Topological Qubit Simulations in t-junction Kitaev Networks

Hrishikesh Belagali

May 18, 2025

1 Introduction

In this work, I aim to simulate a topological qubit through the quantum braiding of 3 Kitaev chains, using Python and NumPy. The goal of doing so is to gain some knowledge on the nuances of topological quantum computing, and a better understanding of non-Abelian statistics.

2 Formalism

2.1 The Physical Model

2.1.1 Kitaev Chains

A Kitaev chain is a 1-dimensional topological superconductor, which can host Majorana Zero Modes (hereafter abbreviated as MZMs). In terms of normal fermionic operators, the Kitaev chain Hamiltonian is given by

$$\hat{H} = -\mu \sum_{j=1}^{N} c_j^\dagger c_j - t \sum_{j=1}^{N-1} \left(c_j^\dagger c_{j+1} + c_{j+1}^\dagger c_j \right) + \Delta \sum_{j=1}^{N-1} \left(c_j c_{j+1} + c_{j+1}^\dagger c_j^\dagger \right)$$

where -

- μ: chemical potential
- t: hopping amplitude
- Δ: p-wave superconducting pairing
- $c_j, c_j^\dagger$ fermionic creation/annihilation operators

Majorana modes appear at the ends of the chain in the topological phase when $|\mu| < 2t$. We can express the Hamiltonian in the more convenient Majorana operator form. These operators are defined as

$$\gamma_{A,j} = c_j + c_j^\dagger$$

$$\gamma_{B,j} = -i(c_j - c_j^\dagger)$$

The convenient Hamiltonian becomes

$$\hat{H} = -\frac{\mu}{2} \sum_{j=1}^{N} (1 + i\gamma_{A,j}\gamma_{B,j}) + \frac{i}{2} \sum_{j=1}^{N-1} (t - \Delta)\gamma_{B,j}\gamma_{A,j+1} + \frac{i}{2} \sum_{j=1}^{N-1} (t + \Delta)\gamma_{A,j}\gamma_{B,j+1}$$

Milestone: Successfully simulate a Kitaev chain and identify MZMs (ideally through visual and mathematical representation)

2.1.2 t-Junction Kitaev Network

The t-junction Kitaev network is the minimal geometry required to host MZMs. They will be simulated as tight-binding lattices, and the full network will be modeled as a graph of Kitaev chains, where coupling terms connect the adjacent sites and time-dependent parameters are used to control MZM motion. This will be essential for the braiding process.

Milestone: Identify time-dependent MZM movement and visually represent the Kitaev network as a graph.

2.2 Simulation Technique - BdG Formalism

2.2.1 The Wavefunction

The Bogoliubov-de Gennes (BdG) formalism will be used to simulate the full quantum state evolution.

$$i\frac{\mathrm{d}}{\mathrm{d}t}\Psi(t) = \hat{H}(t)\Psi(t)$$

where $\Psi(t)$ is the BdG quasiparticle wavefunction. The braiding (which is inherently adiabatic) will be simulated by evolving the Hamiltonian parameters slowly in time.

2.2.2 Adiabatic Quantum Braiding

- Create a terminal MZM in each arm.
- Tune parameters to move one MZM through the junction into the other arm.
- Continue moving until the MZM has encircled another MZM, this is a braid.
- Ensure the evolution is adiabatic to stay in the ground-state.

Track the braiding by projecting the final state back onto the degenerate ground-state basis and compute the overlap to find the unitary braid transformation. Project the Hamiltonian into the Majorana basis, and find the zero-energy eigenstates. Plot their spatial localization to see where the MZMs are during the simulation.

2.3 Topological Protection

Simulate the following errors -

- Local Perturbations
- Pauli Errors

Measure fidelity of the error-states, entanglement entropy, parity conservation and other metrics. Ideally, you will show that only Majorana pair fusion can actually destroy information.