# Quantum Information Analysis of Majorana Fermion Braiding: Exploring Realization and Error Assessment via a Series of Measurements

Ammar S. Alfaifi
Supervised by Dr. Raditya W. Bomantara

January 7, 2024

### Abstract

The report focuses on Majorana fermions, specifically exploring their distinction from Dirac fermions through mathematical articulation using second quantization. Majorana fermions, defined by identical creation and annihilation operators, exhibit non-abelian statistics, making them suitable for topological quantum computing. The study introduces a protocol to achieve braiding of Majorana fermions solely through a series of measurements, utilizing ancilla Majorana fermions. The impact of increasing ancilla fermions on braiding robustness is examined, and the protocol is translated into a quantum circuit, paving the way for simulating and assessing fidelity in a quantum computing context. The results suggest that incorporating more ancilla fermions may not necessarily enhance braiding fidelity, and the study concludes with a discussion on implementing the protocol on quantum hardware.

## 1 Introjection

In 1937, Majorana proposed a groundbreaking idea suggesting that electrically neutral spin-1/2 particles could be accurately described by a wave equation with real-valued solutions, known as the Majorana equation [4]. This proposition introduced the concept that these particles and their antiparticles would be indistinguishable due to the unique relationship between their wave functions through complex conjugation, ultimately leaving the Majorana wave equation unaltered.

Building upon Majorana's pioneering work, this paper delves into the fascinating realm of Majorana fermion braiding, transitioning from conventional approaches to a novel method centred around a sequence of measurements. A key aspect of this exploration involves leveraging the Jordan-Wigner transformation to express Majorana operators within the framework of a spin (fermionic) system. This transformative step not only enhances the understanding of Majorana fermions but also opens up possibilities for utilizing advanced techniques provided by the `qiskit` package.

In the realm of two dimensions, executing two exchanges of particle positions mirrors the topological equivalence of one quasiparticle encircling another along a closed path, known as a braid. This action of exchanging identical particles or manoeuvring one particle around another is formally termed "braiding" in mathematical terms. The act of "braiding" two anyons not only encapsulates a historical record of the event but also results in altered wave functions that effectively "count" the number of braids involved.[5]

Furthermore, the introduction of the braiding operator adds another layer of complexity and significance to the discussion. The braiding operator is a mathematical entity that encapsulates the non-Abelian statistics associated with the braiding of anyons, a fundamental concept in the study of topological quantum computation. Exploring the properties and applications of the braiding operator provides valuable insights into the behaviour of Majorana fermions during braiding processes.

Fault-tolerant quantum computing is of particular interest due to the inherent challenges posed by the delicate nature of quantum information. Unlike classical computation, which can often rely on the reliability of classical gates, quantum systems are highly susceptible to errors and decoherence. In the classical realm, information storage on magnetic media, for instance, benefits from the collective behaviour of spins in individual atoms. Although each spin is sensitive to thermal fluctuations, the interactions between spins tend to align them in the same direction. Consequently, if a spin deviates, the interactions compel it to return to the alignment of other spins, resembling a self-correcting mechanism akin to the repetition code in error correction.[2]

Quantum computers that operate based on measurements employ a sophisticated quantum state referred to as a cluster state. This state comprises three (or more) qubits interconnected through the non-classical phenomenon of entanglement, where the properties of quantum particles remain correlated even across vast distances. The functionality of measurement-based quantum computers revolves around initiating a measurement on the initial qubit within the cluster state. The result of this measurement dictates the subsequent measurement on the second entangled qubit, a process known as feedforward. Subsequently, this determination influences how the third qubit is measured. Through this approach, any desired quantum gate or circuit can be realized by carefully selecting the sequence of measurements.[7] Motivation behind utilizing a measurement-based protocol for realizing the braiding operator deserves attention. By adopting a measurement-based approach, the study aims to exploit the inherent advantages offered by this methodology, such as fault tolerance and scalability.

Topological properties play a pivotal role in hosting quantum computation and Majorana braiding. The unique characteristics of topological systems, such as robustness against local perturbations, make them ideal candidates for quantum information processing. In the context of quantum gates simulation, the discussion extends to the utilization of measurement-only approaches for simulating quantum gates. Specifically, exploring how measurement-only strategies can effectively simulate quantum gates is crucial for comprehending the intricacies involved in the proposed Majorana fermion braiding scheme. Drawing parallels to existing quantum computing platforms, like the IBM Q computer, underscores the practicality and feasibility of implementing such schemes in real-world quantum hardware.

The exploration of accessing and running circuits on platforms such as the IBM Q computer is a crucial aspect. It is imperative to comprehend the processes and mechanisms involved in accessing and executing circuits on quantum computing hardware, as this knowledge forms the foundation for translating theoretical advancements into practical applications. In this report, we successfully employed the `qiskit` library and IBM online quantum computers to simulate our conceptual framework. This involved the creation of a two-qubit system that realizes our sequence of measurements through Pauli unitary gates. Through a comparison with a true braiding operator for a single qubit system, we assessed the results and error sensitivity of such circuits by calculating the average fidelity between the final state vector of the circuit after execution.

2

# 2 Majorana Definition & Non-abelian Theory

The distinction between Majorana fermions and Dirac fermions can be mathematically articulated through the creation and annihilation operators of second quantization. Specifically, the creation operator $\gamma_j^\dagger$ generates a fermion in quantum state $j$ (described by a real wave function), while the annihilation operator $\gamma_j$ eliminates it, or equivalently, creates the corresponding antiparticle. In the case of a Dirac fermion, the operators $\gamma_j^\dagger$ and $\gamma_j$ are distinct, whereas for a Majorana fermion, they are identical. Expressing the ordinary fermionic annihilation and creation operators $f$ and $f^\dagger$ in terms of two Majorana operators $\gamma_1$ and $\gamma_2$ can be achieved as follows:

$$\hat{a} = \frac{1}{\sqrt{2}}\left(\gamma_1 + i\gamma_2\right),$$

$$\hat{a}^\dagger = \frac{1}{\sqrt{2}}\left(\gamma_1 - i\gamma_2\right).$$

Having an even number of Majorana fermions, $2n$, they obey the anitcommutation relation

$$\{\gamma_i, \gamma_j\} = 2\delta_{ij}, \qquad i = 1, 2, \ldots, 2n,$$
$$\gamma_i^2 = 1. \tag{1}$$

Mathematically, the superconductor imposes electron-hole "symmetry" on the quasiparticle excitations, relating the creation operator $\gamma(E)$ at energy $E$ to the annihilation operator $\gamma^\dagger(-E)$ at energy $-E$.[8] Majorana fermions can be bound to a defect at zero energy, and then the combined objects are called Majorana bound states or Majorana zero modes. This name is more appropriate than Majorana fermion (although the distinction is not always made in the literature) because the statistics of these objects are no longer fermionic. Instead, the Majorana bound states are an example of non-abelian anyons: interchanging them changes the state of the system in a way that depends only on the order in which the exchange was performed. The non-abelian statistics that Majorana bound states possess allow them to be used as a building block for a topological quantum computer.

Consider the wavefunctions $\psi_1$ and $\psi_2$ associated with two indistinguishable particles. Indistinguishable particles give

$$|\langle\psi_1\psi_2|\psi_2\psi_1\rangle|^2 = 1.$$

Upon applying for an exchange operator, one anticipates the form

$$\hat{B}|\psi_1\,\psi_2\rangle = e^{i\phi}|\psi_2\,\psi_1\rangle, \tag{2}$$

where for fermions, $\phi = \pi \to -1$, and for bosons, $\phi = 0 \to +1$—both being constant. However, in the case of Majorana fermions, as we will elaborate on shortly, the constant takes the form of a matrix. This distinction categorizes the former as abelian particles and the latter as non-abelian particles [6]. The parity operator plays a crucial role in describing the non-abelian statistics associated with the braiding of these particles. The braiding of Majorana fermions involves exchanging their positions in space, and the resulting transformation is characterized by non-trivial matrix elements rather than simple constants.

To assign a state to a system of Majorana fermions, we need to consider them as a fermionic system, i.e., as pairs of the form of Equation 2. In general, for $2N$ Majorana modes will have $N$ fermionic modes. Each mode has the flexibility to be occupied or unoccupied by a fermion, resulting

in two potentially degenerate quantum states $|0\rangle$ and $|1\rangle$ for every pair of Majoranas. Consider the the *fermion parity* operator

$$\hat{P} = 1 - 2\hat{N} = i\gamma_1\gamma_2, \tag{3}$$

where $N = \hat{a}^\dagger\hat{a}$ is the *fermion number* operator. Let's apply it to a pair of Majoranas states as

$$\hat{P}|0\rangle = (1 - 2\hat{N})|0\rangle = +|0\rangle,$$
$$\hat{P}|1\rangle = (1 - 2\hat{N})|1\rangle = -|1\rangle.$$

The parity operator is characterized by eigenvalues of $+1$ and $-1$, hence its name. The Hilbert space associated with parity can be treated as comparable to a qubit, functioning as a subspace within a qubit for information storage and quantum computation purposes. This non-local operator, endowed with robustness against errors in quantum computation, ensures stability over significant distances. Additionally, the preservation of the total parity within a system of Majorana particles becomes evident. Consequently, the application of unitary operators is constrained to those maintaining the conservation of total parity. This insight leads to the realization that the braiding of Majorana particles adheres to this conservation principle.

In general, any unitary operator $U$ can be expressed in terms of a Hermitian operator $A$ of the form $U = e^{i\beta A}$ with some angle $\beta$. For Majorana fermion operators, we can begin with their parity operator, denoted as $P_{nm} = i\gamma_n\gamma_m$. This allows us to define

$$U \equiv e^{\beta\gamma_n\gamma_m} \quad \text{or} \quad U = \cos\beta + \gamma_n\gamma_m\sin\beta, \tag{4}$$

see Appendix A for a proof of this result. To find a unitary operator that evolves Majorana fermions according to the Heisenberg picture, i.e.,

$$\gamma_n \to U\gamma_n U^\dagger,$$
$$\gamma_m \to U\gamma_m U^\dagger,$$

while leaving other elements unaffected, we substitute Equation 4 into the transformation equation:

$$\gamma_n \to \gamma_n\cos 2\beta - \gamma_m\sin 2\beta,$$
$$\gamma_m \to \gamma_m\cos 2\beta + \gamma_n\sin 2\beta.$$

This leads to the condition $\beta = \pm\pi/4$-through the text, we will refer to $\pi/4$ as the braiding operator and $-\pi/4$ as the inverse braiding operator. Substituting this into the expression, we obtain the braiding unitary operator for $\gamma_n$ and $\gamma_m$:

$$U = \exp\left(\pm\frac{\pi}{4}\gamma_n\gamma_m\right) = \frac{1}{\sqrt{2}}\left(1 \pm \gamma_n\gamma_m\right).$$

In case of two-level fermionic systems, to make projective measurements along $z$-axis we apply the following operator

$$(1 + S\sigma_z),$$

where $S$ can be $\pm 1$ with some probability, corresponding to the eigenvalues of $\sigma_z$. The analogy operator for Majorana fermions to a projective measurement will be

$$(1 + Si\gamma_n\gamma_m),$$

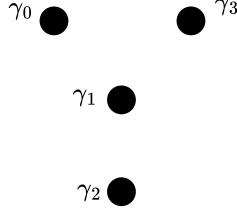where again $S$ can be $+1$ or $-1$.

4

Figure 1: Configuration of four Majorana fermions.

# 3 Realizations of Braiding Via a Series of Measurements

Consider the configuration of Majorana fermions depicted in Figure 1. And, say, we want to braid the two Majorana fermions $\gamma_0$ & $\gamma_3$. The conceptual approach here involves executing measurement-only operations on the configuration of Majorana modes, effectively achieving the same outcome as physically braiding the two Majorana fermions [3]. To illustrate this idea, we begin with a system of 4 Majorana fermions, corresponding to two fermions. The configuration of Majorana fermions is depicted in Figure 1. The true braiding operator between $\gamma_0$ and $\gamma_3$ is given by

$$U = e^{\frac{\pm\pi}{4}\gamma_3\gamma_0} = \frac{1}{\sqrt{2}}(1 \pm \gamma_3\gamma_0). \tag{5}$$

We call $\gamma_0$ and $\gamma_3$ the main Majorana fermions, and the rest are called the ancilla Majorana fermions. By utilizing the ancilla Majorana fermions we want to realize the braiding operator in Equation 5, using only measurement operations. We do not want to do any measurement on both main Majorana fermions. Further, the first step and last step should not involve the main Majoranas.

To realize this braiding operator solely through a series of measurements, we follow these four steps:

1. Apply the operator $(1 + iS_0\gamma_1\gamma_2)$

2. Apply the operator $(1 + iS_1\gamma_1\gamma_0)$

3. Apply the operator $(1 + iS_2\gamma_3\gamma_1)$

4. Apply the operator $(1 + iS_3\gamma_1\gamma_2)$

as illustrated in Figure 2.

Now we need to verify this result whether it realizes the braiding operator. We start by writing all steps as one multiplication series from right to left, as

$$(1 + iS_3\gamma_1\gamma_2)(1 + iS_2\gamma_3\gamma_1)(1 + iS_1\gamma_1\gamma_0)(1 + iS_0\gamma_1\gamma_2)$$

To simplify, we note the left-most projector and the right-most one are different only in $S$, if we could bring them together we can use some helpful properties of projectors. First, expand the middle two projectors

$$(1 + iS_3\gamma_1\gamma_2)[1 + iS_2\gamma_3\gamma_1 + iS_1\gamma_1\gamma_0 - S_2S_1\gamma_3\gamma_0](1 + iS_0\gamma_1\gamma_2),$$
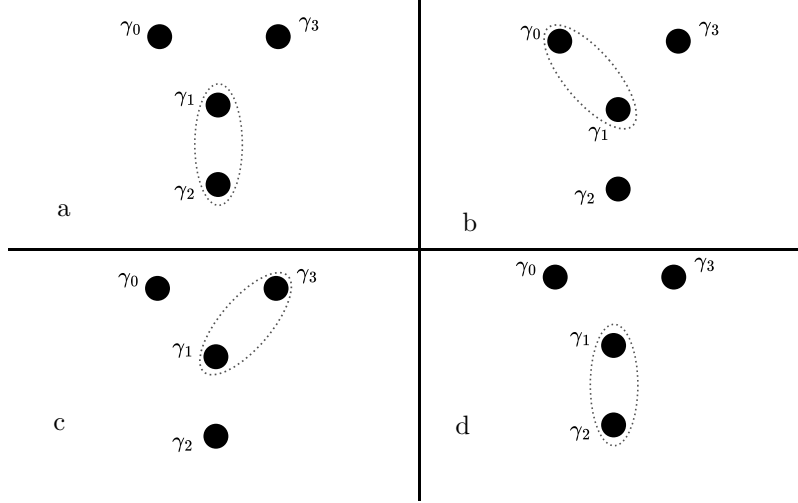
5

Figure 2: a-d shows measurements sequence that eventually realizes braiding of $\gamma_0$ and $\gamma_3$.

we used $\gamma_n^2 = 1$ on the last term. We employ the operators on the sides to encapsulate each term within the brackets. Starting with the first terms we get

$$(1 + iS_3\gamma_1\gamma_2)(1 + iS_0\gamma_1\gamma_2).$$

We have two cases; $S_3 = S_0$, then we get just $(1 + iS_0\gamma_1\gamma_2)$, since projectors are idempotent (i.e., $P^2 = P$). And if $S_3 \neq S_0$, it vanishes; since getting a measurement output $S_0$, then trying to get a measurement on another output will always result in zero. In compact, we write

$$\delta_{S_3, S_0}(1 + iS_0\gamma_1\gamma_2).$$

Moving to the second term, we get

$$(1 + iS_3\gamma_1\gamma_2)iS_2\gamma_3\gamma_1(1 + iS_0\gamma_1\gamma_2).$$

As in Equation 2, we exploit the property that all Majorana fermions anticommute with each other, and commute with themselves. Then, by moving the enclosed factor to the right we introduce a minus sign,

$$(1 + iS_3\gamma_1\gamma_2)(1 - iS_0\gamma_1\gamma_2)iS_2\gamma_3\gamma_1.$$

In terms of the Kronecker Delta, we have

$$\delta_{S_3, -S_0}(1 - iS_0\gamma_1\gamma_2)iS_2\gamma_3\gamma_1.$$

We continue the same steps for the remaining terms, adding them app we get

$$\delta_{S_3, S_0}(1 + iS_0\gamma_1\gamma_2) + \delta_{S_3, -S_0}(1 - iS_0\gamma_1\gamma_2)iS_2\gamma_3\gamma_1$$
$$+ \delta_{S_3, -S_0}(1 - iS_0\gamma_1\gamma_2)iS_1\gamma_1\gamma_0 + \delta_{S_3, S_0}(1 + iS_0\gamma_1\gamma_2)S_2S_1\gamma_3\gamma_0.$$

Factoring out the same Kroneckers and projectors

$$
\begin{aligned}
&\delta_{S_3,S_0}\,(1 + iS_0\gamma_1\gamma_2)(1 + S_2S_1\gamma_3\gamma_0) \\
&+ \delta_{S_3,-S_0}\,(1 - iS_0\gamma_1\gamma_2)(iS_2\gamma_3\gamma_1 + iS_1\gamma_1\gamma_0).
\end{aligned}
\tag{6}
$$

Since we are concerned about the main Majorana fermions ($\gamma_3$ & $\gamma_0$), we ignore the leftmost projector; it just affects the the ancilla Majorana fermions. If $S_3 = S_0$, we get

$$(1 + S_2S_1\gamma_3\gamma_0);$$

but $S_2S_1 = \pm 1$, thus

$$(1 \pm \gamma_3\gamma_0).$$

That's it. We realize the braiding operator in Equation 5. Based on the measurement outcomes, values of $S_0$ through $S_3$, we can get knowledge of the resulting overall operation as well as how to tune it gives the braiding operator. Now consider the case where $-S_3 = S_0$. We have

$$(iS_2\gamma_3\gamma_1 + iS_1\gamma_1\gamma_0) = -iS_2\gamma_1\,(\gamma_3 - S_1S_2\gamma_0),$$

where $S_1S_2 = \pm 1$. Since the global phase factor has no physically observable consequences, we can ignore the factor $-iS_2$. For $\gamma_1$, again its effect is on the ancilla Majorana fermion. However, to realize a braiding operator we have to multiply by $\gamma_3$, to get

$$-iS_2\gamma_1\,(1 \mp \gamma_3\gamma_0).$$

By implementing a protocol that reads the values of the measurement outcomes we can, predictably, realize the braiding of $\gamma_3$ & $\gamma_0$ shown in Equation 5.

As observed, employing the ancilla as an auxiliary resource for Majorana fermions allowed us to implement the braiding operator solely through a series of measurements. Does the augmentation of these ancilla quantities contribute to the robustness and interrelation of the braiding process? This concept needs to be extended to accommodate any number of ancilla Majorana fermions. To explore this, we begin with a configuration involving a total of 8 Majorana fermions, as depicted in Figure 3. So the series of measurements will be, written from right to left,

$$
\begin{aligned}
&(1 + iS_0\gamma_1\gamma_2)(1 + iS_1\gamma_3\gamma_4)(1 + iS_2\gamma_0\gamma_1)(1 + iS_3\gamma_2\gamma_3) \\
&\times (1 + iS_4\gamma_1\gamma_3)(1 + iS_5\gamma_2\gamma_5)(1 + iS_6\gamma_1\gamma_2)(1 + iS_7\gamma_3\gamma_4).
\end{aligned}
\tag{7}
$$

By automating the generation of measurement sequences for any number of Majorana fermions, we aim to investigate the impact of employing a larger number of ancilla Majorana fermions on achieving a more robust and fault-tolerant braiding process. However, to conduct these studies, it is imperative to transform the system into a fermionic one, leveraging established methods for running and simulating such processes. The Jordan-Wigner transformation, detailed in the next section, facilitates this conversion. Subsequently, a custom version of the concept of *state fidelity* will be employed to assess the proximity of the measurement-only braiding operator to the exact one. This involves obtaining the final matrix by multiplying all measurement operators, flattening it into a vector, normalizing it, and then computing the inner product with the true braiding operator using the expression
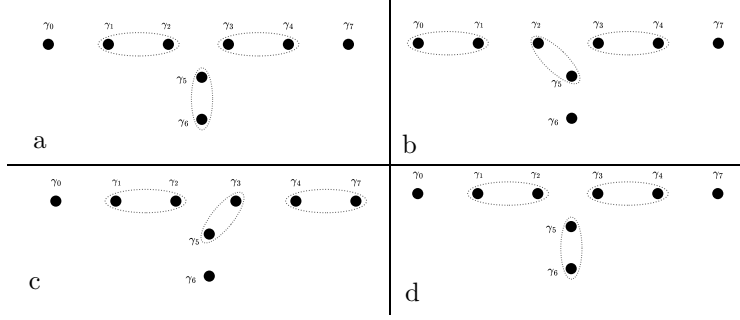
$$|\langle A|B\rangle|^2,$$

Figure 3: a-d shows measurements sequence that eventually realizes braiding of $\gamma_0$ and $\gamma_7$ for a system of eight Majorana fermions.

where $A$ represents the normalized flattened matrix resulting from the product of the series of projectors, and $B$ denotes the true braiding operator. See Appendix C for Python code to calculate the fidelity. To conduct a meaningful comparison of braiding on the primary Majoranas while mitigating the influence of the ancilla system, it is essential to perform a partial trace, isolating only the main Majorana fermions subsystem. Subsequently, the fidelity measure described above can be employed. This computational step is implemented programmatically using the `qutip` package, as detailed in the Appendix D.

We start implementing the sequences in Equation 6; for values of $S_i$, we use all possible combinations for now. As illustrated in Figure 4, in the ideal scenario of fidelity calculation, the results manifest as either 1, indicating that that specific combination of measurement outcomes ($S_i$) perfectly realizes the true braiding operator, or 0, signifying that it falls short of realizing the true braiding operator. Further, the figure tells us that one of the cases where the fidelity equals one is the combination at index 0. This is the trivial case where all $S_i = 1$, thus from Equation 7, the sequence

$$(1 + i\gamma_1\gamma_2)(1 + i\gamma_3\gamma_4)(1 + i\gamma_0\gamma_1)(1 + i\gamma_2\gamma_3)$$
$$\times (1 + i\gamma_1\gamma_3)(1 + i\gamma_2\gamma_5)(1 + i\gamma_1\gamma_2)(1 + i\gamma_3\gamma_4),$$

realizes the braiding of $\gamma_0$ & $\gamma_7$. We will be using this exact form for the rest of the section. Now back to the main question; will increasing the number of ancilla Majoranas help in the fidelity? Clearly, in the ideal case, nothing to improve, as seen in Figure 5. This is because we have been implicitly assuming infinite time for each measurement to take place. To model the real finite case, we rewrite each measurement projector as

$$e^{t(iS_i\gamma_n\gamma_m)}$$

By using the same method explained in Appendix A, we get

$$e^{t(iS_i\gamma_n\gamma_m)} = \cosh t + (i\gamma_n\gamma_m)\sinh t,$$

which implies

$$\cosh t + (i\gamma_n\gamma_m)\sinh t \to (1 + iS_i\gamma_n\gamma_m) \quad \text{as} \quad t \to \infty$$
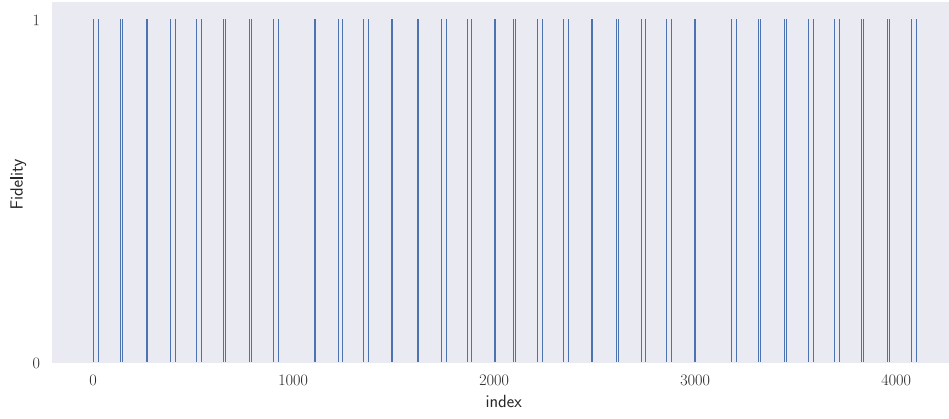
8

Figure 4: Shows the fidelity values, of all possible 4096 combinations of $S_i$ for eight Majoranas, again their index. This implements the sequence in Equation 7.
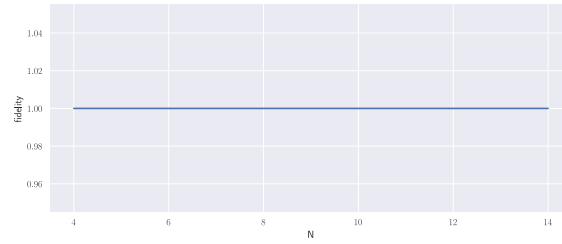


Figure 5: The fidelity of braiding operator realization plotted against the increasing number of employed ancilla Majoranas. The results are obtained using the sequence specified in Equation 7.
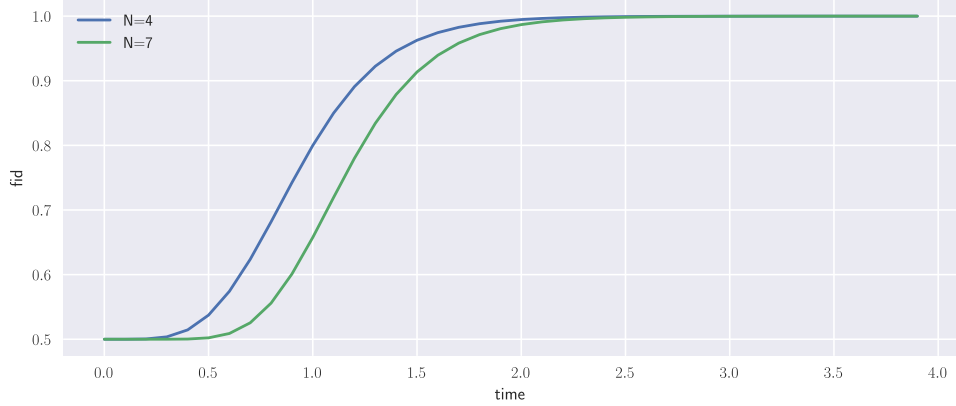
Figure 6: Applying the measurement sequence detailed in Equation 7, where the projectors take on a time-finite exponential form. The figure presents two plots corresponding to two different numbers of Majorana fermions.

Thus replace every projector in Equation 7, with this approximate projector in the simulating code, see Appendix E for this modification. The trends are evident in Figure 6, where two examples with a fixed number of Majorana fermions showcase a continual increase in fidelity as the duration for applying each measurement extends. In essence, granting a relatively prolonged time for the implementation of each measurement, as outlined in the modified sequence Equation 7, leads to a more accurate realization of the braiding operator.

Currently, we must employ this novel modelling of measurement operators to assess its impact as we incrementally increase the number of utilized ancilla Majorana fermions. In Figure 7, a counterargument emerges regarding the notion that increasing the number of utilized ancilla Majorana fermions enhances the realization of the braiding operator, consequently providing greater robustness to the measurement sequence. The plot illustrates that, contrary to this expectation, employing more Majorana fermions, while maintaining a fixed waiting time for each measurement application, results in lower fidelity. This observation leads to the conclusion that using as few Majorana fermions as ancilla produces a braiding that closely aligns with the true braiding operator.

In Figure 8, we examine the impact of introducing variable time intervals, ranging from -0.1 to +0.1, during the simulation. This not only accounts for variations in the waiting time for each measurement but also introduces noise into the process. Additionally, we present a reference scenario, which represents the ideal case without noise and utilizes approximate finite-time measurements.

# 4 Constructing Protocol & Simulating on Quantum Circuit

In this section, we will develop a quantum circuit as well as a protocol that can read the outcomes of each measurement to align the final projectors to be exactly *the* braiding operator. To achieve this, we redefine the series of measurements from the last section, for four Majorana fermions in total, in terms of a two-level fermionic system, i.e., qubits. We shall redefine the $\gamma$s in terms of fermionic spin operators, giving us a way to model this system in much more familiar systems, such as qubits
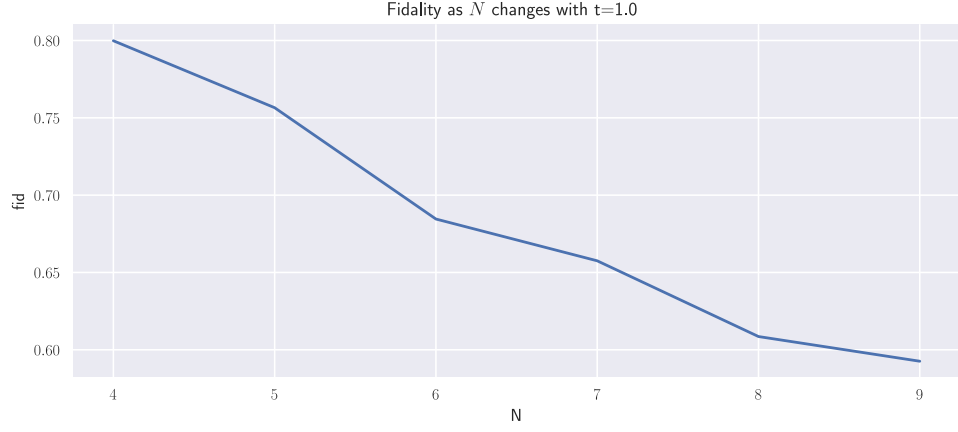
Figure 7: Illustrating fidelity under the approximate measurement operator, this plot depicts how fidelity varies with the changing number of utilized Majorana fermions. The analysis maintains a constant time value set at 1.
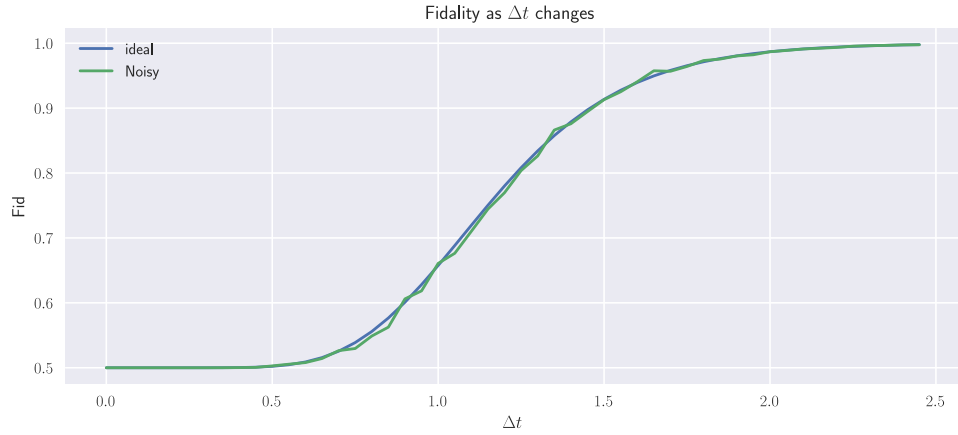


Figure 8: The figure displays two plots: one depicting the results of a noisy model with randomness in the waiting time for each measurement, and the other illustrating the ideal scenario similar to the previous ideal measurements—without the incorporation of the approximate finite-time form. The system size used here is $N = 7$.

in quantum computing information. So we'll have, based on Jordan-Wigner transformation [1],

- $\gamma_0 = Z_0$,

- $\gamma_1 = X_0 Z_1$,

- $\gamma_2 = X_0 Y_1$,

- $\gamma_3 = Y_0$,

Here, $X$, $Y$, and $Z$ represent the Pauli matrices associated with the two-level qubit, with the subscripts indicating the specific subsystem (qubit) upon which a Pauli gate is applied. It's important to note that the tensor product is implied; if only one gate is present, it is tacitly understood that it is tensor-producted with the identity of other subsystems. Consequently, the four-step series of measurements on the system, when the redefined $\gamma$ operators are substituted into the sequence originally designed for four Majorana fermions, becomes:

1. $(1 + iX_0 Z_1 X_0 Y_1) = (1 + X_1)$

2. $(1 + iX_0 Z_1 Z_0) = (1 + Y_0 Z_1)$

3. $(1 + iY_0 X_0 Z_1) = (1 + Z_0 Z_1)$

4. $(1 + X_1)$

Doing the same to redefine the true braiding operator in Equation 5, we get

$$e^{i\frac{\pi}{4}X_0} = \frac{1}{\sqrt{2}}\left(1 + iX_0\right) \quad \text{or} \quad e^{-i\frac{\pi}{4}X_0} = \frac{1}{\sqrt{2}}\left(1 - iX_0\right)$$

Thus, the sequence of measurements that realizes braiding of $\gamma_0$ & $\gamma_3$ in a spinful system is,

$$(1 + S_3 X_1)(1 + S_2 Z_0 Z_1)(1 + S_1 Y_0 Z_1)(1 + S_0 X_1) \tag{8}$$

Let's understand the possible outcomes from the general case of the measurement operator. Expanding the middle two factors as

$$(1 + S_3 X_1)(1 + S_2 Z_0 Z_1 + S_1 Y_0 Z_1 + S_2 S_1 Z_0 Z_1 Y_0 Z_1)(1 + S_0 X_1)$$

Utilizing the Pauli gates anitcommutation relations, we move the LHS factor to RHS, as for the first term we get

$$(1 + S_3 X_1)(1 + S_0 X_1) = \delta_{S_0, S_3}\left(1 + S_0 X_1\right)$$

For the second term,

$$(1 + S_3 X_1)S_2 Z_0 Z_1(1 + S_0 X_1) = \delta_{S_0, -S_3} S_2 Z_0 Z_1(1 + S_0 X_1)$$

For the 3rd term,

$$(1 + S_3 X_1)S_1 Y_0 Z_1(1 + S_0 X_1) = \delta_{S_0, -S_3} S_1 Y_0 Z_1(1 + S_0 X_1)$$

12

For the 3rd term,

$$(1 + S_3X_1)S_2S_1Z_0Z_1Y_0Z_1(1 + S_0X_1) = \delta_{S_0,S_3} - iX_0S_2S_1(1 + S_0X_1)$$

Now, we'll investigate the protocol's classic outcomes, then we shall decide based on it whether we did realize a braiding between $\gamma_0$ & $\gamma_3$, if not, what operators to apply to fix it. From the previous paragraph, we simplify it to

$$[\delta_{S_0,S_3} + \delta_{S_0,-S_3} S_2Z_0Z_1 + \delta_{S_0,-S_3} S_1Y_0Z_1 + \delta_{S_0,S_3} - iX_0S_2S_1](1 + S_0X_1)$$

Let's study different cases:

**Case 1:** $S_0 = S_3$ We get

$$[1 - iX_0S_2S_1](1 + S_0X_1)$$

Note, the right factor just acts on subsystem 1 that we don't care about it outcomes.

**Case 1.1:** $S_1 = -S_2$

$$[1 + iX_0](1 + S_0X_1)$$

realizing counterclockwise braiding operator in Equation 4.

**Case 1.2:** $S_1 = S_2$

$$[1 - iX_0](1 + S_0X_1)$$

realizing clockwise braiding operator in Equation 4.

**Case 2:** $S_0 \neq S_3$ We get

$$[S_2Z_0Z_1 + S_1Y_0Z_1] (1 + S_0X_1)$$

let's factor out $Z_0Z_1$

$$Z_0Z_1[S_2 - iS_1X_0] (1 + S_0X_1)$$

In this case we always want to multiply by $Z_0$, then we'll have

**Case 2.1:** $S_1 = S_2$

$$S_1Z_0Z_1[1 - iX_0](1 + S_0X_1)$$

realizing the inverse braiding operator

**Case 2.2:** $S_1 = -S_2$

$$S_1Z_0Z_1[S_1S_2 - iX_0](1 + S_0X_1)$$

But $S_1S_2 = -1$, then

$$-S_1Z_0Z_1[1 + iX_0](1 + S_0X_1)$$

realizing the braiding operator

We move now to building the quantum circuits simulating the true braiding operator and its realization based on the above protocol. Using `qiskit`, we need to determine all unitary gates and measurements to apply, then apply our protocol to correct the statevector based on the readout values. According to Equation 8, we need to apply a measurement on the $x$-basis, but `qiskit` only provides a measurement operator along the computational basis (along $\sigma_z$ eigenvectors). We can
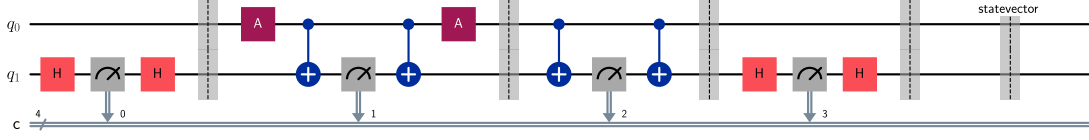
13

Figure 9: The figure illustrates the ultimate two-qubit circuit designed to implement the sequence of measurements outlined in Equation 8. The drawing highlights the four distinct steps of applying each measurement operator, delineated by vertical dashed lines. The final line indicates the point where we extract the reference final state vector of the circuit for subsequent comparisons.
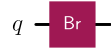


Figure 10: Illustration of the one-qubit circuit featuring a single unitary gate, denoted as Br, defined in Equation 9.

change the basis by enclosing any computational basis measurement by the Hadamard gate, named $H$,

$$H \rightarrow H(1 + Z)H^\dagger = (1 + HZH) = (1 + X),$$

where $H = H^\dagger$, i.e., Hermitian operator. For the joint measurement, $(1 + S_2 Z_0 Z_1)$, we can encapsulate the computational measurement by a controlled-X gate (i.e., CX or CNOT). Applying the same to the measurement $(1 + S_1 Y_0 Z_1)$, but with enclosing with a custom unitary gate to change the bases of $Z_0$ to $Y_0$, that is

$$A_0 = Y_0 + Z_0 \rightarrow A_0(1 + S_1 Z_0 Z_1)A_0^\dagger = (1 + S_1 Y_0 Z_1)$$

Armed with this information, our construction of the quantum circuit proceeds from left to right, aligning with the right-to-left order of measurements specified in Equation 8. This alignment mirrors the standard operation sequence in `qiskit` quantum circuits. Additionally, as indicated by the Jordan-Wigner transformation, it is observed that the inclusion of two qubits in the circuit is adequate for the required representation. As depicted in Figure 9, the circuit comprises four classical registers, each dedicated to storing the readout value following the application of a computational basis measurement. For instance, in the initial section of the circuit, we observe operations involving $H_1$, Measurement-Z, and $H_1$, collectively simulating the operator $(1 + S_i X_1)$. This pattern extends to the subsequent parts of the circuit as well.

For the true braiding circuit, so we can compare the end result of both circuits' statevectors, is as simple as one qubit applied to it the following redefined braiding operator in terms spinful system, from Equation 5,

$$\mathrm{Br} = e^{\frac{\pm\pi}{4}\gamma_3\gamma_0} = e^{\frac{\pi}{4}Y_0 Z_0} = e^{\frac{\pi}{4}iX_0}. \tag{9}$$

Evidently, there is a notable advantage in treating the primary Majorana fermions as a single pair, facilitating their mapping onto a single qubit and simplifying the isolation of the main Majoranas
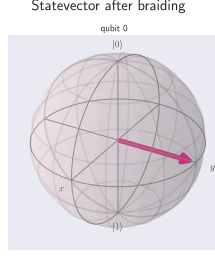
14

Figure 11: Visualization of the statevector for the single-qubit circuit subjected to the true braiding operator on the Bloch sphere.

subsystem. Refer to Figure 10, where we implemented this novel unitary gate within a single qubit circuit. Running the circuit only once suffices, given the absence of measurement operations, and the final circuit state vector is consistently predictable in a non-noisy environment.

The dashed vertical bar labelled 'statevector' in Figure 9 indicates the point at which we record the statevector for each run of the circuit. This recorded statevector is then processed through the protocol to validate the readout values and make corrections accordingly. Subsequently, we compare this statevector to the true-braiding circuit's statevector using fidelity; a fidelity closer to 1 indicates a higher alignment with the simulated sequence of measurements. However, a challenge arises as the first circuit represents a two-qubit system, resulting in a statevector representing the composite system—a $4 \times 1$ vector. To address this, we utilize the `qiskit` method called `partial_trace`. This method begins by transforming the statevector into a density matrix, given by

$$\rho = |\psi\rangle\langle\psi|.$$

However, a $2 \times 2$ density matrix possesses two eigenvalues (0 and 1), with the statevector of the main qubit corresponding to the non-zero eigenvalue. See Appendix F, for our implementation of the protocol and calculating the fidelity. We can visualize the ultimate statevectors resulting from the true braiding circuit and its simulated counterpart after the tracing operation on the Bloch sphere. The statevector of the braiding circuit, depicted in Figure 10, is illustrated in Figure 11. Our analysis of the measurement sequence indicates that the simulating circuit is capable of generating either the braiding operator or its inverse due to randomness. Consequently, the traced statevector of the first circuit either precisely aligns with the representation in Figure 11 or points in the opposite direction. And indeed, by running the code with this protocol we always match the braiding operator.

Given the correcting protocol running the first and using fidelity to compare the result with the second circuit, we are ready to simulate real-life conditions. Moreover, such circuits will always be exposed to different kinds of noise. Our job is to investigate the effect of this noise on the first circuit and redo the same analysis without a change in the protocol. The first type of error introduced by the `qiskit` package is the Pauli Error, which involves applying a Pauli gate before each occurrence of operations with a probability assigned between 0 and 1. A probability of 0.5 implies a 50% chance of applying a Pauli gate before an operation, while the remaining 50% involves applying an identity gate (effectively doing nothing to the circuit). As an example, simulating a
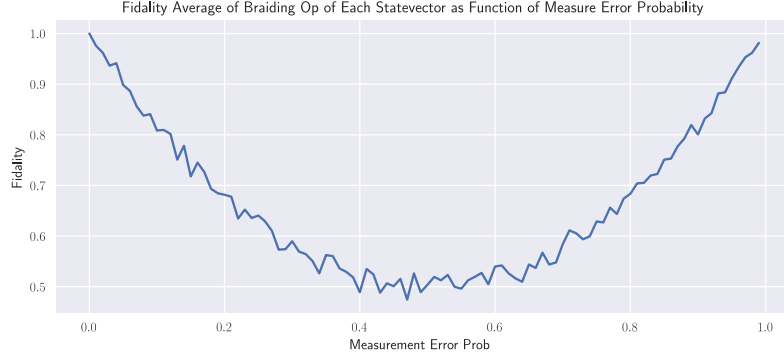
15

Figure 12: Illustration of the circuit in Figure 9 subjected to Pauli X error, with 1024 iterations for each probability setting. The resulting fidelities are averaged over the 1024 shots.
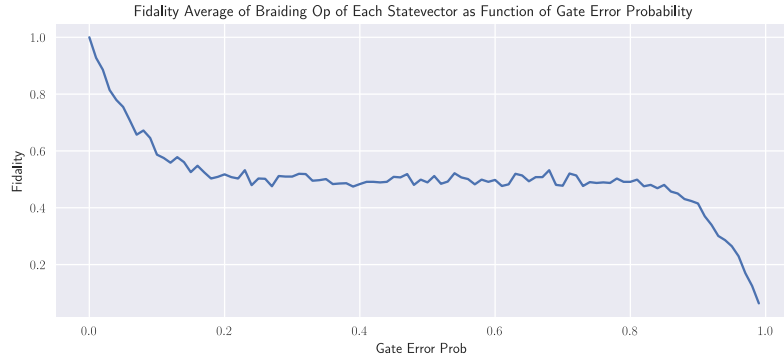


Figure 13: As in Figure 12, but with bit-flip applied to before each occurrence of `h` and `cx` operations.

bit-flip error in each measurement operation within the circuit involves applying an X gate before any measurement with a probability $P$. The outcome of this example is depicted in Figure 12. Notably, the plot exhibits an interesting symmetry around a 0.5 probability. This symmetry arises because if a bit flip is applied 100% of the time, it results in doing nothing—since there are four measurements, leading to four X gates, which collectively reduce to an identity operation.

Now, let's introduce another type of error, specifically targeting different operations, namely the h (Hadamard) and cx (CNOT) gates. In Figure 13, we observe the impact on the realization of braiding in the circuit when subjected to a bit-flip error on each occurrence of h and cx operations. It's important to note that cx is a two-qubit gate, so we extend the Pauli Error by taking the tensor product with itself to create a $4 \times 4$ matrix operator. Additionally, we observe that around the middle of the plot, there is an almost constant average fidelity of approximately 0.5. However, as the probability exceeds 0.8, the fidelity drops to zero, significantly impairing the realization of the braiding process.

# 5    Conclusion & Future Work

The paper delves into the realm of Majorana fermion braiding, exploring a novel method based on a sequence of measurements. Majorana fermions, initially proposed by Majorana in 1937, have unique properties, and their braiding plays a crucial role in topological quantum computation. The study introduces the braiding operator, a mathematical entity capturing non-Abelian statistics associated with anyon braiding.

The work employs qiskit for simulations on IBM Q computers, aiming to assess the feasibility of realizing braiding operators via a series of measurements. The study investigates the impact of errors, specifically Pauli errors, on the braiding process. The introduction of ancilla Majorana fermions as auxiliary resources is explored, with a focus on increasing robustness through fault-tolerant approaches.

The Jordan-Wigner transformation is utilized to represent Majorana operators within a qubit system. The paper then presents a protocol for realizing braiding operators solely through measurements, leveraging the conservation principle associated with fermion parity. The impact of employing different numbers of ancilla Majorana fermions on fidelity is examined, revealing unexpected results suggesting that fewer ancilla fermions may lead to higher fidelity.

The study further addresses the challenges of finite-time measurements, noise, and the incorporation of variable time intervals. The quantum circuit construction is detailed, with a specialized unitary gate representing the braiding operator in a single-qubit system. The correcting protocol is applied, and the fidelity between the true braiding circuit and its simulated counterpart is used for validation.

The investigation extends to real-life conditions, introducing Pauli X errors and exploring their effects on fidelity. Additionally, bit-flip errors on specific gates, such as h and cx, are analyzed. The study concludes with a comprehensive analysis of the impact of errors on the realization of braiding operators, providing valuable insights into the challenges and potential solutions in the field of Majorana fermion braiding for topological quantum computation.

Future work in this research could explore several avenues to enhance the depth and applicability of the study. Firstly, investigating additional error models beyond the Pauli errors, such as realistic noise profiles and decoherence, would provide a more comprehensive understanding of the challenges faced by Majorana fermion braiding in practical quantum computing environments. Moreover, the study could benefit from an exploration of error mitigation techniques and quantum error correction codes tailored to the unique characteristics of Majorana fermion systems. Additionally, considering more complex braiding scenarios involving multiple anyons or different braiding paths would contribute to a richer understanding of the topological quantum computation landscape. Furthermore, the paper could delve into experimental implementations, collaborating with experimentalists to validate the proposed protocols on physical quantum hardware. Finally, exploring the scalability of the proposed methods for larger quantum systems and assessing their compatibility with emerging quantum technologies would be crucial for the advancement of Majorana fermion braiding in the context of fault-tolerant quantum computation.

# References

[1] Stefan Backens, Alexander Shnirman, Yuriy Makhlin, Yuval Gefen, Johan E. Mooij, and Gerd Schön. Emulating majorana fermions and their braiding by ising spin chains. *Physical Review B*, 96(19), November 2017.

[2] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, January 2003.

[3] Martin Leijnse and Karsten Flensberg. Introduction to topological superconductivity and majorana fermions. *Semiconductor Science and Technology*, 27(12):124003, November 2012.

[4] Ettore Majorana and Luciano Maiani. *A symmetric theory of electrons and positrons*, pages 201–233. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[5] J. Nakamura, S. Liang, G. C. Gardner, and M. J. Manfra. Direct observation of anyonic braiding statistics. *Nature Physics*, 16(9):931–936, September 2020.

[6] Chetan Nayak, Steven H. Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Reviews of Modern Physics*, 80(3):1083–1159, September 2008.

[7] Atsushi Sakaguchi, Shunya Konno, Fumiya Hanamura, Warit Asavanant, Kan Takase, Hisashi Ogawa, Petr Marek, Radim Filip, Jun-ichi Yoshikawa, Elanor Huntington, Hidehiro Yonezawa, and Akira Furusawa. Nonlinear feedforward enabling quantum computation. *Nature Communications*, 14(1):3817, 2023.

[8] Frank Wilczek. Majorana returns. *Nature Physics*, 5(9):614–618, 2009.

# A  Expanding Matrix Exponential

We start from the follwoing expression
$$e^{\beta \gamma_n \gamma_m}.$$

Note that this is a matrix exponential, so we need to expand it using the matrix exponential formula,
$$e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k$$

Then we have
$$e^{\beta \gamma_n \gamma_m} = \sum_{k=0}^{\infty} \frac{1}{k!} (\beta \gamma_n \gamma_m)^k = \sum_{k=0}^{\infty} \frac{\beta^k}{k!} (\gamma_n \gamma_m)^k$$
$$= 1 + \beta \gamma_n \gamma_m + \frac{\beta^2}{2!} (\gamma_n \gamma_m)^2 + \frac{\beta^3}{3!} (\gamma_n \gamma_m)^3 + \cdots,$$

where $\beta$ is a real number. But Majoranas anitcommute with each other,
$$(\gamma_n \gamma_m)^2 = (\gamma_n \gamma_m)(\gamma_n \gamma_m) = - \underbrace{\gamma_n \gamma_n}_{1} \underbrace{\gamma_m \gamma_m}_{1} = -1.$$

Thus the expansion becomes

$$e^{\beta \gamma_n \gamma_m} = 1 + \frac{\beta}{1!} (\gamma_n \gamma_m) - \frac{\beta^2}{2!} - \frac{\beta^3}{3!} (\gamma_n \gamma_m) + \frac{\beta^4}{4!} + \frac{\beta^5}{5!} (\gamma_n \gamma_m) + \cdots$$
$$= \left( 1 - \frac{\beta^2}{2!} + \frac{\beta^4}{4!} - \cdots \right) + (\gamma_n \gamma_m) \left( \beta - \frac{\beta^3}{3!} + \frac{\beta^5}{5!} - \cdots \right)$$
$$= \cos \beta + \gamma_n \gamma_m \sin \beta.$$

# B  Statevector and Density Matrix

Assume you have a state vector $|\psi\rangle$, which is a column vector with complex numbers:

$$|\psi\rangle = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{bmatrix}$$

Here, $n$ is the dimensionality of the quantum system. Take the conjugate transpose of the state vector, denoted by $\langle \psi|$. This is obtained by taking the complex conjugate of each element and then transposing the vector:
$$\langle \psi| = \begin{bmatrix} \psi_1^* & \psi_2^* & \cdots & \psi_n^* \end{bmatrix}^T$$

Compute the outer product to obtain the density matrix $\rho$:

$$\rho = |\psi\rangle\langle\psi| = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{bmatrix} \begin{bmatrix} \psi_1^* & \psi_2^* & \cdots & \psi_n^* \end{bmatrix}$$

The result is a square matrix of size $n \times n$, representing the density matrix. The density matrix $\rho$ is Hermitian (equal to its conjugate transpose) and positive semi-definite. If the state vector $|\psi\rangle$ is normalized (has a magnitude of 1), then the trace of the density matrix is equal to 1, which is a requirement for a valid density matrix representing a physical state.

**Pure State:**

For a pure state density matrix $\rho$, which satisfies $\rho^2 = \rho$ (idempotent), you can find the state vector $|\psi\rangle$ by:

$$|\psi\rangle = \sqrt{\lambda} \cdot |\phi\rangle$$

Here, $\lambda$ is the non-zero eigenvalue of $\rho$, and $|\phi\rangle$ is the corresponding eigenvector.

**Mixed State:** For a mixed state density matrix $\rho$ with multiple non-zero eigenvalues, it is not possible to uniquely determine a single state vector. The system is in a statistical mixture of pure states. However, you can find a set of state vectors and their corresponding probabilities:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

Here, $|\psi_i\rangle$ are the eigenvectors of $\rho$, and $p_i$ are the corresponding eigenvalues. The state vectors represent the pure states in the mixture, and the probabilities $p_i$ give the weight of each pure state in the mixture.

**Implementation:** To implement this in practice, you can use a numerical linear algebra library (such as NumPy for Python) to compute the eigenvalues and eigenvectors of the density matrix. Depending on the specific form of the density matrix, you might need to use different methods. Keep in mind that this process may not be unique, especially for mixed states, where different sets of pure states and probabilities can lead to the same density matrix.

# C  Code for Calculating Fidelity

```python
import numpy as np

def fid(matrix1, matrix2):
    vec1 = np.array(matrix1).flatten()
    vec2 = np.array(matrix2).flatten()

    if not (np.linalg.norm(vec1) and np.linalg.norm(vec2)):
        return 0.0

    vec1 /= np.linalg.norm(vec1)
    vec2 /= np.linalg.norm(vec2)

    return np.round(np.abs(np.vdot(vec1, vec2))**2, 15)
```

# D  Jordan-Wigner Transformation Code

```
# From https://mareknarozniak.com/2020/10/14/jordan−wigner−transformation/
# By dr marek narozniak
def Is(i, levels=2): return [qeye(levels) for j in range(0, i)]
def Sx(N, i): return tensor(Is(i) + [sigmax()] + Is(N − i − 1))
def Sy(N, i): return tensor(Is(i) + [sigmay()] + Is(N − i − 1))
def Sz(N, i): return tensor(Is(i) + [sigmaz()] + Is(N − i − 1))
def I(N): return Sz(N, 0)*Sz(N, 0)

def osum(lst):
    return np.sum(np.array(lst, dtype=object))

def oprd(lst, d=None):
    if len(lst) == 0:
        return d
    p = lst[0]
    for U in lst[1:]:
        p = p*U
    return p

# N is the exponent, L is the length of the chain
def opow(L, op, N): return oprd([op for i in range(N)])

def commutator(A, B):
    return A*B − B*A

def anticommutator(A, B):
    return A*B + B*A

def gamma_a(N, n, Opers=None):
    Sa, Sb, Sc = Sz, Sx, Sy
    if Opers is not None:
        Sa, Sb, Sc = Opers
    return oprd([Sb(N, j) for j in range(n)], d=I(N)) * Sa(N, n)

def gamma_b(N, n, Opers=None):
    Sa, Sb, Sc = Sz, Sx, Sy
    if Opers is not None:
        Sa, Sb, Sc = Opers
    return oprd([Sb(N, j) for j in range(n)], d=I(N)) * Sc(N, n)
```

And to iterate through all measurement projector from singel method call, we have

```
def calc_prj_meas(Ss, N):
    pairs = get_prj_meas(N)
    a, b, c = pairs.shape
    assert len(Ss) == a*b
    pairs = pairs.reshape((a * b, c))
```

```
    res = I(N)
    for i in range(a*b):
        il, ir = pairs[i]
        res *= (1 + Ss[i] * 1j   * get_gamma(N, il) * get_gamma(N, ir))

    return res
```

# E   Applying Exponential Form of Measurements

```
def calc_prj_meas_t(Ss, N, t):
    pairs = get_prj_meas(N)
    a, b, c = pairs.shape
    assert len(Ss) == a*b
    pairs = pairs.reshape((a * b, c))

    res = I(N)
    for i in range(a*b):
        il, ir = pairs[i]
        res *= (t * 1j * Ss[i] * get_gamma(N, il) * get_gamma(N, ir)).expm()
```

# F   Protocol to Correct and get Fidelity

```
def get_corrected_fid(circ_sv, mem):
    for i in range(10):
        s3, s2, s1, s0 = mem

        if s0 == s3:
            if s1 == s2:
                circ_sv = circ_sv.evolve(Pauli('ZZ'))
            else:
                pass
        else:
            if s1 == s2:
                pass
            else:
                circ_sv = circ_sv.evolve(Pauli('ZZ'))

        circ_sv = partial_trace(circ_sv, [1]).to_statevector()
        return np.abs(np.vdot(circ_sv, sv_br))**2
```