
229       }
230       cout<<endl<<endl<<endl; //new index
231    }
232    if(prm.v&32)   //Majorana spatial distribution
233    {
234       cout<<"# Location"<<'\t'<<"u-Spin Up"<<'\t'<<"u-Spin Down"<<'\t'
```

```
235         <<"v-Spin Up"<<'\t'<<"v-Spin Down"<<endl;
236       for(int i=0; i<nn; i++) //i iterates thru the spatial coordinate
237       {
238          cout<<i*latSize;
239          for(int j=0; j<mN; j++) //j iterates thru the lowest energy ↩
                 level eigenvectors
240          {
241             printf("\t %10.9g \t %10.9g \t %10.9g \t %10.9g",
242                   real(symL(2*i,j)), real(symL(2*i+1,j)), real(symL(2*nn↩
                      +2*i,j)), real(symL(2*nn+2*i+1,j)));
243          }
244          cout<<endl;
245       }
246       cout<<endl<<endl; //new index
247
248       cout<<"# Location"<<'\t'<<"u-Spin Up"<<'\t'<<"u-Spin Down"<<'\t'
249          <<"v-Spin Up"<<'\t'<<"v-Spin Down"<<endl;
250       for(int i=0; i<nn; i++) //i iterates thru the spatial coordinate
251       {
252          cout<<i*latSize;
253          for(int j=0; j<mN; j++) //j iterates thru the lowest energy ↩
                 level eigenvectors
254          {
255             printf("\t %10.9g \t %10.9g \t %10.9g \t %10.9g",
256                   real(symR(2*i,j)), real(symR(2*i+1,j)), real(symR(2*nn↩
                      +2*i,j)), real(symR(2*nn+2*i+1,j)));
257          }
258          cout<<endl;
259       }
260       cout<<endl<<endl; //new index
261
262       cout<<"# Location"<<'\t'<<"u-Spin Up"<<'\t'<<"u-Spin Down"<<'\t'
263          <<"v-Spin Up"<<'\t'<<"v-Spin Down"<<endl;
264       for(int i=0; i<nn; i++) //i iterates thru the spatial coordinate
265       {
266          cout<<i*latSize;
267          for(int j=0; j<mN; j++) //j iterates thru the lowest energy ↩
                 level eigenvectors
268          {
269             printf("\t %10.9g \t %10.9g \t %10.9g \t %10.9g",
270                   norm(locL(2*i,j)), norm(locL(2*i+1,j)), norm(locL(2*nn↩
                      +2*i,j)), norm(locL(2*nn+2*i+1,j)));
271          }
272          cout<<endl;
273       }
274       cout<<endl<<endl; //new index
275
276       cout<<"# Location"<<'\t'<<"u-Spin Up"<<'\t'<<"u-Spin Down"<<'\t'
277          <<"v-Spin Up"<<'\t'<<"v-Spin Down"<<endl;
278       for(int i=0; i<nn; i++) //i iterates thru the spatial coordinate
279       {
280          cout<<i*latSize;
281          for(int j=0; j<mN; j++) //j iterates thru the lowest energy ↩
                 level eigenvectors
282          {
283             printf("\t %10.9g \t %10.9g \t %10.9g \t %10.9g",
284                   norm(locR(2*i,j)), norm(locR(2*i+1,j)), norm(locR(2*nn↩
                      +2*i,j)), norm(locR(2*nn+2*i+1,j)));
285          }
286          cout<<endl;
```

```
287         }
288         cout<<endl<<endl; //new index
289 
290         cout<<"# Location"<<'\t'<<"u-Spin Up"<<'\t'<<"u-Spin Down"<<'\t'
291            <<"v-Spin Up"<<'\t'<<"v-Spin Down"<<endl;
292         for(int i=0; i<nn; i++) //i iterates thru the spatial coordinate
293         {
294            cout<<i*latSize;
295            for(int j=0; j<mN; j++) //j iterates thru the lowest energy ↩
                level eigenvectors
296            {
297               printf("\t %10.9g \t %10.9g \t %10.9g \t %10.9g",
298                     norm(maj(2*i,j)), norm(maj(2*i+1,j)), norm(maj(2*nn+2*i↩
                        ,j)), norm(maj(2*nn+2*i+1,j)));
299            }
300            cout<<endl;
301         }
302         cout<<endl<<endl; //new index
303 
304      }
305      if(prm.v&64)   //check whether subgap space is unitary to test leakage
306      {
307         printf("%10.16g \t", (Iden-(Td.adjoint()*Td)).norm());
308 //      printf("%10.16g \t", real(Usub(0,0))); //check step size for RK4
309      }
310 }
311 
312 //finds sign of maximum absolute coefficient of vec
313 int maxSign(VectorXcd vec){
314    int sign = 1;
315    double max = 0;
316    for(int i=0; i<nn; i++){
317          if( (norm(vec(2*i)) + norm(vec(2*i+1)) ) > max ){
318                max = (norm(vec(2*i)) + norm(vec(2*i+1)) );
319                if( (real(vec(2*i)) + real(vec(2*i+1)) ) > 0 )
320                   sign = 1;
321                else
322                   sign = -1;
323          }
324    }
325    return sign;
326 }
327 
328 int maxLoc(VectorXcd vec){
329    int loc = 0;
330    double max = 0;
331    for(int i=0; i<nn; i++){
332          if( (norm(vec(2*i)) + norm(vec(2*i+1)) ) > max ){
333                max = (norm(vec(2*i)) + norm(vec(2*i+1)) );
334                loc = i;
335          }
336    }
337    return loc;
338 }
339 
340 double moveWallL(double t)
341 {
342    return (prm.AL*0.5)*(tanh((5/prm.TauL)*(t-startL))-tanh((5/prm.TauL)*(↩
       t-startL-prm.durL)));
343 }
```

```
344
345 double moveWallR(double t)
346 {
347    return (prm.AR*0.5)*(tanh((5/prm.TauR)*(t-startR))-tanh((5/prm.TauR)*(↩
       t-startR-prm.durR)));
348 }
349
350 //returns a step that goes from 0 to 1, over length lambda, and is ↩
    centered at origin
351 double step(double x, double lambda, double origin)
352 {
353    return 0.5*(tanh( (5/lambda)*(x - origin)) + 1);
354 }
355 double delFunction(double x, double t)
356 {
357    return prm.delta+x*0.0+t*0.0; //replace with position and/or time ↩
       dependent delta if desired
358 }
359 double magXFunction(double x, double t)
360 {
361    return prm.Vx+x*0.0+t*0.0; //replace with position and/or time ↩
       dependent if desired
362 }
363 double magYFunction(double x, double t)
364 {
365    return prm.Vy+0.0*t+0.0*x; //replace with position and/or time ↩
       dependent if desired
366 }
367 double magZFunction(double x, double t)
368 {
369    return prm.Vz+0.0*t+0.0*x; //replace with position and/or time ↩
       dependent if desired
370 }
371
372 void update_chem(double t) //t is the percentage of time elapsed
373 {
374    for(int i=0;i<nn;i++)
375    {
376       if(prm.step==0)   //if constant
377          chem(i)=prm.mu;
378       if(prm.step==1)   //if 1 topological region, L2 and orig2 used for ↩
           the middle
379          chem(i)=prm.mu*(1+(
380                -1*step(i*latSize, prm.lambda, prm.orig2-prm.L2/2)
381                +1*step(i*latSize, prm.lambda, prm.orig2+prm.L2/2+↩
                      moveWallL(t))
382                   ));
383       if(prm.step==2)   //if 2 regions, L1 and L3 are used tyipically ↩
           toward the ends
384          chem(i)=prm.mu*(1+(
385                -step(i*latSize, prm.lambda, prm.orig1-prm.L1)
386                +step(i*latSize, prm.lambda, prm.orig1+prm.L1+moveWallL(t)↩
                      )
387                -step(i*latSize, prm.lambda, prm.orig3-prm.L3-moveWallR(t)↩
                      )
388                +step(i*latSize, prm.lambda, prm.orig3+prm.L3)
389                   ));
390       if(prm.step==3)   //if 3 regions, all are used
391          chem(i)=prm.mu*(1+(
392                -step(i*latSize, prm.lambda, prm.orig1-prm.L1)
```

```
393                 +step(i*latSize, prm.lambda, prm.orig1+prm.L1+moveWallL(t)↩
                        )
394                 -step(i*latSize, prm.lambda, prm.orig2-prm.L2/2-moveWallL(↩
                        t))
395                 +step(i*latSize, prm.lambda, prm.orig2+prm.L2/2+moveWallR(↩
                        t))
396                 -step(i*latSize, prm.lambda, prm.orig3-prm.L3-moveWallR(t)↩
                        )
397                 +step(i*latSize, prm.lambda, prm.orig3+prm.L3)
398                       ));
399    }
400 }
401
402 void update_del(double t)
403 {
404    for(int i=0;i<nn;i++)
405    {
406       del(i)=delFunction(double(i),t);
407    }
408 }
409 void update_V(double t)
410 {
411    for(int i=0;i<nn;i++)
412    {
413       Vx(i)=magXFunction(double(i),t);
414       Vy(i)=magYFunction(double(i),t);
415       Vz(i)=magZFunction(double(i),t);
416    }
417 }
418 void init_ham(double t) // init the Hamiltonian matrix
419 {
420    int j;
421    double temp;   //used for efficiency
422    update_chem(t);
423    update_del(t);
424    update_V(t);
425    /* Helpful key for Hamiltonian basis
426     *
427     * +1 changes spin
428     * +2 moves to the right in real space
429     * +2nn switches particle to hole
430     *
431     */
432    for(int i=0; i<nn; i++) //on-site term, chemical potential, and Z ↩
           Zeeman term
433    {
434       temp=2.0*prm.t-chem(i);
435       ham(2*i,2*i)=temp+prm.Vz;
436       ham(2*i+1,2*i+1)=temp-prm.Vz;
437       ham(2*i+2*nn,2*i+2*nn)=-1.0*temp-prm.Vz;
438       ham(2*i+2*nn+1,2*i+2*nn+1)=-1.0*temp+prm.Vz;
439    }
440    for(int i=0; i<nn-1; i++)  //hopping terms
441    {
442       j=2*i+2;
443       ham(2*i,j)=-1.0*prm.t;
444       ham(j,2*i)=-1.0*prm.t;
445       ham(2*i+1,j+1)=-1.0*prm.t;
446       ham(j+1,2*i+1)=-1.0*prm.t;
447       ham(2*i+2*nn,j+2*nn)=prm.t;
```

```
448         ham(j+2*nn,2*i+2*nn)=prm.t;
449         ham(2*i+2*nn+1,j+2*nn+1)=prm.t;
450         ham(j+2*nn+1,2*i+2*nn+1)=prm.t;
451     }
452     for(int i=0; i<nn; i++) //X and Y Zeeman terms
453     {
454         j=2*i+1;
455         ham(2*i,j)=complex<double>(Vx(i),-1.0*Vy(i));
456         ham(j,2*i)=complex<double>(Vx(i),1.0*Vy(i));
457         ham(2*i+2*nn,j+2*nn)=complex<double>(-1.0*Vx(i),-1.0*Vy(i));
458         ham(j+2*nn,2*i+2*nn)=complex<double>(-1.0*Vx(i),1.0*Vy(i));
459         //Eigen isn't overloaded to handle real scalar multiplication by ↩
                complex vectors
460     }
461     for(int i=0; i<nn-1; i++)  //Spin-orbit terms
462     {
463         j=2*i+2;
464         temp=0.5*prm.alpha;
465         ham(2*i+1,j)=temp;
466         ham(j,2*i+1)=temp;
467         ham(2*i,j+1)=-1.0*temp;
468         ham(j+1,2*i)=-1.0*temp;
469         ham(2*i+1+2*nn,j+2*nn)=-1.0*temp;
470         ham(j+2*nn,2*i+1+2*nn)=-1.0*temp;
471         ham(2*i+2*nn,j+1+2*nn)=temp;
472         ham(j+1+2*nn,2*i+2*nn)=temp;
473     }
474     for(int i=0; i<nn; i++)    //Superconductor pairing terms
475     {
476         j=2*i+2*nn;
477         ham(2*i,j+1)=del(i);
478         ham(j+1,2*i)=del(i);
479         ham(2*i+1,j)=complex<double>(-1.0,0)*del(i);
480         ham(j,2*i+1)=complex<double>(-1.0,0)*del(i);
481     }
482 }
483
484 void init_ham_sparse(double t)   // init the Hamiltonian matrix as a ↩
    sparse matrix
485 {
486     int j;
487     double temp;   //used for efficiency
488     update_chem(t);
489     update_del(t);
490     update_V(t);
491     for(int i=0; i<nn; i++) //on-site term, chemical potential, and Z ↩
        Zeeman term
492     {
493         temp=2.0*prm.t-chem(i);
494         tripletList.push_back(T(2*i,2*i,temp+prm.Vz));
495         tripletList.push_back(T(2*i+1,2*i+1,temp-prm.Vz));
496         tripletList.push_back(T(2*i+2*nn,2*i+2*nn,-1.0*temp-prm.Vz));
497         tripletList.push_back(T(2*i+2*nn+1,2*i+2*nn+1,-1.0*temp+prm.Vz));
498     }
499     for(int i=0; i<nn-1; i++)  //hopping terms
500     {
501         j=2*i+2;
502         tripletList.push_back(T(2*i,j,-1.0*prm.t));
503         tripletList.push_back(T(j,2*i,-1.0*prm.t));
504         tripletList.push_back(T(2*i+1,j+1,-1.0*prm.t));
```

```
505         tripletList.push_back(T(j+1,2*i+1,-1.0*prm.t));
506         tripletList.push_back(T(2*i+2*nn,j+2*nn,prm.t));
507         tripletList.push_back(T(j+2*nn,2*i+2*nn,prm.t));
508         tripletList.push_back(T(2*i+2*nn+1,j+2*nn+1,prm.t));
509         tripletList.push_back(T(j+2*nn+1,2*i+2*nn+1,prm.t));
510     }
511     for(int i=0; i<nn; i++) //X and Y Zeeman terms
512     {
513         j=2*i+1;
514         tripletList.push_back(T(2*i,j,complex<double>(Vx(i),-1.0*Vy(i))));
515         tripletList.push_back(T(j,2*i,complex<double>(Vx(i),1.0*Vy(i))));
516         tripletList.push_back(T(2*i+2*nn,j+2*nn,complex<double>(-1.0*Vx(i)↩
                ,-1.0*Vy(i))));
517         tripletList.push_back(T(j+2*nn,2*i+2*nn,complex<double>(-1.0*Vx(i)↩
                ,1.0*Vy(i))));
518         //Eigen isn't overloaded to handle real scalar multiplication by ↩
               complex vectors
519     }
520     for(int i=0; i<nn-1; i++)  //Spin-orbit terms
521     {
522         j=2*i+2;
523         temp=0.5*prm.alpha;
524         tripletList.push_back(T(2*i+1,j,temp));
525         tripletList.push_back(T(j,2*i+1,temp));
526         tripletList.push_back(T(2*i,j+1,-1.0*temp));
527         tripletList.push_back(T(j+1,2*i,-1.0*temp));
528         tripletList.push_back(T(2*i+1+2*nn,j+2*nn,-1.0*temp));
529         tripletList.push_back(T(j+2*nn,2*i+1+2*nn,-1.0*temp));
530         tripletList.push_back(T(2*i+2*nn,j+1+2*nn,temp));
531         tripletList.push_back(T(j+1+2*nn,2*i+2*nn,temp));
532     }
533     for(int i=0; i<nn; i++)    //Superconductor pairing terms
534     {
535         j=2*i+2*nn;
536         tripletList.push_back(T(2*i,j+1,del(i)));
537         tripletList.push_back(T(j+1,2*i,del(i)));
538         tripletList.push_back(T(2*i+1,j,complex<double>(-1.0,0)*del(i)));
539         tripletList.push_back(T(j,2*i+1,complex<double>(-1.0,0)*del(i)));
540     }
541 }
542
543 void per_ham(double k)  //add periodic terms if prm.per=1
544 {
545     double temp;    //Periodic SO terms
546     temp=0.5*prm.alpha;
547     ham(0,2*nn-1)=polar(temp,k);
548     ham(2*nn-1,0)=polar(temp,-1.0*k);
549     ham(1,2*nn-2)=polar(-1.0*temp,k);
550     ham(2*nn-2,1)=polar(-1.0*temp,-1.0*k);
551     ham(2*nn,4*nn-1)=polar(-1.0*temp,-1.0*k);
552     ham(4*nn-1,2*nn)=polar(-1.0*temp,k);
553     ham(2*nn+1,4*nn-2)=polar(temp,-1.0*k);
554     ham(4*nn-2,2*nn+1)=polar(temp,k);
555
556     temp=-1.0*prm.t;      //Periodic hopping terms
557     ham(0,2*nn-2)=polar(temp,k);
558     ham(2*nn-2,0)=polar(temp,-1.0*k);
559     ham(1,2*nn-1)=polar(temp,k);
560     ham(2*nn-1,1)=polar(temp,-1.0*k);
561     ham(2*nn,4*nn-2)=polar(-1.0*temp,-1.0*k);
```

```
562     ham(4*nn-2,2*nn)=polar(-1.0*temp,k);
563     ham(2*nn+1,4*nn-1)=polar(-1.0*temp,-1.0*k);
564     ham(4*nn-1,2*nn+1)=polar(-1.0*temp,k);
565
566     temp=0.5*prm.alpha;  //Periodic SO terms
567     tripletList.push_back(T(0,2*nn-1,polar(temp,k)));
568     tripletList.push_back(T(2*nn-1,0,polar(temp,-1.0*k)));
569     tripletList.push_back(T(1,2*nn-2,polar(-1.0*temp,k)));
570     tripletList.push_back(T(2*nn-2,1,polar(-1.0*temp,-1.0*k)));
571     tripletList.push_back(T(2*nn,4*nn-1,polar(-1.0*temp,-1.0*k)));
572     tripletList.push_back(T(4*nn-1,2*nn,polar(-1.0*temp,k)));
573     tripletList.push_back(T(2*nn+1,4*nn-2,polar(temp,-1.0*k)));
574     tripletList.push_back(T(4*nn-2,2*nn+1,polar(temp,k)));
575
576     temp=-1.0*prm.t;      //Periodic hopping terms
577     tripletList.push_back(T(0,2*nn-2,polar(temp,k)));
578     tripletList.push_back(T(2*nn-2,0,polar(temp,-1.0*k)));
579     tripletList.push_back(T(1,2*nn-1,polar(temp,k)));
580     tripletList.push_back(T(2*nn-1,1,polar(temp,-1.0*k)));
581     tripletList.push_back(T(2*nn,4*nn-2,polar(-1.0*temp,-1.0*k)));
582     tripletList.push_back(T(4*nn-2,2*nn,polar(-1.0*temp,k)));
583     tripletList.push_back(T(2*nn+1,4*nn-1,polar(-1.0*temp,-1.0*k)));
584     tripletList.push_back(T(4*nn-1,2*nn+1,polar(-1.0*temp,k)));
585 }
586
587 void update_ham(double t)  // update the Hamiltonian matrix in real space
588 {
589     double temp;   //used for efficiency
590     update_chem(t);
591     for(int i=0; i<nn; i++)    //on-site term, chemical potential, and Z ↩
        Zeeman terms
592     {
593        temp=2.0*prm.t-chem(i);
594        sparseHam.coeffRef(2*i,2*i)=temp+prm.Vz;
595        sparseHam.coeffRef(2*i+1,2*i+1)=temp-prm.Vz; //sparse version
596        sparseHam.coeffRef(2*i+2*nn,2*i+2*nn)=-1.0*temp-prm.Vz;
597        sparseHam.coeffRef(2*i+2*nn+1,2*i+2*nn+1)=-1.0*temp+prm.Vz;
598     }
599 }
600
601 void do_solve(void)
602 {
603     sol.compute(ham);
604     evec = sol.eigenvectors();
605     eval = sol.eigenvalues();
606     maj = evec.block(0,2*nn,4*nn,mN);    //starting row, starting col, row ↩
        length, col length
607     // maj vector corresponds to lowest 3 positive eigenvectors
608     // which are stored in a 4*nn x mN matrix, with the lowest energy ↩
        eigenvector on the left
609     //
610
611     //make the particle-hole symmetric states
612     maj1L = evec.col(2*nn - 1) + evec.col(2*nn);
613     maj2L = evec.col(2*nn - 2) + evec.col(2*nn + 1);
614     maj3L = evec.col(2*nn - 3) + evec.col(2*nn + 2);
615     maj1R = evec.col(2*nn - 1) - evec.col(2*nn);
616     maj2R = evec.col(2*nn - 2) - evec.col(2*nn + 1);
617     maj3R = evec.col(2*nn - 3) - evec.col(2*nn + 2);
618
```

```
619     // check to which is more to the right, switch if needed
620     if( maxLoc(maj1L) > maxLoc(maj1R) ){
621        tempMaj = maj1L;
622        maj1L = maj1R;
623        maj1R = tempMaj;
624     }
625     if( maxLoc(maj2L) > maxLoc(maj2R) ){
626        tempMaj = maj2L;
627        maj2L = maj2R;
628        maj2R = tempMaj;
629     }
630     if( maxLoc(maj3L) > maxLoc(maj3R) ){
631        tempMaj = maj3L;
632        maj3L = maj3R;
633        maj3R = tempMaj;
634     }
635
636     fixMat1L = projMat1L * maxSign(projMat1L*maj1L)
637           + projMat2L * maxSign(projMat2L*maj1L)
638           + projMat3L * maxSign(projMat3L*maj1L);
639     fixMat2L = projMat1L * maxSign(projMat1L*maj2L)
640           + projMat2L * maxSign(projMat2L*maj2L)
641           + projMat3L * maxSign(projMat3L*maj2L);
642     fixMat3L = projMat1L * maxSign(projMat1L*maj3L)
643           + projMat2L * maxSign(projMat2L*maj3L)
644           + projMat3L * maxSign(projMat3L*maj3L);
645     fixMat1R = projMat1R * maxSign(projMat1R*maj1R)
646           + projMat2R * maxSign(projMat2R*maj1R)
647           + projMat3R * maxSign(projMat3R*maj1R);
648     fixMat2R = projMat1R * maxSign(projMat1R*maj2R)
649           + projMat2R * maxSign(projMat2R*maj2R)
650           + projMat3R * maxSign(projMat3R*maj2R);
651     fixMat3R = projMat1R * maxSign(projMat1R*maj3R)
652           + projMat2R * maxSign(projMat2R*maj3R)
653           + projMat3R * maxSign(projMat3R*maj3R);
654
655     symL << fixMat1L*maj1L, fixMat2L*maj2L, fixMat3L*maj3L;
656     symR << fixMat1R*maj1R, fixMat2R*maj2R, fixMat3R*maj3R;
657
658 // symL << maj1L, maj2L, maj3L;
659 // symR << maj1R, maj2R, maj3R;
660
661     //below are the localized versions of the maj
662     locL += projMat1L*symL*addMat1;
663     locL += projMat2L*symL*addMat2;
664     locL += projMat3L*symL*addMat3;
665
666     locR += projMat1R*symR*addMat1;
667     locR += projMat2R*symR*addMat2;
668     locR += projMat3R*symR*addMat3;
669
670     maj = 0.5*(locL + locR);
671
672     //re-normalize
673     MatrixXcd norms = MatrixXcd::Identity(3,3);
674     double sum;
675     for(int i=0; i<mN; i++) //i iterates thru the spatial coordinate
676        {
677           sum = 0;
```

```
678         for(int j=0; j<nn; j++) //j iterates thru the lowest energy level eigenvectors
679         {
680             sum += norm(maj(2*j,i)) + norm(maj(2*j+1,i)) + norm(maj(2*nn+2*j,i)) + norm(maj(2*nn+2*j+1,i));
681         }
682         norms(i,i) = 1.0 / sqrt(sum);
683       }
684    maj = maj * norms;
685 }
686
687 int orderStates(int parity) //re-orders the maj vector to place parity state first, then right to left
688 {
689    int rowCount=0, j=1;
690    double weight, threshold=0.02;   //value of eigenstate which clearly marks majorana location
691    MatrixXcd reOrder = MatrixXcd::Zero(mN,mN);  //matrix to reorder state vectors
692       //i is current location of state and j is final locations
693    reOrder(parity, 0)=1;   //makes the parity state the new 0
694    for(int i=nn-1;i>=0;i--)   //goes thru the wire from right to left
695    {
696       for(int state=0; state<mN; state++) //checks if each state hits threshold
697       {
698          weight=(norm(maj(2*i,state))+norm(maj(2*i+1,state))   //weight of state at current site
699             +norm(maj(2*nn+2*i,state))+norm(maj(2*nn+2*i+1,state)));
700          if( (threshold<weight)&&(weight<threshold+0.003))  //upper bound to reduce sites with-in threshold
701          {
702 //           cout<<"Mode found at site "<<i*1.0/nn<<" for state "<<state<<endl;
703             rowCount=0; //resets rowCount
704             for(int k=0; k<mN;k++)
705             {
706                if(reOrder(state,k)==complex<double>(1,0) )  //counts coefficients in row of state
707                   rowCount++;
708             }
709             if(rowCount==0)   //makes sure state isn't already used by checking the matrix row and col
710             {
711                reOrder(state,j)=1;  //makes the right-most state next state
712                j++;  //move to new row
713                if(j==mN)   //once matrix is done, re-order maj and exit function
714                {
715                   maj=maj*reOrder;
716                   return 0;
717                }
718             }
719          }
720       }
721    }
722    return 1;
723 }
724
```

```
725 MatrixXcd makeU(void)   //returns zero-energy states with Nambu-symmetry
726 {
727    MatrixXcd temp = MatrixXcd::Zero(4*nn,2*mN);
728    for(int i=0; i<2*nn; i++)
729    {
730       for(int j=0; j<mN; j++)
731       {
732          temp(i, mN+j)=maj(i,j); //top right comes from top
733          temp(2*nn+i, mN+j)=maj(2*nn+i,j);   //bottom right comes from ↩
                bottom
734          temp(i, mN-1-j)=conj(maj(2*nn+i,j));   //top left comes from ↩
                bottom
735          temp(2*nn+i, mN-1-j)=conj(maj(i,j));   //bottom left comes from ↩
                top
736       }
737    }
738    return temp;
739 }
740
741 //performs RK4 with mN vectors
742 void rk4(void)
743 {
744    MatrixXcd k1 = MatrixXcd::Zero(4*nn,mN);
745    MatrixXcd k2 = MatrixXcd::Zero(4*nn,mN);
746    MatrixXcd k3 = MatrixXcd::Zero(4*nn,mN);
747    MatrixXcd k4 = MatrixXcd::Zero(4*nn,mN);
748    complex<double> I (0.0,-1.0*stepSize); //-i and step size combined for↩
         efficiency
749    int dur=prm.tSteps*prm.TT;
750    for(int tt=0; tt<dur; tt+=2)
751    {
752       k1 = I*sparseHam*maj;
753       update_ham((tt+1.0)*stepSize);
754       k2 = I*sparseHam*(maj+k1);
755       k3 = I*sparseHam*(maj+k2);
756       update_ham((tt+2.0)*stepSize);
757       k4 = I*sparseHam*(maj+2.0*k3);
758       maj=maj+(k1+2.0*k2+2.0*k3+k4)/3.0;
759       if((tt+2)%(prm.tSteps*prm.dispRate)==0)      //only display every ↩
             time unit
760       {
761          Usub=makeU();
762          Td=(Uinit.adjoint())*Usub;
763          disp((tt+2.0)*stepSize);
764       }
765    }
766 }
767
768 int powInt(int x, int y)   //overload pow for integers
769 {
770    if(x==0)
771       return 0;
772    if(y==0)
773       return 1;
774    int temp=x;
775    for(int i=1;i<y;i++)
776       temp*=x;
777    return temp;
778 }
779
```

```
780 int biForm(int i) //converts decimal to binary
781 {
782    int x=0;
783    for(int j=mN;j>0;j--)
784    {
785       if(i>=powInt(2,j-1))
786       {
787          x+=powInt(10,j-1);
788          i-=powInt(2,j-1);
789       }
790    }
791    return x;
792 }
793
794 void dispBi(int i)
795 {
796    printf("%0*d", mN, i);  //% flag to pad with 0's to form * digits with↩
       *=mN
797 }
798
799 int decForm(int i, int length)   //converts binary with length bits to ↩
    decimal
800 {
801    int x=0;
802    for(int j=length;j>0;j--)
803    {
804       if(i>=powInt(10,j-1))
805       {
806          x+=powInt(2,j-1);
807          i-=powInt(10,j-1);
808       }
809    }
810    return x;
811 }
812
813 int partNum(int i, int length)   //returns number of particles in the ↩
    first length bits
814 {
815    int p=0;
816    for(int j=mN;j>(mN-length);j--)  //runs loop to check length digits on↩
       the left
817    {
818       if(i>=powInt(2,j-1))
819       {
820          p++;
821          i-=powInt(2,j-1);
822       }
823       //    cout<<powInt(2,j-1)<<'\t'<<x<<'\t'<<i<<endl;
824    }
825    return p;
826 }
827
828 void multBasis(void) //displays multiparticle basis
829 {
830    MatrixXcd temp = MatrixXcd::Zero(size,size);
831    int rowE=0, colE=0, rowO=size/2, colO=size/2;
832    int c,r;
833    for(int i=0;i<size;i++)
834    {
```

```
835         if(partNum(i,mN)%2==0)  //checks if the bra of matrix element is ↩
                even
836         {
837            r=rowE;
838            rowE++;
839         }
840         else
841         {
842            r=row0;
843            row0++;
844         }
845         for(int j=0;j<size;j++)
846         {
847            if(partNum(j,mN)%2==0)  //checks if the ket of matrix element is↩
                   even
848            {
849               c=colE;
850               colE++;
851            }
852            else
853            {
854               c=col0;
855               col0++;
856            }
857            //       cout<<"c is "<<c<<" and r is "<<r<<endl;
858            temp(r,c)=complex<double>(biForm(i),biForm(j));
859            //       printf("%0*d,%0*d  ", mN, biForm(i), mN, biForm(j));  ↩
                  //% flag to pad with 0's to form * digits with *=mN
860         }
861         colE=0;     //reset the column count for next row
862         col0=size/2;
863      }  //no need to reset since matrix is done
864      for(int i=0;i<size;i++)
865      {
866         cout<<"#  [ ";
867         for(int j=0;j<size;j++)
868         {
869            printf("%0*d,%0*d  ", mN, int(real(temp(i,j))), mN, int(imag(↩
                  temp(i,j)))); //% flag to pad with 0's to form * digits with↩
                   *=mN
870         }
871         cout<<"]\n";
872      }
873   }
874
875   int movesNeeded(int b, int k) //returns number of operators needed to ↩
       match bra with ket
876   {
877      int xB, xK, x=0;
878      for(int j=mN;j>0;j--)
879      {
880         xB=0;
881         xK=0;
882         if(b>=powInt(2,j-1))
883         {
884            xB=1;
885            b-=powInt(2,j-1);
886         }
887         if(k>=powInt(2,j-1))
888         {
```

```
889             xK=1;
890             k-=powInt(2,j-1);
891         }
892         if(xB!=xK)
893             x+=powInt(10,j-1);
894     }
895     return partNum(decForm(x,mN), mN);
896 }
897
898 complex<double> multDiag(int k, int i) //calculates diagonal terms for ↩
      multiparticle basis
899 {
900     complex<double> x=0.0;
901 /* dispBi(biForm(i));
902     cout<<'\t';
903     dispBi(biForm(i));
904 */
905     for(int j=mN;j>0;j--)   // j iterates thru the bit form left to right
906     {
907         if(i>=powInt(2,j-1))
908         {
909             x+=conj(Td(k,mN-1+j))*Td(k,mN-1+j);
910             i-=powInt(2,j-1);
911 //          cout<<'\t'<<conj(Td(k,mN-1+j))*Td(k,mN-1+j);
912         }
913         else
914         {
915             x+=conj(Td(k,mN-j))*Td(k,mN-j);
916 //          cout<<'\t'<<conj(Td(k,mN-j))*Td(k,mN-j);
917         }
918     }
919 // cout<<endl<<endl;
920     return x;
921 }
922
923 //calculates coefficients for normal ordered term x, and complement y of ↩
      multiparticle basis
924 void checkCre(complex<double> &x, complex<double> &y, int k, int bra, int↩
      ket)
925 {
926     int xKet, xBra;   //int to hold the bits of the bra and ket
927     int secCreate=0;
928     int secAnn=0;
929     int tempBra=bra;
930     int tempKet=ket;  //use temp ints to keep bra and ket for second use ↩
          if needed
931 /* dispBi(biForm(bra));
932     cout<<'\t';
933     dispBi(biForm(ket));
934     cout<<endl;
935 */
936     for(int j=mN;j>0;j--)   //starting from the left
937     {
938         xBra=0;  //default binary digit is 0, unoccupied
939         xKet=0;
940         if(tempBra>=powInt(2,j-1)) //check if the bit of bra state is ↩
              occupied
941         {
942             xBra=1;  //if so, change the digit to 1
943             tempBra-=powInt(2,j-1); //and change tempBra for next bit test
```

```
944         }
945         if(tempKet>=powInt(2,j-1))
946         {
947            xKet=1;
948            tempKet-=powInt(2,j-1);
949         }
950         if((xBra-xKet)==1)   //check if bra is occupied and ket isn't
951         {
952            if((partNum(bra,mN-j)%2)==1)  //change sign if odd number of ↩
                fermion exhanges needed
953            {
954               x*=-1;   //i.e number of 1's to the left of the position ↩
                   being checked in the bra
955               y*=-1;
956            }
957            if(secCreate==0)  //if first creation operator used
958            {
959               x*=conj(Td(k,mN+j-1));  //positive matrix entry for T_i*
960               y*=Td(k,mN-j); //corresponding negative entry for T_-i
961               secCreate++;
962               bra-=powInt(2,j-1);
963 /*            cout<<"Bra\tKet\tx\ty\tT_i*\tT_-i"<<endl;
964               dispBi(biForm(bra));
965               cout<<'\t';
966               dispBi(biForm(ket));
967               cout<<'\t'<<x<<'\t'<<y<<'\t'<<conj(Td(k,mN+j-1))<<'\t'<<Td(k,↩
                   mN-j)<<endl;
968 */
969            }
970            else  //if second creation operator
971            {
972               x*=Td(k,mN-j); //negative entry T_-j
973               y*=conj(Td(k,mN+j-1));  //T_j*
974               bra-=powInt(2,j-1);
975 /*            cout<<"Bra\tKet\tx\ty\tT_-j\tT_j*"<<endl;
976               dispBi(biForm(bra));
977               cout<<'\t';
978               dispBi(biForm(ket));
979               cout<<'\t'<<x<<'\t'<<y<<'\t'<<Td(k,mN-j)<<'\t'<<conj(Td(k,mN+↩
                   j-1))<<endl;
980 */
981            }
982         }
983         if((xKet-xBra)==1)   //check if ket is occupied and bra isn't
984         {
985            if((partNum(ket,mN-j)%2)==1)  //change sign if odd number of ↩
                fermion exhanges needed
986            {
987               x*=-1;   //i.e number of 1's to the left of the position ↩
                   being checked in the ket
988               y*=-1;
989            }
990            if(secAnn==0)  //if first annihilation operator used
991            {
992               x*=Td(k,mN+j-1);  //positive matrix entry for T_j
993               y*=conj(Td(k,mN-j)); //corresponding negative entry for T_-j*
994               ket-=powInt(2,j-1);
995               secAnn++;
996 /*            cout<<"Bra\tKet\tx\ty\tT_j\tT_-j*"<<endl;
997               dispBi(biForm(bra));
```

```
998             cout<<'\t';
999             dispBi(biForm(ket));
1000            cout<<'\t'<<x<<'\t'<<y<<'\t'<<Td(k,mN+j-1)<<'\t'<<conj(Td(k,↩
                    mN-j))<<endl;
1001 */
1002         }
1003         else  //if second annihilation operator
1004         {
1005            x*=conj(Td(k,mN-j)); //negative entry T_-i*
1006            y*=Td(k,mN+j-1);  //T_i
1007            ket-=powInt(2,j-1);
1008 /*         cout<<"Bra\tKet\tx\ty\tT_-i*\tT_i"<<endl;
1009            dispBi(biForm(bra));
1010            cout<<'\t';
1011            dispBi(biForm(ket));
1012            cout<<'\t'<<x<<'\t'<<y<<'\t'<<conj(Td(k,mN-j))<<'\t'<<Td(k,mN↩
                   +j-1)<<endl;
1013 */
1014         }
1015      }
1016   }
1017 }
1018
1019 MatrixXcd multProj(int k)  //returns N matrix in multiparticle basis
1020 {
1021    MatrixXcd temp = MatrixXcd::Zero(size,size);
1022    int rowE=0, colE=0, rowO=size/2, colO=size/2;
1023    int c,r;
1024    complex<double> x;
1025    complex<double> y;
1026    for(int i=0;i<size;i++)
1027    {
1028       if(partNum(i, mN)%2==0) //checks if the bra of matrix element is ↩
               even
1029       {
1030          r=rowE;
1031          rowE++;
1032       }
1033       else
1034       {
1035          r=rowO;
1036          rowO++;
1037       }
1038       for(int j=0;j<size;j++)
1039       {
1040          x=1.0;   //reset x and y for each bra and ket
1041          y=1.0;
1042          if(partNum(j, mN)%2==0) //checks if the ket of matrix element is↩
                  even
1043          {
1044             c=colE;
1045             colE++;
1046          }
1047          else
1048          {
1049             c=colO;
1050             colO++;
1051          }
1052 //       cout<<"c is "<<c<<" and r is "<<r<<endl;
1053          if(i==j)
```

```
1054            {
1055               temp(r,c)=multDiag(k,i);
1056            }
1057            else if(movesNeeded(i,j)==2)
1058            {
1059               checkCre(x,y,k,i,j);
1060 //            cout<<endl;
1061               temp(r,c)=x-y;
1062            }
1063            else
1064               temp(r,c)=0;
1065         }
1066         colE=0;     //reset the column count for next row
1067         colO=size/2;
1068      } //no need to reset since matrix is done
1069      return temp;
1070 }
1071
1072 void findProb(void)  //finds probabilities using all combos of projectors↩
       and initState
1073 {
1074    MatrixXcd temp;   //matrix to hold projector product
1075    int rowE=0, rowO=size/2;   //places even probabilties in top half
1076    int r;   //holds the row after adjusting for even and odd sectors
1077    int tempI;  //temp iterator to store value of i
1078    for(int i=0;i<size;i++) //i iterates thru the
1079    {
1080       temp = MatrixXcd::Identity(size,size); //initialize matrix
1081       if(partNum(i,mN)%2==0)  //checks if the row is even
1082       {
1083          r=rowE;
1084          rowE++;
1085       }
1086       else
1087       {
1088          r=rowO;
1089          rowO++;
1090       }
1091 //    cout<<"i="<<i<<'\t'<<"r="<<r<<endl;
1092       tempI=i;
1093       //determine projectors for each state
1094       for(int j=mN;j>0;j--)   //starts from the left
1095       {
1096          if(tempI>=powInt(2,j-1))   //determine if jth bit is 1
1097          {
1098 //          cout<<"tempI="<<tempI<<"  j=1"<<'\t'<<mN-1+j<<endl;
1099             temp*=multProj(mN-1+j); //if 1, use positive projector
1100             tempI-=powInt(2,j-1);   //decreases i to find binary form, ↩
                   but doesn't affect i
1101          }
1102          else  //if jth bit is 0
1103          {
1104             temp*=multProj(mN-j);   //if 0, use negative projector
1105 //          cout<<"tempI="<<tempI<<"  j=0"<<'\t'<<mN-j<<endl;
1106          }
1107       }
1108 //    cout<<endl;
1109       prob(r)=( (initState.adjoint())*temp*initState );
1110    }
1111 }
```

```
1112
1113 /* prints maj for future use so diagonalization isn't repeated for the ↩
     same parameters
1114  * then file is inserted as a header
1115  */
1116 void copyMaj(void)
1117 {
1118    for(int i=0; i<4*nn; i++)
1119    {
1120       for(int j=0; j<mN; j++)
1121       {
1122          printf("maj(%d,%d)=%13.12g; ", i, j, real(maj(i,j)));
1123       }
1124    }
1125    cout<<endl;
1126 }
1127
1128 int main (int argc, char *argv[])
1129 {
1130     init_params(argc, argv);
1131    init_ham(0.0);
1132    init_ham_sparse(0.0);
1133    if(prm.per==1)
1134       per_ham(0.0);
1135    sparseHam.setFromTriplets(tripletList.begin(), tripletList.end());
1136    if(prm.v>3)
1137    {
1138       do_solve();
1139       orderStates(1);    //argument is the column number of the parity ↩
           eigenvector
1140       copyMaj();
1141    }
1142    disp(0.0);
1143    if(prm.v>4)
1144       Uinit=makeU();
1145    if(prm.v>3)
1146       rk4();
1147    if(prm.v&128)
1148    {
1149       findProb();
1150       for(int i=0;i<size/2;i++)
1151       {
1152          cout<<real(prob(i))<<'\t';
1153       }
1154    }
1155    cout<<endl;
1156    return 0;
1157 }
```