Topological excitations in magnets, superconductors, and magnet-superconductor heterostructures

Dissertation zur Erlangung des Grades "Doktor rerum naturalist" (Dr. rer. nat.)

der Fachbereiche
08 - Physik, Mathematik und Informatik,
09 - Chemie, Pharmazie und Geowissenschaften,
10 – Biologie und der
Universitätsmedizin

vorgelegt am 31. Juli 2023

Jonas Nothhelfer

MPGC

JG|U
JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

Gutacher*innen:
Prof. Dr. Karin Everschor-Sitte
Prof. Dr. Ferdinand Schmidt-Kaler

Skyrmion-Vortex pairs dance in the sky
A sight to behold, they catch the eye
Tiny particles, yet so much power
To revolutionize quantum computing, they empower

Universally capable, they can compute it all
Wrap your mind around it, it'll make you feel tall
The power to process beyond our wildest dreams
Skyrmion-Vortex pairs, they're a sight to be seen

Quantum information, encrypted and secure
Processing speeds, lightning-fast and pure
A new horizon of computation awaits
Thanks to Skyrmion-Vortex pairs, opening new gates

Tiny but mighty, they pave the way
For computing to evolve, to a brighter day
Skyrmion-Vortex pairs, the future is near
To a new era of quantum computing, they steer

So look up at the sky and marvel at their dance
For Skyrmion-Vortex pairs aren't left to chance
A glimpse into the future, they're ready to share
Welcome the age of universal quantum computing, if you dare.

- ChatGPT, 07.04.2023

List of publications

Peer-reviewed publications

1. **Asymmetric Skyrmion Hall effect in systems with a hybrid Dzyaloshinskii-Moriya interaction**
   K-W. Kim, K-W. Moon, N. Kerber, J. Nothhelfer, K. Everschor-Sitte
   Physical Review B **97**, 224427 (2018)
   The numerical computations in this publication were performed by myself, and the results are shown in Figure 4 of this publication. Additionally, I was involved in describing the numerical calculations presented in the paper. The results of this publication are not contained in the thesis.

2. **Nonlinear dynamics of topological ferromagnetic textures for frequency multiplication**
   D. R. Rodrigues, J. Nothhelfer, M. Mohensi, R. Knapman, P. Pirro, K. Everschor-Sitte
   Physical Review Applied **16**, 014020 (2021)
   Here, I contributed most of the numerical calculations, which resulted in Figures 3 and 4 of this publication. I also created Figure 5 from the provided data and contributed to Figure 1 of the paper. I played a significant role in describing the numerical methods. The results of this work are presented in Chapter 4 of this thesis.

3. **Steering of Majorana braiding via Skyrmion-Vortex Pairs: a scalable platform**
   J. Nothhelfer, S. A. Díaz, S. Kessler, T. Meng, M. Rizzi, K. M. D. Hals, K. Everschor-Sitte
   Physical Review B **105**, 224509 (2022)
   I am the principal author of this publication. I developed the original idea and shaped the project to a stage where it can be published. I planned the project from the beginning to completion and brought together the co-authors with their broad expertise in various topics needed for this work. Furthermore, I provided the backbone of this project by carrying out the numerical calculations, producing the figures, and writing large parts of the manuscript. The results of this publication are presented in Chapter 10 of this thesis.

Manuscripts in preperation

4. **Corralling Superconducting Vortices with Magnetic Domain Walls in Superconductor-Ferromagnet Heterostructures**
   S. A. Díaz, J. Nothhelfer, L. Goerzen, Kjetil Hals, K. Everschor-Sitte
   Manuscript in preparation
   I will be the shared first author of this work. I came up with the idea to manipulate superconducting vortices using magnetic domain walls in superconductor-ferromagnet heterostructures. I co-supervised the Bachelor thesis of L. Goerzen entitled “Composite Topological Structures in Superconductor-Ferromagnet Heterostructures” in which we established the proof of concept numerical results. For the manuscript that is in preparation, I carried out the numerical simulations. I

participated in discussions with S. A. Díaz and K. Hals, shaping the fundamental understanding of the interaction between magnetic domain walls and superconducting vortices. The results of this publication are presented in Chapter 9 of this thesis.

Patents

5. **METHOD AND DEVICE FOR PROVIDING ANYONS, USE OF THE DEVICE**, pending
J. Nothhelfer, K. M. D. Hals, M. Rizzi, K. Everschor-Sitte
WO002020249630A1, EP000003751472A1, US020220261678A1, JP002022536714A, CN000114008639A
I am the main inventor of this pending patent. I came up with the idea and prepared the patent application containing concepts for experimental implementation and designs of a device. I accompanied the procedure and gave the input to the patent lawyers that they needed to formulate the patent manuscript. The ideas and designs of the device are contained in Chapter 10 of this thesis.

Abstract

Topological excitations pervade many fields throughout physics. This thesis deals with topological excitations in magnets, superconductors, and their heterostructures. The results of my dissertation can be divided into four main points: (I) We demonstrated that topological magnetic structures, such as magnetic Vortices, droplets, and Skyrmions, can serve as efficient frequency multipliers in spintronic systems. By exciting skyrmion-bound modes at fractions of their eigenfrequencies, we extended the range of controllably accessible frequencies. (II) I extensively enhanced and extended a robust numerical framework to compute the eigensystem of a superconductor proximity-coupled to a magnetic thin film. This computational framework facilitates an accurate analysis of the adiabatic evolution of the self-consistently calculated superconducting order parameter. (III) By studying superconductor-magnet heterostructures, we investigated the intricate interactions between superconducting vortices and magnetic domain walls. The magnetoelectric effect induced by spin-orbit coupling revealed a rich interplay of interactions in these systems, that crucially depend on the domain wall helicity and the vorticity of the Vortex. (IV) I explored the potential of utilizing Skyrmion-Vortex pairs in superconductor-magnet heterostructures for topological quantum computing. Through adiabatic braiding of these hybrid topological excitations, I numerically confirmed the non-Abelian statistics of the attached Majorana zero modes. This braiding operation provides the basis for a scalable topological quantum computing platform.

Kurzzusammenfassung

Topologische Anregungen sind in zahlreichen Bereichen der Physik verbreitet. Diese Arbeit widmet sich den topologischen Anregungen in Magneten, Supraleitern und deren Heterostrukturen. Die Ergebnisse meiner Dissertation lassen sich in vier Hauptpunkte gliedern: (I) Wir konnten zeigen, dass topologische magnetische Strukturen wie magnetische Wirbel, Droplets und Skyrmionen in spintronischen Systemen effektiv als Frequenzvervielfacher eingesetzt werden können. Durch gezielte Anregung gebundener Moden in Skyrmionen bei Bruchteilen ihrer Eigenfrequenzen konnte ich den Bereich der kontrollierbaren Frequenzen erweitern. (II) Ein besonderer Schwerpunkt meiner Arbeit lag auf der Entwicklung und Verbesserung eines robusten numerischen Rahmens zur Berechnung des Eigensystems eines Supraleiters, der an einen magnetischen Dünnfilm gekoppelt ist. Mit dieser Methode war es möglich, die adiabatische Entwicklung des supraleitenden Ordnungsparameters präzise zu analysieren. (III) Durch unsere Untersuchungen von Supraleiter-Magnet-Heterostrukturen konnte wir wertvolle Erkenntnisse über die komplexen Wechselwirkungen zwischen supraleitenden Wirbeln und magnetischen Domänenwänden gewinnen. Besonders der magnetoelektrische Effekt, der durch die Spin-Bahn-Kopplung induziert wird, enthüllte ein reichhaltiges Zusammenspiel dieser Wechselwirkungen, das maßgeblich von der Helizität der Domänenwand und der Wirbelstärke abhängt. (IV) Ein weiterer bedeutsamer Beitrag meiner Forschung war die Untersuchung der Anwendbarkeit von Skyrmion-Vortex-Paaren in Supraleiter-Magnet-Heterostrukturen für topologische Quantencomputer. Durch adiabatische Verflechtung dieser hybriden topologischen Anregungen konnte ich die nicht-abelsche Statistik der gebundenen Majorana-Nullmoden numerisch bestätigen. Dies eröffnet vielversprechende Perspektiven für die Realisierung skalierbarer Plattformen für topologische Quantencomputer.

Contents

1 Introduction

Topology is a mathematical discipline that analyzes objects' characteristics that remain unchanged under continuous transformations. Despite its abstract nature, it has unexpected connections to our daily lives. Take, for instance, the simple act of navigating through a city. When we choose the most efficient path to reach a destination, we are essentially also engaging in a topological problem. By considering the connectivity and arrangement of streets, we optimize our routes, avoiding unnecessary detours, and crossings, thereby minimizing travel time. This application of topology to everyday navigation underscores its practical significance. Furthermore, the utilization of topological concepts in designing technical devices, such as circuit boards and computer networks, allows for efficient routing, fault tolerance, and robust communication, enabling seamless connectivity in our interconnected world. Topology permeates various domains of physics, including excitations in magnets, superconductors, and their heterostructures in this thesis. In particular, the intricate design and manipulation of superconductor-magnet interfaces offer a unique platform to harness the principles of topology, allowing for the emergence of topological excitations and their potential applications in fields such as quantum computation and information storage.

1.1. Topology

Topology as a mathematical discipline can be traced back to the 18th and 19th centuries. The pioneering works of mathematicians such as Leonhard Euler, Carl Friedrich Gauss, and Bernhard Riemann laid the foundation for the development of this field [1]. The Euler characteristic [2], which relates the number of vertices, edges, and faces of a polyhedron, exemplified the interplay between geometry and topology.

In the early 20th century, topology began to take shape as a distinct branch of mathematics, with Henri Poincaré making significant contributions. Poincaré introduced the concept of homotopy and homology, providing a rigorous framework for studying the properties of spaces and their transformations. He also investigated the fundamental group and the notion of a manifold, which are key elements in understanding the topological properties of objects [3].

The field of topology found surprising applications in condensed matter physics, particularly in the study of exotic phases of matter and their excitations. In recognition of the significant contributions made in this area, the Nobel Prize in physics was awarded to David J. Thouless, F. Duncan M. Haldane and J. Michael Kosterlitz [4]. Their theoretical discoveries of topological phase transitions and topological phases of matter have had a profound impact on our understanding of condensed matter systems. These groundbreaking achievements have paved the way for the exploration and development of topological materials with unique properties, such as topological insulators and superconductors. The field of topological superconductors has made significant progress by combining superconducting materials and topological insulators. Moreover, this has laid the foundation for research in superconductor-magnet heterostructures. Despite topological insulators and superconductors, topology also plays a crucial role in magnets with magnetic skyrmions being the most prominent example, which will be discussed in the subsequent section.

1.2. Magnets and Topological Excitations

Magnetic materials were discovered as early as ancient times, when people found naturally occurring magnetic minerals, such as lodestone. However, it was not until the 19th century that systematic investigations into the properties of magnetic materials began. The development of electromagnetic theory by James Clerk Maxwell [5] and the groundbreaking discoveries of Michael Faraday [6] laid the foundation for understanding the fundamental principles governing magnetism. Subsequent advancements in materials science and quantum mechanics led to the exploration of various magnetic phenomena, including ferromagnetism, antiferromagnetism, and spintronics.

The history of topological magnetic excitations dates back several decades and has witnessed significant breakthroughs in our understanding of fundamental physics and the potential applications of these novel phenomena. The journey began with the discovery of magnetic solitons [7], which laid the foundation for exploring topological aspects of magnetic systems.

In the early 1970s, the pioneering work of Tony Skyrme introduced the concept of solitons in field theories and gauge theories, respectively [8, 9]. These solitons, known as skyrmions, were initially proposed in the context of particle physics to describe topologically stable configurations.

Based on these concepts, also similar skyrmion field configurations were predicted in condensed matter systems. In magnets, however, it was just within the last decade that their existence was experimentally confirmed [10, 11, 12]. Since the initial discovery, skyrmions have been observed in diverse materials, ranging from metallic systems to insulating multilayers [13]. The realization of room-temperature skyrmions in thin films of certain magnetic materials has been a significant milestone in this field [13]. Further investigations revealed that magnetic skyrmions possess intriguing topological characteristics, making them robust against external perturbations and highly mobile [14]. These attributes opened up new avenues for utilizing skyrmions in various technological applications in spintronics [13, 15].

1.3. Topological Superconductors

Parallel to the progress in magnetic materials, the study of superconductivity emerged as a distinct field in the early 20th century. The discovery of superconductivity in mercury by Heike Kamerlingh Onnes in 1911 marked a significant milestone [16]. Superconductivity, characterized by the absence of electrical resistance below a critical temperature, has attracted the attention of scientists worldwide. Theoretical frameworks, such as the BCS theory [17] proposed by John Bardeen, Leon Cooper, and Robert Schrieffer, provided a fundamental understanding of conventional superconductivity.

There are two types of superconductors: Type I and Type II. Type I superconductors exhibit the so-called Meissner effect, i.e. when subjected to a magnetic field they lose their superconductivity abruptly when the magnetic field surpasses a critical value. In contrast, Type II superconductors display a more complex response to magnetic fields, allowing the penetration of magnetic flux in specific regions, where superconducting vortices form. In the early 21st century, there was a growing focus on exploring the topological aspects of superconductors. Alexei Kitaev's seminal work in 2001 [18] revolutionized the field by introducing the concept of topological superconductors. Kitaev proposed a toy model [19] of a one-dimensional p-wave superconductor that hosted exotic quasiparticles known as Majorana modes. These Majorana zero-energy modes, acting as their own antiparticles, exhibit non-Abelian statistics and hold promise for fault-tolerant quantum computation [20].

In 2008, Fu and Kane proposed a platform for the realization of topological superconductivity by combining topological insulators with superconducting materials [21]. This combination of s-wave superconductivity with a topological insulator possessing strong spin-orbit coupling and inducing proximity effects at their interfaces recovers an effective p-wave superconductor [22]. Subsequent experimental efforts focused on the search for topological superconductivity in various systems, including hybrid structures and exotic materials [23, 24, 25].

1.4. Superconductor-Magnet Heterostructures

The study of type II superconductors laid the foundation for subsequent research on superconductor-magnet systems [26]. Although strong magnetic fields destroy superconductivity, proximity effects have been harnessed to create hybrid structures composed of superconductors and ferromagnetic or magnetic materials [27]. This breakthrough allowed for precise control over the proximity-induced superconducting properties at the interface between the two materials. The manipulation of the superconductor-magnet interface enabled the exploration of intriguing phenomena, such as the possibility of creating topological superconducting states.

In recent years, the field of superconductor-magnet heterostructures has witnessed a surge of interest due to its potential applications in quantum information processing and topological quantum computation. The search for Majorana zero modes, exotic quasiparticles that possess non-Abelian statistics, has become a focal point of research in this area. Superconductor-magnet heterostructures provide a promising platform for the creation and manipulation of Majorana zero modes, which are essential building blocks for fault-tolerant quantum computation.

1.5. Content of this Thesis

This thesis contributes to the rich tapestry of research on topological excitations in chiral magnets (Part I), superconductors (Part II), and their hybrid heterostructures (Part III).

In part I, we present our findings on the utilization of magnetic Skyrmion eigenmodes for frequency multiplication published in [28]. By employing micromagnetic simulations, we demonstrate that the breathing mode and elliptical mode of Skyrmions can be excited at fractions of their eigenfrequencies. Notably, we observe that exciting these eigenmodes under low damping conditions lead to amplified amplitudes. Furthermore, the tunability of these frequencies through temperature changes and magnetic or electric fields showcases the potential for industrial applications. Importantly, our developed theory establishes a framework that transcends the specific microscopic characteristics of the magnetic structure, thus enabling broad applicability.

In part II, we discuss quantum computation with topological superconductors. We elucidate the principles of braiding in a Kitaev chain. Additionally, we introduce a novel method for eliminating numerical phases in zero energy eigenvalues, using the Kitaev chain as an example. This method proves effective in both analytical and numerical models, with potential applicability to more intricate systems. Moreover, we detail the implementation of braiding in a T-junction discussed in the literature, achieved by controlling the superconductor's topological and trivial phases through the chemical potential. Remarkably, our numerical simulations confirm the expected braiding statistics, as demonstrated in Ivanov's seminal work [29].

In part III, we explore the realm of superconductor-magnet heterostructures and their potential for quantum computation. We establish a crucial link between these heterostructures, the Kitaev chain, and the associated physics. We propose a model Hamiltonian that represents the key physics of the superconductor-magnet heterostructure. Subsequently, we develop a powerful numerical framework capable of computing the system's eigensystem and solving the self-consistency condition of the BCS theory. Using the numerical tools we study the interaction between magnetic domain walls and superconducting vortices, enabling the manipulation and ordering of vortices. To explain our findings of the interactions between magnetic domain walls and superconducting vortices, we developed an effective analytic theory based on the emergent superconducting currents. In a separate study, we predict the localization of Majorana modes at Skyrmion-Vortex pairs, highlighting their potential for braiding enabling topological quantum computing. The research findings have been incorporated into a pending patent [30] and have been published in Ref. [31]. Leveraging these findings, we predict the creation, motion, and fusion of Majoranas using Skyrmion-Vortex pairs on a scalable platform. Notably, we demonstrate how Skyrmion-Vortex pairs can serve as qubit states and perform Clifford gates, thus establishing their utility for quantum computation.

In summary, this thesis dives into the realms of magnetic materials, topological superconductors, and superconductor-magnet heterostructures, connecting their historical development with the contemporary focus on topological excitations.

Part I.

Chiral Magnets

2 Micromagnetic Approach to Magnetism

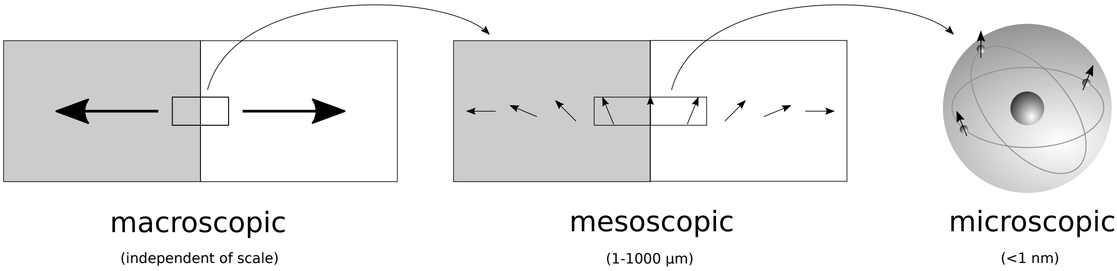

Figure 2.1.: Overview of the different length scales.

Magnetically ordered materials can be studied on different length scales. A pictorial description of the different length scales can be seen in Figure 2.1. We can explain the origin and the magnitude of magnetic moments, crystal anisotropy and magnetoelastic interaction on the atomic scale by dealing with the arrangement of spins on the crystal lattice sites. Furthermore, we can also describe magnetism on a macroscopic length scale, where the average magnetization is described. In between those two extreme cases, there is the mesoscopic approach. In this instance, we do not consider the individual magnetic moments, but their sums in a certain neighbourhood. In this way, one considers the local average of magnetic moments over a small volume under the Curie temperature, described by the magnetisation vector $\boldsymbol{M}$. The magnitude of the magnetization vector is called saturation magnetization M_s and it is in general a function of the temperature. By dividing the magnetization by its magnitude we get a unit vector of magnetization $\boldsymbol{m}(\boldsymbol{r}) = \boldsymbol{M}(\boldsymbol{r})/M_s$ with $\boldsymbol{m}^2 = 1$. In the following, we will consider the continuum theory of micromagnetics within this mesoscopic length scale. Therefore, we will study this vector field $\boldsymbol{m}(\boldsymbol{r})$ and its response to different energy terms. In this chapter, we summarize the theory presented in [32, 33].

2.1. Magnetostatics

We begin our discussion by introducing magnetostatic theory, which involves finding the equilibrium configuration of magnetic materials by minimizing their total energy. In the context of a chiral ferromagnet, where inversion symmetry is broken at the surface, the energy of the system can be expressed in a specific manner [34].

$$E[\boldsymbol{m}] = \int dV \mathcal{E}_{\mathrm{exch}}[\boldsymbol{m}] + \mathcal{E}_{\mathrm{ani}}[\boldsymbol{m}] + \mathcal{E}_{\mathrm{ext}}[\boldsymbol{m}] + \mathcal{E}_{\mathrm{DM}}[\boldsymbol{m}] + \mathcal{E}_{\mathrm{demag}}[\boldsymbol{m}], \tag{2.1}$$

where the different terms respectively represent the exchange interaction, the anisotropy energy, the external magnetic field, the Dzyaloshinskii-Moriya interactions and the magnetostatic energy. Further properties of the material can be added to this energy functional (2.1). Two examples of additional terms which are often used are magnetoelastic energy, which describes the mechanical stress on a material, and magnetostrictive self-energy, which describes the spontaneous deformations of the material due to certain domain configurations. In our case, these further properties are neglected. In the following, we will describe the different micromagnetic energy terms in more detail.

Exchange interaction: The exchange interaction is based on the Pauli principle. When two Fermions are exchanged their wave function changes its sign. In the case of ferromagnetic materials, this interaction leads to a constant magnetization direction of the spins and deviations from this magnetization direction lead to an energy penalty. This energy penalty can be expressed in the continuum limit as

$$\mathcal{E}_{\text{exch}} = A\,(\nabla \boldsymbol{m})^2 \tag{2.2}$$

where A is the exchange stiffness and $(\nabla \boldsymbol{m})^2$ is defined by $\sum_{\alpha,\beta}(\partial_\alpha m_\beta)(\partial_\alpha m_\beta)$, where summation includes all spatial indices α and β.

External magnetic field: If a magnetic field $\boldsymbol{H}_{\text{ext}}$ is applied to the magnet it will interact with the magnetization. This effect can be described by the Zeeman term

$$\mathcal{E}_{\text{ext}} = -\mu_0\, M_s\, \boldsymbol{H}_{\text{ext}} \cdot \boldsymbol{m} = -\mu_0\, M_s\, H_{\text{ext}}\, \cos(\varphi) \tag{2.3}$$

in which φ is the angle between magnetization and field. Equation (2.3) is minimized when the magnetization is aligned to the magnetic field ($\varphi = 0$).

Anisotropy: The magnetisation of a crystal tends to align along the structural axis of the material. This effect originates from spin-orbit effects and is called anisotropy energy. Since we consider a thin film, we will discuss the surface or interface anisotropy. This type of anisotropy can be expressed in first-order as

$$\mathcal{E}_{\text{ani}} = -K\,(m_z^2 - 1), \tag{2.4}$$

where we have chosen the direction of the vector normal to the surface of the thin film along the $\hat{z}$-axis.

Magnetostatic interactions: The magnetisation in the magnetic material produces a stray field, which has to be minimized in order to minimize the energy. In general, this term is very hard to calculate since it is a long-range effect. In the case of a thin film, one can approximate this term by considering an infinitely extended plate. In this one-dimensional case, the magnetostatic term has the form of anisotropy energy [33]. We can then add it to our integral by modifying the anisotropy constant from K to $K + K_d = K - \frac{\mu_0}{2} M_s^2$ [35].

Dzyaloshinskii-Moriya interaction (DMI): The DMI was first proposed phenomenologically by Dzyaloshinskii [36], and later derived from spin-orbit coupling by Moriya [37]. It is present in materials with no inversion symmetry or when inversion symmetry is broken at interfaces. The most general form of the DMI can be written as

$$\mathcal{E}_{\mathrm{DM}} = -D^{\alpha}_{ij} m_i \partial_\alpha m_j. \tag{2.5}$$

The DMI is an asymmetric exchange interaction which tries to orient the closest neighbours perpendicular to each other. In a crystal, this leads to a finite angle between the magnetic moments, so that they are neither parallel nor antiparallel to each other. In the bulk of chiral magnets, the DMI tensor simplifies to $D^{\alpha}_{ij} = D\epsilon_{i\alpha j}$, which is called Bloch DMI. In our case, we consider thin films where the inversion symmetry along the normal direction of the film is broken. Therefore, the DMI tensor is given by $D^{\alpha}_{ij} = D(\delta_{i\alpha}\delta_{jz} - \delta_{iz}\delta_{j\alpha})$, where the films normal vector is pointing in the $\hat{z}$-direction. If we write that out we get

$$\mathcal{E}_{\mathrm{DM}} = -D\,\boldsymbol{m} \cdot ((\hat{\boldsymbol{z}} \times \nabla) \times \boldsymbol{m})\,. \tag{2.6}$$

Rewriting using the latter results, we get the following model

$$E[\boldsymbol{m}] = \int dV\,[A\,(\nabla \boldsymbol{m})^2 - D\,\boldsymbol{m} \cdot ((\hat{\boldsymbol{z}} \times \nabla) \times \boldsymbol{m}) - K\,(m_z^2 - 1) - \mu_0\, M_s\, \boldsymbol{H}_{\mathrm{ext}}\, \cdot \boldsymbol{m}]. \tag{2.7}$$

Until this point, we discussed all these different terms separately. In the full model, every energy contribution wants to align the magnetization in different directions. The configuration with minimal energy ultimately depends on the relative strengths of the coupling terms. This competition of the terms allows different structures to be realized. We will discuss some of the most interesting structures throughout the thesis, for example, Skyrmions, Antiskyrmions and domain walls.

2.2. Magnetisation Dynamics

The dynamics of a physical system can be described by studying the time derivative of the internal structure. In our case, we will look at the time derivative of the magnetization. Assuming a constant local magnitude of the magnetization vector as well as slow and smooth magnetization dynamics the equation for the time derivative of magnetization is the Landau-Lifshitz-Gilbert (LLG) equation [15, 28]

$$\frac{d\boldsymbol{m}}{dt} = -\gamma\, \boldsymbol{m} \times \boldsymbol{B}_{\mathrm{eff}} + \alpha\, \boldsymbol{m} \times \dot{\boldsymbol{m}} + \boldsymbol{\tau}, \tag{2.8}$$

in which γ is the gyromagnetic ratio, M_s is the saturation magnetisation, α is the Gilbert damping and τ represents additional torque terms. The core of this equation is the first term in which an effective field is acting on the magnetisation. This effective field is given by the functional derivative of the energy functional (2.1) concerning the magnetization vector and is given by

$$\boldsymbol{B}_{\text{eff}} = -\frac{1}{M_s}\frac{\delta E[\boldsymbol{m}]}{\delta \boldsymbol{m}}. \tag{2.9}$$

One can imagine that the action of this magnetic field is like a torque acting on the magnetisation leading to a precession of the magnetisation around the magnetic field. If we only consider the first term, the precession would continue forever and never reach an equilibrium. However, in the real world, there is also damping which will lead the magnetisation towards the direction of the magnetic field (depicted in Figure 2.2). Such a damping term was first introduced by Landau and Lifshitz [38] and was later replaced by the more physically intuitive term introduced by Gilbert leading to the Landau-Lifshitz-Gilbert (LLG) equation [39].

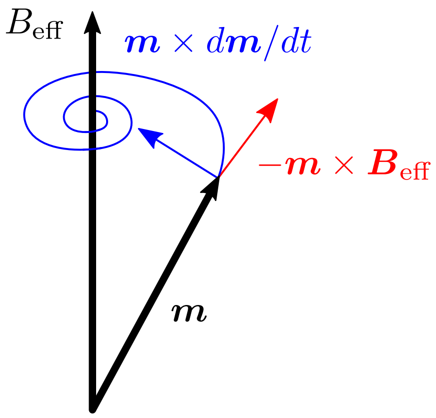
B_{eff}
$\boldsymbol{m} \times d\boldsymbol{m}/dt$
$-\boldsymbol{m} \times \boldsymbol{B}_{\text{eff}}$
$\boldsymbol{m}$

Figure 2.2.: Schematic representation of a magnetic vector aligning with the effective field due to the precession term and the damping term.

In general, this complex partial differential equation cannot be solved analytically. But it can be solved numerically and we will discuss the techniques for the numerical solution in the next section.

2.2.1. Numerical Implementation of the Landau-Lifshitz-Gilbert Equation

There are multiple different numeric implementations of the LLG equation such as micromagnum [40], OOMF [41] and **mumax**^{3} [42]. In the following, we will use **mumax**^{3} which uses two different interlocking methods to solve the differential equation. At first, it calculates the magnetic ground state by calculating the interplay between the different forces on the magnetisation presented in (2.7). In the second step, it calculates the dynamics with the time integration method. The details on how one can define different geometries and how the different energy terms are implemented can be read in their paper [42].

Simulating magnetic ground states with mumax^3: The numerical relaxation is implemented by the function `relax()`, which attempts to find the system's energy minimum. We discussed before that the alignment of the magnetisation to the effective magnetic field is represented by the damping term. The energy is minimized due to the alignment with respect to this effective field. To reach this energy minimum faster, the precession term of the LLG equation is neglected here. The first one relaxes until the total energy cuts the noise floor, which means that we are already close to equilibrium. To approach the equilibrium even closer one switches now to monitor the magnitude of the torque, because close to the equilibrium the torque will decrease monotonically and it is less noisy. We continue to relax like this and every time the torque cuts the noise floor one sets the maximal error lower until a maximal error is much lower than the typical scales of the system. At this point, the system can be seen as relaxed.

Simulating Magnetization Dynamics with mumax^3: **Mumax^3** implements different variants of explicit Runge-Kutta methods for advancing the Landau Lifshitz equation. It is a well-established numerical method to integrate ordinary differential equations by using a trial step at the midpoint of the interval to cancel out lower-order error terms. Furthermore, **mumax^3** uses adaptive time step control which automatically chooses the time step to keep the error step close to a preset error threshold.

3

Dynamic Excitations of Magnetic Skyrmions

3.1. Magnetic Skyrmions

Using the model in equation (2.7), we have the opportunity to explore various magnetic configurations. Our specific focus lies on investigating a small region within the phase diagram where Skyrmions manifest. Skyrmions represent topologically protected vortices in the magnetization field, and their topological nature becomes evident when the magnetization vector $\boldsymbol{m}$ is wrapped around the unit sphere. This wrapping allows Skyrmions to be characterized by the number of times the magnetization winds around the sphere, giving rise to the concept of the winding number, which can be expressed as follows: [15, 13, 43, 44]

$$\mathcal{W} = \frac{1}{4\pi} \int \mathrm{d}x\mathrm{d}y \, \boldsymbol{m} \cdot (\partial_x \boldsymbol{m} \times \partial_y \boldsymbol{m}) \,. \tag{3.1}$$

For a rotational symmetric Skyrmion, the magnetization profile can be written as [44]

$$\boldsymbol{m}(\boldsymbol{r}) = (\cos\Phi(\varphi)\sin\Theta(r), \sin\Phi(\varphi)\cos\Theta(r), \cos\Theta(r)) \,, \tag{3.2}$$

where we introduce the polar coordinates $\boldsymbol{r} = (r\cos(\varphi), r\sin(\varphi))$. If we plug this profile into the winding number we get

$$\mathcal{W} = \frac{1}{4\pi} \int_0^\infty \mathrm{d}r \int_0^{2\pi} \mathrm{d}\varphi \frac{\mathrm{d}\Theta(r)}{\mathrm{d}r} \frac{\mathrm{d}\Phi(\varphi)}{\mathrm{d}\varphi} \sin\Theta(r) = [\cos\Theta(r)]_{r=0}^{r=\infty} \, [\Phi(\varphi)]_{\varphi=0}^{\varphi=2\pi} \,. \tag{3.3}$$

Suppose the magnetisation points up at $r \to \infty$ and down at $r = 0$, then $[\cos\Theta(r)]_{r=0}^{r=\infty} = 2$. We can now write $\Phi(\varphi) = m\varphi + \gamma$, where m is the vorticity and γ is the helicity of the Skyrmion. In Figure 3.1, we show a configuration with $m = 1, \gamma = 0$, which is called a Néel Skyrmion. Note that topology is a continuous theory and we classify simulations and experiments, which are discrete systems. In discrete systems, non-conserving transformations are allowed but have a finite non-zero energy penalty.

The formation of a spin configuration like the Skyrmion arises from competing interactions, where certain interactions promote the alignment of magnetic moments, while others prefer their twisting. In experiments, the first Skyrmions have been observed in bulk crystals with broken inversion symmetry, where they form a Skyrmion lattice [10]. In this experiment Skyrmions are formed by a competition between a uniform stiffness A, DMI strength D, an applied magnetic field B and thermal fluctuations at temperatures slightly below the critical temperatures. There are different ways to achieve such a configuration and the most prominent mechanisms are the following [44]: (1) In a system with long-ranged magnetic dipolar interactions, favouring in-plane magnetisation, and

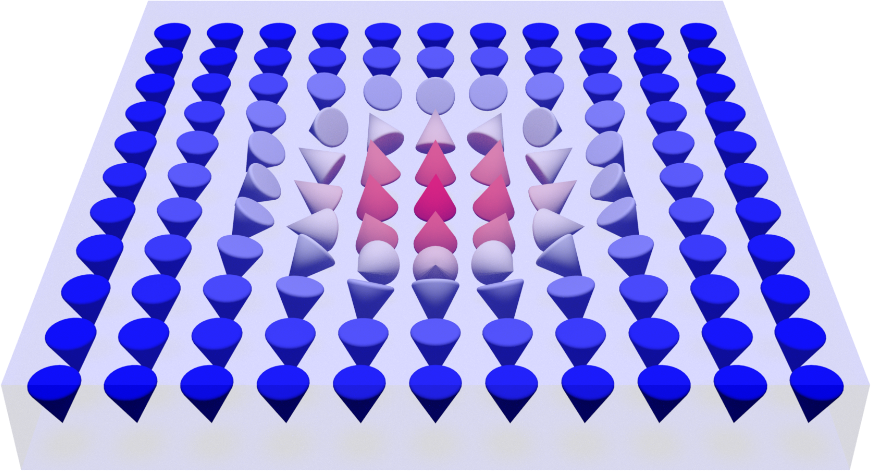

Figure 3.1.: Depiction of a Néel Skyrmion.

perpendicular easy-axis anisotropy, favouring out-of-plane magnetization, we encounter a competition between these interactions. This competition leads to the rotation of magnetization in a plane perpendicular to the film, resulting in periodic stripes that can further break up into magnetic bubbles or Skyrmions when subjected to a magnetic field. (2) The presence of the Dzyaloshinskii-Moriya Interaction (DMI) induces a transformation from the helical spiral structure into a triangular Skyrmion crystal when an external magnetic field is applied. (3) In the presence of frustrated exchange interactions, or (4) four-spin interactions, the system can form atomic-sized Skyrmions.

Skyrmions were not only found as lattices in bulk materials but also as single objects [45, 46], as clusters [47] in a helimagnet background [48, 49] or inplane Skyrmions [50, 51]. They occur in various systems and not only in bulk materials as in the first experiments. We are mostly interested in Skyrmions present in thin films [52]. Skyrmions can also be present at room temperature in systems with a strong interfacial DMI [53, 54]. Apart from Skyrmions being present in many different materials and exhibiting various shapes, they also possess diverse properties. For example, they can be created through magnetic and electric fields, thermal excitations, and spin torques, and can be moved using magnetic and electric fields, spin torques, magnons, temperature gradients, and thermal fluctuations [13]. The multitude of techniques available for manipulating Skyrmions makes them particularly intriguing for various applications. However, when a Skyrmion is moved, it will move under an angle to the direction of the applied current. Given this effect of Skyrmion Hall motion, many studies focus on its removal, especially in applications like the Skyrmion racetrack memory [55].

Additionally, it is also interesting to study the excitations of Skyrmion lattices as well as single Skyrmions. In the following, we will review the excitations of single Skyrmions in chiral magnets and we will discuss the importance of magnons for these excitations and how magnons interact with Skyrmions.

3.2. Eigenmodes of Skyrmions

Magnons are the quanta of spin wave excitations and can move Skyrmions as well as excite their eigenmodes. Thereby spin waves carry momentum but no charges, which will be the main reason they can excite the eigenmodes. Mochizuki [56] introduced the terms clockwise (CW), counterclockwise (CCW), and breathing modes to describe Skyrmion lattices. These categories can also be applied to classify the eigenmodes of individual Skyrmions. In Figure 3.2, we present magnon-Skyrmion bound states, which correspond to modes that fall into these distinct classes. Here, the subscript of the frequency ω labels the angular momentum quantum number n, where the sign of n determines the rotation direction. The frequencies of the bound states can be calculated by solving the Equation 2.8 without the damping term with the energy in equation (2.7) at zero magnetic fields [57]. In this model, we exclude the damping term and the magnetic field as the eigenfrequencies are determined analytically. The discussion on the damping term and magnetic field will be reserved for the study of single Skyrmions using a numerical framework.

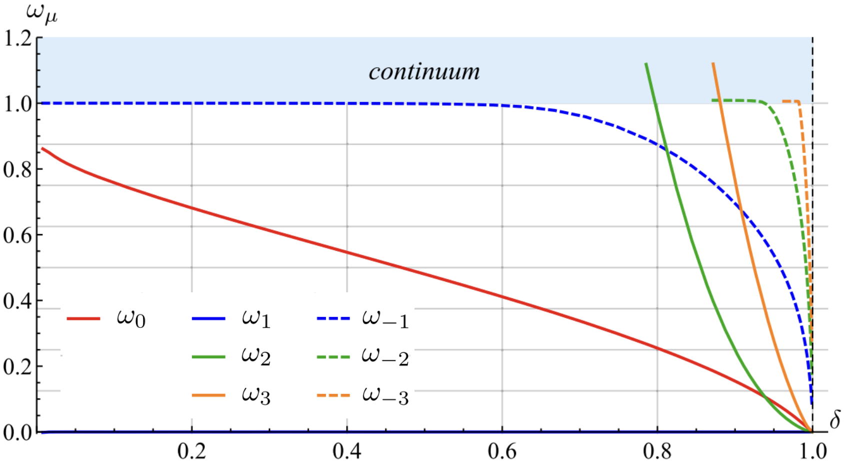

Figure 3.2.: The eigenfrequencies ω_μ of localized modes as functions of the normalized DMI constant $\delta = d/d_0$. Reprinted figure with permission from [35]. Copyright 2018 by the American Physical Society.

The eigenfrequencies of the analytic model can be excited by different mechanisms, which we will then use in the simulations to excite eigenmodes. The mechanism of the **breathing mode** is the following: When a magnetic field is applied, it aligns the spins in the core of the Skyrmion which results in the core growing. A magnetic field oriented antiparallel to the spins in the core results in the core shrinking. If the magnetic field now oscillates in such a way that it matches the growing and shrinking of the Skyrmion, it excites the corresponding mode of the Skyrmion. Since this motion looks like breathing, this mode is called breathing mode. Such modes are radially symmetric and can be excited best with an out-of-plane field, but it is also possible to excite them with an in-plane field. The behaviour of the **CW and CCW eigenmodes**, corresponding to the elliptical, triangular, and additional modes, is different. The shape of the Skyrmion during a CW or CCW mode breaks rotational symmetry (see Figure 3.3) so the excitation has to be one that also breaks rotational symmetry. That's why these modes do not couple to out-of-plane fields but only to in-plane fields. To explain the behaviour of these rotational modes intuitively one has to explain how magnons and Skyrmions interact with each other [58]. We begin with a magnetic system containing a Skyrmion and apply an oscillating magnetic field, leading to the propagation of a magnon within the material. The presence of a Skyrmion causes the magnon to scatter and transfer momentum to the Skyrmion. When the magnons are not in resonance with any Skyrmion eigenmode, they can induce a motion of the Skyrmion with a large Skyrmion Hall effect [59]. On the other hand, if the magnons are in resonance with the Skyrmion eigenmodes, they form magnon-skyrmion bound states [58]. These states originate from an initial bump that a magnon induces on the Skyrmion due to momentum transfer [60]. This bump then propagates along the Skyrmion. As the magnons are resonant with the bumps travelling around the Skyrmions, the shape becomes clearer the longer the excitation lasts. The Skyrmion will even remain in this excited state for a while after the oscillating magnetic field is turned off.

To study the shape of these Skyrmion modes in more detail the Skyrmion can be described by a domain wall that is connected on both ends and which can be parametrized as [60]

$$\boldsymbol{X}(s,t) = r(s,t)\,(\cos(s), \sin(s), 0) \tag{3.4}$$

where $s = [0, 2\pi)$ and $r(s)$ is a smooth function with $r(0,t) = r(2\pi,t)$. Under certain conditions [60], we find that

$$r(s) \approx r_{\text{min}} + \mathit{\Delta}_r \sin^2\left(\frac{ns}{2}\right) \tag{3.5}$$

where $\mathit{\Delta}_r = r_{\text{max}} - r_{\text{min}}$ describes the difference between the minimal radius r_{min} and the maximal radius r_{max}. Here n is an integer number representing the different modes of the skyrmion. These equations define Skyrmion profiles with well-established bumps that travel either clockwise or counterclockwise around the Skyrmion boundary. A representation of these shapes can be found in Figure 3.3. The existence of two travelling modes is connected to the particle-hole symmetry of the system [59] and it is represented by Skyrmions with different polarities.

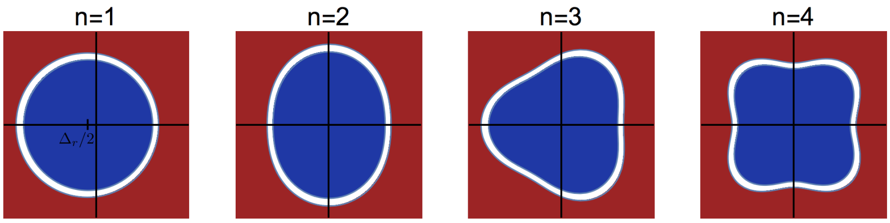

Figure 3.3.: Sketch of solutions described by Equation (3.5) with $\mathit{\Delta}_r/r_{\text{min}} = 0.3$ and $n = 1, 2, 3, 4$. These solutions correspond to the excitations of circular skyrmions. Their evolution in time is a rigid rotation with a frequency depending on their size. Notice that the mode $n = 1$ corresponds to what is known as (counter)clockwise modes for skyrmions. Reprinted figure with permission from [60]. Copyright 2018 by the American Physical Society.

4

Frequency multiplication of Skyrmion Modes

Every computing system depends critically on the production of periodic signals, or clocks. In the case of modern microprocessor integrated circuits and high-performance broadband communications, clocks must be very steady over lengthy periods of time. Standard oscillators, which can reach a broad frequency range, are not able to deliver such stable oscillations. Here frequency multipliers are able to generate really stable oscillations. They have the distinctive quality that, if the input frequency fluctuates within a specified finite range, the frequency multipliers' frequency multiplication ratio is not significantly affected [61, 62, 63, 64, 65, 66]. One downside of frequency multipliers is that they can only provide frequency multiplication if the system provides nonlinearity. Because of this, we want to employ topological objects which can provide this non-linearity through their excitation modes. Currently, Skyrmion systems are used in parameter regimes where a linear approach to their dynamics is appropriate for the majority of applications. The Skyrmion excitation modes, however, are fundamentally non-linear [28]. This implies, for example, an amplitude dependence of the excitation mode frequencies, as well as the existence of harmonic generation, i.e. the excitation of integer multiples of applied frequencies [28].

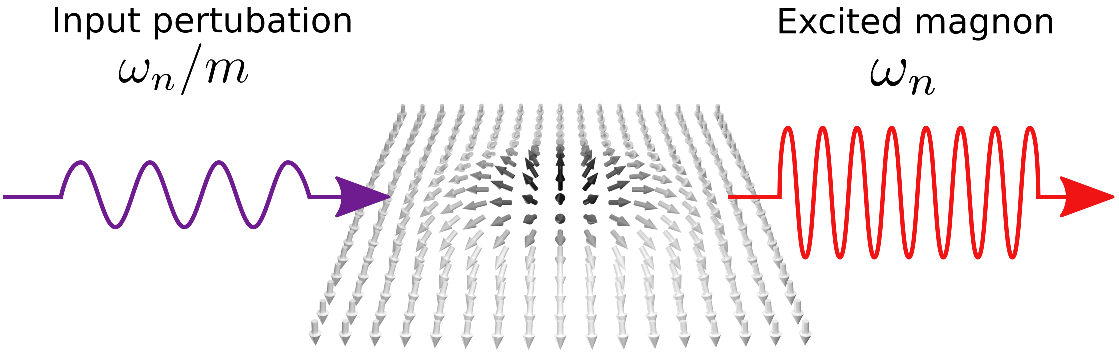

Figure 4.1.: Sketch of an example of frequency multiplication using a magnetic Skyrmion. Perturbations with a fraction of the eigenfrequency ω_n can excite the corresponding eigenmode of the Skyrmion. Moreover, when the input perturbation is created by an AC magnetic field, the amplitude of the exited eigenmode can be larger than the amplitude of the input magnon. The picture published in [28]

This part of the thesis is organized into different sections. In Section 4.1 we study the spectrum of eigenfrequencies of the skyrmion to give an overview. Next, we study in

Parameters	Symbol	Value
Sample dimensions	x, y, z	600nm, 600nm, 1nm
Saturation magnetization	M_s	$1.1 \cdot 10^6 \frac{A}{m}$
Exchange energy	A	$1.6 \cdot 10^{-11} \frac{J}{m}$
Anisotropy	K	$0.51 \cdot 10^6 \frac{J}{m^3}$
Damping	α	$1.0 \cdot 10^{-3}$

Table 4.1.: Sample Parameters of the magnetic material in which a Skyrmion is stabilized

Section 4.2 how applied perturbations with a fraction of the Skyrmion excitation eigenfrequency leads to resonance at the corresponding eigenfrequency as shown in Figure 4.1. For that, we analyze the breathing $n = 0$ mode in Section 4.2.1 of an isolated Skyrmion by applying AC magnetic fields, both in-plane and out-of-plane, with half and a third of the corresponding eigenfrequency. Additionally, we examine the amplitude dependence of the excited eigenmode in terms of the amplitude of the applied AC field and the damping parameter. We find the following interesting responses to applied fields

1. above a certain amplitude of the applied field, the Skyrmion mode is more efficiently excited with half of its eigenfrequency.
2. the perturbations with fractional frequencies are not eigenmodes of the system and decay after letting the system evolve for some time.
3. the eigenfrequencies are highly tunable since the material parameters can be tuned by temperature changes or applied electromagnetic fields. This means that one can design a device which can provide frequency multiplication for a continuous range of different frequencies.

After that, we show in Section 4.2.2 that also the elliptical mode can be excited with fractions of its eigenfrequency and it can be more efficient than exciting with its eigenfrequency. Furthermore, we embed in Section 4.3 our findings in a more general theory that not only proves that every mode in the Skyrmion can be excited with fractions of its eigenfrequency but also that this is true for any nonlinear system. In Section 4.4 we summarize the results and put them into a bigger context.

4.1. Determining Eigenfrequencies of the Skyrmion

In this section, we conduct micromagnetic simulations using **mumax**^{3} [42] to demonstrate the excitation modes of a Néel Skyrmion can be excited with fractions of their eigenfrequency. The Néel Skyrmion is stabilized by perpendicular anisotropy in a sample with the parameters detailed in Table 4.1. The model for these Skyrmions is given by Equation (2.7), which incorporates an external magnetic field used to excite the Skyrmion. We apply an oscillating in-plane magnetic field with an amplitude of 0.05 Tesla. The simulations consider $D < D_c = 4\sqrt{AK}/\pi$, ensuring that the ground state remains ferromagnetic [67, 35, 68].

To obtain an overview of all eigenmodes of the Skyrmion in our sample, we numerically reproduce Figure 3.2 from Kravchuk et al.'s paper [35]. The results are presented in Figure 4.3, where the frequency ω_n of the magnon-Skyrmion bound states is plotted

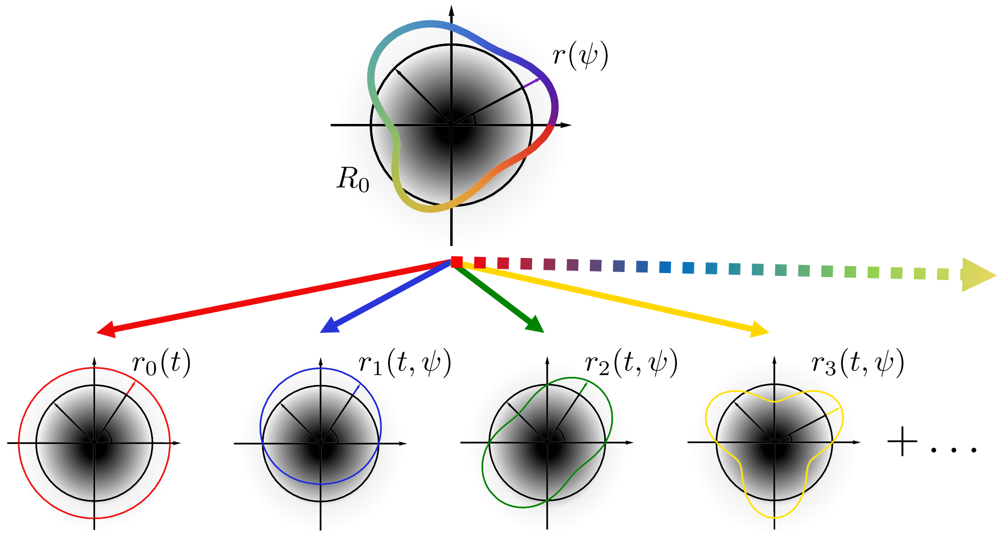

Figure 4.2.: Sketch of the internal Skyrmion modes, where black means the magnetization pointing down and white magnetization pointing up (opposite direction on the inside and outside). R_0 is the ground state radius of the Skyrmion, and, r_n characterizes the bound modes. The mode $n = 0$ represents the breathing mode. Modes with $|n| > 0$ rotate (counter-) clockwise with an amplitude-dependent frequency. Picture published in [28]

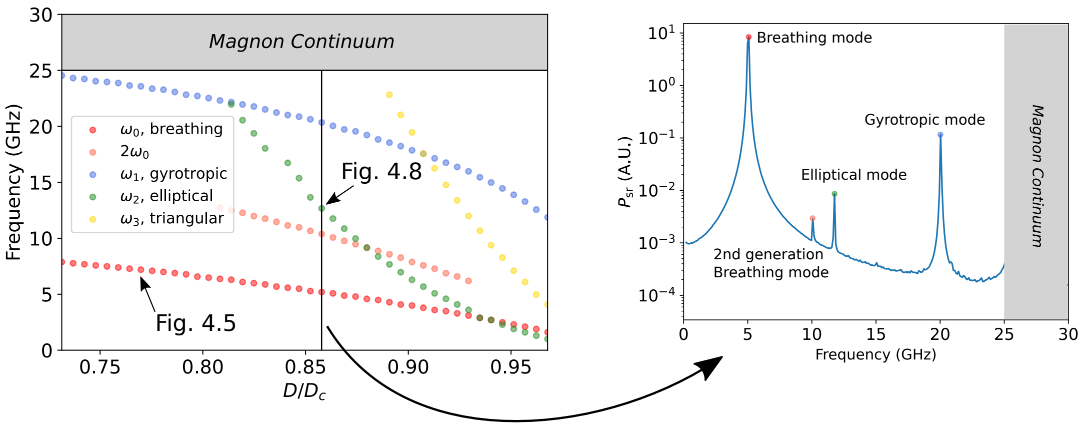

Figure 4.3.: Eigenfrequencies of a Néel Skyrmion as a function of rescaled DMI strength D/D_c (left panel). These modes increase in frequencies if we go to lower DMI until they eventually disappear in the mangon continuum. The ferromagnetic resonance which represents the start of the Magnon continuum amount to 25 GHz for the given parameters of the sample. The different colours in the plot associate the points with different modes. Detailed analysis for a fixed DMI strength $D/D_c = 0.86$ (right panel). The power spectrum in the right panel was obtained by performing a Fourier transform of the magnetization dynamics [69]. Picture published in [28]

with respect to the DMI. Each mode is represented by a different color, and for the anisotropy-stabilized Skyrmions (modeled using Equation (2.7) at zero magnetic fields), most of the modes are below the magnon gap and emerge close to the critical DMI strength D_c [35, 70].

We excite all modes of the Skyrmion simultaneously with an in-plane magnetic Sinc pulse

$$h(t) = h \cdot \sin(2\pi f \cdot t)/(2\pi f \cdot t)), \tag{4.1}$$

where h is the amplitude of the magnetic field and f is the cutoff frequency of the pulse. To be sure that we excite every magnon-Skyrmion bound state we apply a train of pulses with different cutoff frequencies (25 GHz, 50 GHz, 100 GHz, 200 GHz, 500 GHz). After that, we let the system evolve for 20 ns to avoid any non-eigenmodes, which decay much faster than the eigenmodes of the Skyrmion. Then, we let the eigenmodes evolve for another 10 ns before we calculate the power spectral density (PSD) according to Ref. [71]

$$P_{sd}(f) = \sum_{k=x,y,z} \frac{1}{N} \sum_{i=1}^{N} \left| \sum_{j=1}^{n} m_k(\boldsymbol{r}_i, t_j) e^{-i2\pi f t_j} \right|^2 , \tag{4.2}$$

where we do a discrete Fourier transform on the x-, y- and z-component of every spin in our system. We repeat the same approach for different DMI values, sorting out the points that belong to the same mode and making them noticeable by using the same color for all points that belong to the same mode. Additional to linearization calculations as in Ref. [35] for the breathing mode, elliptical mode, and triangular mode, we also observe the second harmonic of the breathing mode and a gyrotropic mode. The continuous magnon modes are then above the anisotropy gap $\omega = (2\gamma/M_s)(K + Ak^2)$ in which k is the wave number [35]. The result of this analysis is presented in Figure 4.3.

4.2. Analysis of Eigenmodes

After having an overview plot shown in Figure 4.3 we can start by analyzing the different modes in detail. Therefore, we apply an oscillating in-plane magnetic field and analyze the time evolution of this mode.

To study the radius of the Skyrmion we extracted the contour of the Skyrmion. Therefore, we first took the data we got from the time evolution we implemented in **mumax**^{3}, and then we took the absolute value of the data so that we are only sensitive to the out-of-plane (black) and in-plane (white) components, as shown in Figure 4.4 a). Then we find the inner (green) and outer (red) contours of the absolute values with the help of a python package named *skimage* [72], which is a package for image processing. To measure the radius of the Skyrmion, we determine the centroid of the Skyrmion from the contours and calculate the distance of different points on the contour to the centroid. This leaves us with a list of many radii, so that we have a skeleton of the Skyrmion which we can analyze over time to find out the frequencies of the Skyrmion modes. Since we have a complete skeleton of the Skyrmion, we can study the breathing mode as well as the elliptical mode and even more complicated modes.

A time evolution of one point on the contour, while applying an oscillating magnetic field with a frequency of 3.59 GHz, can be seen in Figure 4.4 b). In the beginning, the radius changes a lot since the Skyrmion has to adjust to the frequency of the applied pulse. After 40 ns, the oscillation of the Skyrmion has reached an equilibrium. Because of this, we calculate the Fourier transform of the oscillation for every point on the contour and

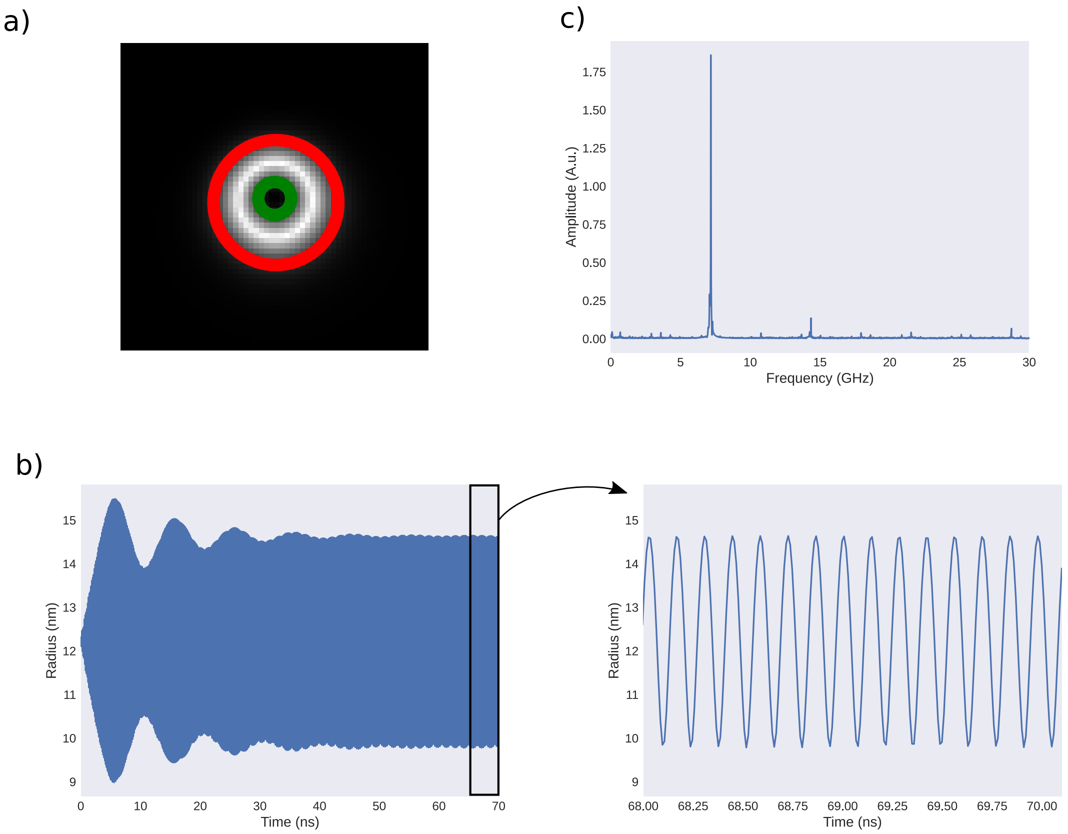

Figure 4.4.: a) Schematic picture of the contour of a Skyrmion. The black colour represents the out-of-plane spins and the white colour the in-plane spins. Therefore the white ring represents the boundary of the Skyrmion. Using the measured package of Python we were able to find the core (green) and the exterior (red) of the Skyrmion. b) Radius of the Skyrmion as a function of time while excited with an oscillating magnetic field with a frequency of 3.59 GHz. Approximately the first 40 ns the Skyrmion mode is in a transient phase, where the eigenmodes are superimposed with the applied pulse. During this phase, the Skyrmion adjusts to the applied frequency until it has reached an equilibrium where only the eigenfrequencies are visible. On the right, we show a zoom of the oscillation after reaching the equilibrium. c) Fourier transform of b) showing the breathing mode at 7.17 GHz as the most prominent mode. There are also some other modes excited by the inplane oscillating magnetic field.

we take the mean of these Fourier transformations to identify the frequency of all modes present in the Skyrmion. In Figure 4.4 c) the breathing mode is most prominent at a frequency of 7.17 GHz. Since the in-plane magnetic field also couples to other modes and multiples of these modes, they are also excited with a much smaller amplitude. We study the excitation of the breathing mode and the elliptical mode in more detail in the following sections.

4.2.1. Breathing Mode

In this section, we study the excitation of the breathing mode ω_0 by different frequencies ω of the magnetic field. In Figure 4.5, we plot the rescaled amplitude r_0/R_0 of the

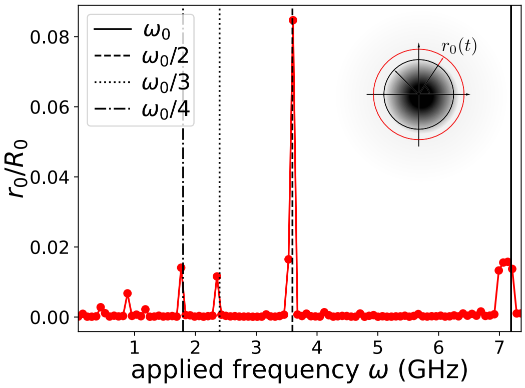

Figure 4.5.: Rescaled amplitude r_0/R_0 of the breathing (ω_0) mode as a function of the frequency of the in-plane applied field $B_{\text{ext}} = 0.05\,\text{T}$ with $\alpha = 10^{-3}$. Here $r_0(t) = r(t) - R_0$, where R_0 is the radius of the relaxed Skyrmion without any excitation. An applied field with fractions of the breathing eigenfrequency can still strongly excite the breathing mode. Picture published in [28].

Skyrmion as a function of the applied frequency, where $r_0(t) = r(t) - R_0$ in which we define R_0 as the radius of the relaxed Skyrmion without any excitation. We noticed that the eigenmodes are excited by applying fractions of the corresponding frequency, i.e. $\omega = \omega_n/m$ where $m \in \mathbb{Z}$. The most striking result is, that the amplitude of the mode excited with half of the eigenfrequency has the highest amplitude.

Furthermore, we studied the excitation of the breathing mode excited by an out-of-plane field. With an out-of-plane field, one only excites the breathing mode which enables us to explore the breathing mode independently from all other modes[1]. We analyze the amplitude of the eigenmode r_n concerning the applied field amplitude B_{ext}, and the material damping α. We applied out-of-plane AC magnetic fields with half and one-third of the breathing mode frequency ω_0.

In Figure 4.6, we show the amplitudes as a function of the applied AC magnetic field strengths B_{ext}, extracted from the frequency spectrum (see right panel), of the breathing mode r_0 and the forced perturbation $\tilde{m}$ at the AC field frequency as a log-log plot. The top (bottom) panel corresponds to $\omega_0/2$ ($\omega_0/3$) for a fix damping constant $\alpha = 10^{-3}$. The second and third harmonic generation requires that the amplitude of the breathing mode grows with power two for $\omega_0/2$ and with power three for $\omega_0/3$. In Figure 4.6 we represent a slope with a power of 2 and a power of 3 by straight lines. For higher applied fields the data points deviate from the straight line. This is because there will be additional higher-order magnon scattering. At a certain value of the amplitude of the applied field, the amplitude of the breathing mode exceeds the one of the forced perturbation, which shows that exciting the breathing mode with fractions of its eigenfrequency can indeed be more efficient than with its eigenfrequency.

[1] Notice that an out-of-plane AC field can only excite the breathing mode of an isolated radially symmetric Skyrmion. Exciting the other modes requires a non-radially symmetric perturbation. This can either be done by other means or by out-of-plane fields accompanied by an additional radial symmetry breaking mechanism such as introducing temperature fluctuations.

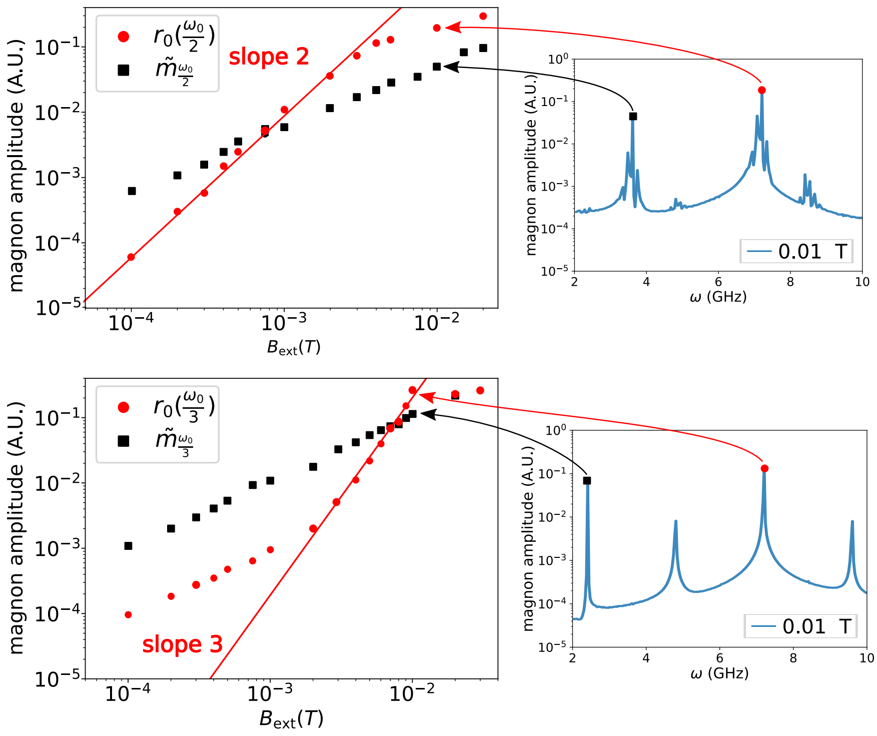

Figure 4.6.: Plots of the amplitude of the excited modes in terms of the amplitude of the applied field with $\alpha = 10^{-3}$. We observe the growth of the second (third) harmonic generation represented by the straight line growing with a power of 2 (3). Above a certain amplitude of the applied field, the breathing mode exceeds the mode with the same frequency. Picture published in [28]

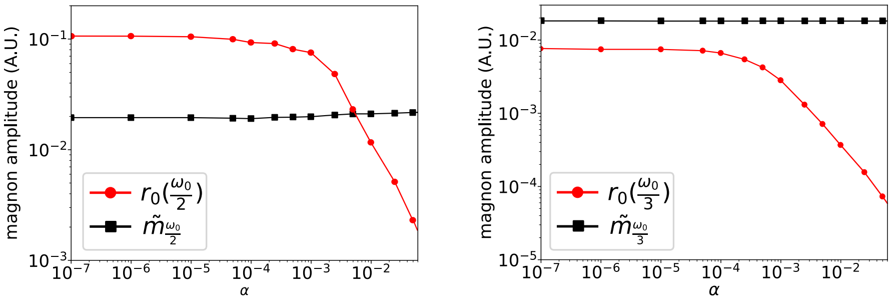

Figure 4.7.: Plots of the amplitude of the excited modes in terms of the damping parameter for an out-of-plane AC magnetic field $B_{\mathrm{ext}} = 0.003$. Picture published in [28]

In Figure 4.7, we show r_0 and $\tilde{m}$ as a function of the damping parameter α for $B_{\mathrm{ext}} = 0.003\,\mathrm{T}$. Analogous to a), the left (right) panel corresponds to an applied AC magnetic field with frequency $\omega_0/2$ ($\omega_0/3$). The forced perturbation amplitude $\tilde{m}$ is constant concerning the damping which shows, that it is completely independent of the damping. However, the breathing mode r_0 reduces in amplitude for higher values of the damping. This means, that the dissipation damps the energy flow from the forced perturbation to the resonant mode. So we find that the frequency multiplication is more efficient for systems with small damping.

4.2.2. Elliptical Mode

We have demonstrated that the breathing mode can be excited with fractions of its eigenfrequency. However, this phenomenon is not exclusive to the breathing mode, as we will show using the elliptical mode, proving that it is possible for other Skyrmion modes as well. Figure 4.5 displays the elliptical mode ($n = 2$) concerning the frequency of the applied oscillating in-plane magnetic field. We extract the radius at various points of the contour as shown in Figure 4.4. Interestingly, the mode is not radially symmetric and rotates during the excitation. Nonetheless, by performing a Fourier transformation of each radius on the contour and taking the average afterwards, we effectively capture even this rotating mode. The elliptical mode can also be excited with fractions of its eigenfrequency. In the next section, we aim to embed these findings in a general theory to prove that this concept is not only applicable to all other Skyrmion modes but also to any nonlinear system.

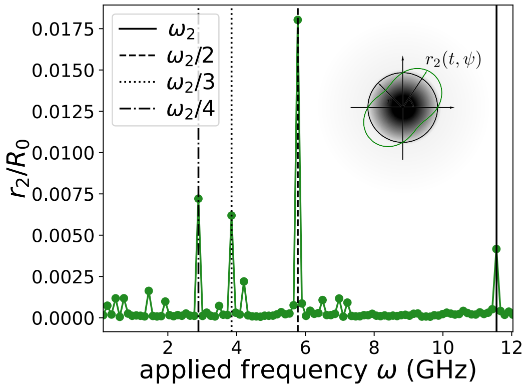

Figure 4.8.: Rescaled amplitude r_0/R_0 of the elliptical (ω_2) mode as a function of the frequency of the in-plane applied field $B_{\mathrm{ext}} = 0.05\,\mathrm{T}$ with $\alpha = 10^{-3}$. The picture published in [28].

4.3. Theory of Frequency Multiplication for Non-Linear Systems

For any nonlinear system with a field ϕ, the dynamics of small perturbations $\tilde{\phi}(t) \ll 1$ of the static background can be expressed by

$$\frac{d\tilde{\phi}}{dt} \approx \mathcal{L}_1(\phi_0)\tilde{\phi} + \mathcal{L}_2\,(\phi_0)\tilde{\phi}^2 + \cdots + \mathcal{L}_p(\phi_0)\tilde{\phi}^p + \cdots \tag{4.3}$$

where the first term, $\mathcal{L}1\tilde{\phi}$, represents a linear approximation for the nonlinear system, yielding eigenstates with frequencies ω_n. The terms $\mathcal{L}k$ with $k > 1$ account for interactions between perturbations, causing a renormalization of the eigenfrequencies ω_n

and resulting in amplitude-dependent frequencies. By applying a perturbation with a fraction of the eigenfrequency, like

$$\tilde{\phi}(t) \approx \phi_{\omega_n/2} \cos(\omega_n t/2) \tag{4.4}$$

the quadratic term, up to constants, will take the form

$$\tilde{\phi}(t)^2 \approx \tilde{\phi}^2_{\omega_n/2} \cos(\omega_n t) \tag{4.5}$$

which corresponds to a linear term of a perturbation with the eigenfrequency. Since this principle can be extended to $m > 2$, this reveals that an eigenmode ω_n can be excited with a fraction of the frequency and the amplitude growths with the mth-power of the driving amplitude with frequency ω_n/m. This general theory can be transferred to magnetic systems. Here, the dynamics of the unitary magnetization $\boldsymbol{m} = \boldsymbol{M}/M_s$, where M_s is the saturation magnetization, is described by the Landau-Lifshitz-Gilbert (LLG) Equation 2.8. If $\boldsymbol{m}$ corresponds for example to a Skyrmion solution, then the excitation modes of the Skyrmion can be viewed as small perturbations of the ground state configuration of $\boldsymbol{m}$ (Figure 4.2) and recover the results of the breathing and elliptical mode of the Skyrmion from before.

4.4. Discussion and Summary

In this project, we study the excitation of Skyrmion modes by applying a fraction of the eigenfrequencies. With the help of micromagnetic simulations, we show that it is possible to excite the breathing mode and the elliptical mode with a fraction of their eigenfrequency as shown in Section 4.2. Moreover, we showed in this example of the Skyrmion that exciting with a fraction of the eigenfrequency increases the amplitude of the response in the low damping regime. Additionally, we showed that for higher values of the Gilbert damping coefficient α, the energy flow from the forced perturbation to the resonant mode is reduced. These results make experimental excitations of Skyrmion modes realistic since lower frequencies are easier to produce and there are no magnons that are resonantly excited at the pumping frequency because the applied frequency is well below the magnon gap. On top of that, the eigenfrequencies are highly tunable since the material parameters can be tuned by temperature changes or applied magnetic and electric fields. This could lead to a device which can provide frequency multiplication for a continuous range of different frequencies.

While we explicitly confirmed this effect using micromagnetic simulations of solitary magnetic Skyrmions' excitation modes, our theoretical analysis demonstrates its universal behaviour, i.e. being independent of microscopic features of the magnetic structures as well as the source of the perturbation shown in Section 4.3. Not only does this provide a new method to excite the eigenmodes by applying lower frequency amplitudes, but this frequency multiplication mechanism also propounds novel applications in spintronics devices such as an in-material frequency multiplier for magnonic applications [73, 74, 75, 76, 77]. In particular, bound modes are not excited via linear resonance and are weakly coupled (e.g. by dipolar stray fields or spin-wave excitations). They serve as a building block for so-called parametrons, a computing scheme which has seen a strong revival in the recent discussions of oscillator-based computing [78]. Furthermore, the tunability of the magnetic textures by altering the material properties, temperature and applied fields makes them very versatile.

Part II.

Quantum Computation with Topological Superconductors

5

Topological Quantum Computation

Quantum computers are advanced computational systems that hold the potential to accelerate numerous processes across various domains. To harness the power of quantum computing, researchers are actively pursuing the development of more stable systems that can better withstand external disturbances. Among these efforts are the endeavours to create topological quantum computers, which leverage the unique properties of materials to enhance stability and enable robust quantum information processing. In the subsequent sections, we will delve into the fundamental principles of quantum computers, exploring how the topological characteristics of materials can be utilised to realise the vision of a topological quantum computer.

5.1. Quantum Computation

The concept of quantum computers arose from the idea of Richard Feynman to compute quantum mechanical systems with a quantum simulator [79]. Compared to classical computers, information is stored in so-called quantum bits (qubits) [80] instead of bits. A qubit can be represented by a Bloch sphere as shown in Figure 5.1 a). For illustration purposes, one considers a conceptual spin inside this Bloch sphere. A qubit, which is a two-state quantum system described by a wave function, can occupy state $|0\rangle$ and state $|1\rangle$, representing the conceptual spin pointing in the $+z$- or $-z$-direction, respectively. These states serve as the basis for measurement outcomes of the qubit's wave function. Qubits possess a significant property known as superposition, enabling them to exist in a combination of both $|0\rangle$ and $|1\rangle$ states simultaneously. This implies that the conceptual spin can be oriented in any direction within the Bloch sphere. Various physical systems can serve as qubits. However, for a quantum computer to be suitable for quantum computation, it must satisfy the DiVincenzo criteria [81]. According to these criteria, a quantum computer must. This implies that the conceptual spin can be oriented in any direction within the Bloch sphere. Various physical systems can serve as qubits. However, for a quantum computer to be suitable for quantum computation, it must satisfy the DiVincenzo criteria [81]. According to these criteria, a quantum computer must

1. consist of a scalable physical system with well-characterized qubits
2. have the ability to initialize the state of the qubits to a simple fiducial state
3. have long relevant coherence times
4. build a universal set of quantum gates

5. provide a qubit-specific measurement capability.

The manipulation of the conceptual spin to change the state of one or multiple qubits is known as a quantum gate. Mathematically, these quantum gates are represented by unitary matrices on an individual basis which act on the state of the qubit. In an experimental setup, such manipulation can be achieved through various means, such as using a laser pulse or a microwave pulse. However, the specific method for manipulating the qubit state depends on the details of the physical system under consideration. To account for imperfections in the experimental manipulation of qubits, the introduction of a quality measure, such as fidelity, becomes necessary to quantitatively assess the accuracy of operations performed on qubits. Fidelity is a metric that assesses the similarity between the desired state and the actual state achieved by a quantum operation. The most intuitive modification of a point on the Bloch sphere by a rotation around the $x-, y-$ and $z-$axis by an arbitrary angle already gives us a complete set of 1-qubit operations with which one can reach every point on the Bloch sphere. In terms of quantum gates, the minimal set needed for universal quantum computation is the Clifford gates together with the $\frac{\pi}{8}$-gate [82]. The Pauli matrices give the Clifford gates, which are only 1-qubit gates but can be generalized to n-qubit gates by combining different Pauli matrices with a tensor product.

$$P_n = \left\{ e^{i\theta\pi/2}\sigma_{j_1} \otimes \cdots \otimes \sigma_{j_n} \mid \theta = 0, 1, 2, 3; j_k = 0, 1, 2, 3 \right\} \tag{5.1}$$

N-qubit gates play a pivotal role in quantum computation by facilitating the creation of entanglement between two qubits. Entanglement refers to the correlation between particles, making the state of one particle inseparable from that of the others, which is of fundamental importance for quantum computations. A representation of the 1-qubit Clifford gates and the $\frac{\pi}{8}$-gate are shown in Figure 5.1 b).

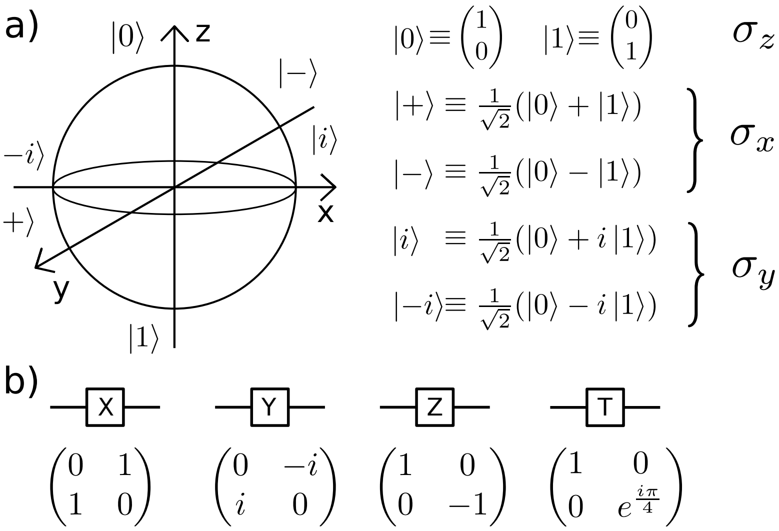
a)
$|0\rangle$
z
$|-\rangle$
$|i\rangle$
$|-i\rangle$
x
$|+\rangle$
y
$|1\rangle$
$|0\rangle \equiv \begin{pmatrix}1\\0\end{pmatrix}$ $|1\rangle \equiv \begin{pmatrix}0\\1\end{pmatrix}$ σ_z
$|+\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|-\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ σ_x
$|i\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$
$|-i\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle)$ σ_y
b)
X
Y
Z
T
$\begin{pmatrix}0&1\\1&0\end{pmatrix}$
$\begin{pmatrix}0&-i\\i&0\end{pmatrix}$
$\begin{pmatrix}1&0\\0&-1\end{pmatrix}$
$\begin{pmatrix}1&0\\0&e^{\frac{i\pi}{4}}\end{pmatrix}$

Figure 5.1.: In a) a representation of a qubit as a Bloch sphere with the different basis states $|0\rangle, |1\rangle, |+\rangle, |-\rangle, |i\rangle, |-i\rangle$ is shown. In b) we also show different quantum operations in the basis of $|0\rangle$ and $|1\rangle$, where the X, Y and Z gates as n-qubit gates together with the $\frac{\pi}{8}$-gate T build a universal set of gates.

5.2. Physical Implementations of Quantum Computation

As quantum computing continues to push the boundaries of technology, exploring different physical systems becomes imperative to uncover their distinctive attributes, paving the way for more versatile and efficient quantum computing platforms. Let's embark on an intriguing exploration of diverse physical systems poised for quantum computation, delving into their unique strengths and limitations. For an overview, we made a table inspired by [83].

Quantum computing platform	Trapped ion qubits	Superconducting qubits	Silicon qubits	Topological quits
Number of qubits	IonQ: 79 (2018) [84], AQT: 24 (2021) [85]	IBM: 127 (2022) [86] Google: 54 (2019) [87]	6 (2022) [88]	In progress
Coherence times	$\sim 18 - 100\,\text{ms}$ [85]	$\sim 16 - 98\,\text{ms}$ [87] [86]	$\sim 2.5 - 45\,\mu\text{s}$ [88]	-
Gate Fidelity	$\sim$ 99.5%-99.9% [85] [89]	$\sim$ 99.9% [87] [86]	$\sim$ 99.9% [88]	Expected: $\sim$ 99.9999%
Gate operation time	$\sim 15\,\mu\text{s}$ [85]	$\sim 12 - 470\,\text{ns}$ [87] [86]	-	-
Scalability	Some potential	Medium to high potential	High potential	-

Trapped Ion Qubits

The fundamental idea is to trap charged ions with magnetic and electric fields to stay in place. The conceptual spin is represented by the population of different energy levels of the ion. The population of the different energy levels can be manipulated by applying a laser pulse, which then relates to single qubit gates. The process of applying a laser pulse to two ions results in the entanglement of those ions. In order to achieve entanglement among multiple ions, segmented traps, as discussed in [90], are employed, offering the flexibility to manipulate atoms along one dimension or within a two-dimensional space. This movement of the ions has to be slow since even the motion of the ions can induce vibrational modes (phonons) that can also have an effect on the qubit. Therefore trapped ion qubits have really long coherence times but quantum gates are slow.

Superconducting qubits

In this approach, one uses a superconducting circuit to represent a qubit. There are different ways to combine electrical components so that the circuit can act as a qubit: In the charge qubit, the circuit consists of a bias voltage, a capacitor, and a Josephson junction; a Flux qubit consists of an inductance, a flux line, and a Josephson junction; a phase qubit consists of a bias current and a Josephson junction. In all circuits, one uses a Josephson junction with which one creates an uneven spacing between the energy levels of the circuit. This is important so that transitions to other energy levels are suppressed. In these circuits, the conceptual spin is represented by the population of the energy levels of the circuit. The population of these energy levels can be controlled via microwave pulses which in turn implement quantum gates. Two circuits can then be coupled by connecting both to an intermediate electrical coupling circuit. The presence of these coupling circuits restricts the entanglement between every qubit, making the construction of a quantum computer significantly less efficient. Overall, superconducting qubits have really short coherence times and the quantum gates are really fast.

Silicon qubits

In these systems, the conceptual spin is directly represented by the electron spin, since in such semiconductor systems, the movement of electrons or holes is restricted in all three directions of space. Quantum gates can be implemented by manipulating the electron spin using electrical and magnetic fields. Multiple qubit gates can be done by modifying the exchange coupling between the electron spins by applying gate voltages. This layout also hinders entanglement between arbitrary qubits. Until now one is not able to reach big qubit systems with silicon qubits but they are expected to have a good scaling towards a high number of qubits.

Topological qubits

Topological qubits function based on a fundamentally distinct principle from conventional systems, overcoming challenges related to manipulating conceptual spins in classical qubits using electric and magnetic fields, which are susceptible to alignment and environmental influences. To tackle this issue in classical qubit systems, quantum error correction is employed, wherein multiple physical qubits are paired to form a logical qubit and specific algorithms are utilized to correct errors. In contrast, topological qubits implement error correction at the hardware level [91, 92, 18], representing the conceptual spin through the occupation of topological excitations that can be manipulated by the spatial exchange.

Since topological qubits are a rather new field, no coherence times or gate times are known yet. However, in the course of this work, we will understand in more detail why the fidelity of the gates is so high and how the exchange of topological excitations can be used for quantum computing. For this purpose, we will go into the theory of topological quantum computation in the next chapter.

5.3. Theory of Topological Quantum Computation

In this chapter, we will discuss the theory of topological quantum computation with the help of a review by Lathinen et. al. [93].

5.3.1. Anyons

In a three-dimensional system, there is only the possibility of Fermions and Bosons. This means that the wavefunction describing the system can get a phase of $e^{i\varphi}$, where φ is π for Fermions and 2π for Bosons upon exchanging two particles. However, in a two-dimensional system, there is a much bigger variety of statistics possible. Apart from bosonic and fermionic exchange statistics, exchanging two particles can lead to arbitrary phase factors or even non-trivial unitary evolutions. This difference can be explained in the following way: In three dimensions every path of a particle encircling another particle can be contracted to one point. So a path λ_1 that does not encircle another particle can be continuously deformed to a path λ_2 which encircles another particle (the path can be deformed to pass behind the other particle), see Figure 5.2.

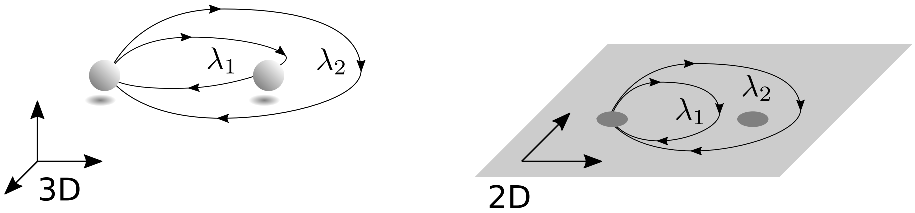

Figure 5.2.: Exchange statistics in 2D vs. 3D. In 3D the path λ_2 describing two particle exchanges is continuously deformable to λ_1 by taking it behind or in front of the right-most particle, and in turn, λ_1 is contractible to a point. Hence, all the paths have the same topology and thus correspond to the same statistical quantum evolution. In 2D, however, the paths λ_2 and λ_1 are topologically inequivalent since λ_2 can not be deformed through the right-most particle, while λ_1 is still contractible to a point. Hence, the paths now have different topologies and different statistical quantum evolutions can be assigned to each [91].

This results in the condition

$$\text{3D}: \quad |\Psi(\lambda_2)\rangle = |\Psi(\lambda_1)\rangle = |\Psi(0)\rangle$$

As one particle encircles the other twice, the evolution of the system can be represented by the exchange operator R such that $|\Psi(\lambda_2)\rangle = R^2 |\Psi(0)\rangle$. The contractibility of the loop requires that $R^2 = 1$, which has only the solutions $R = \pm 1$ that correspond to the exchange statistics of either Bosons or Fermions. Because of this, the order of the exchange does not matter and the statistics can be described by the permutation group. This can not be done in a two-dimensional system since the path λ_2 can not be continuously deformed (the path is not allowed to pass through the encircled particle) to the contractable path λ_1. This means that the final state $|\Psi(\lambda_2)\rangle$ no longer needs to equal the initial state $|\Psi(\lambda_1)\rangle$ resulting in the condition

$$\text{2D}: \quad |\Psi(\lambda_2)\rangle \neq |\Psi(\lambda_1)\rangle = |\Psi(0)\rangle \,.$$

Hence, the exchange operator R is no longer constrained to square to identity either. Instead, it can be represented by a complex phase or even a unitary matrix in the case of a degenerate Hilbert space. In the first case, the anyons are called Abelian anyons due to their commuting exchange operators, while in the latter case, the anyons are referred to as non-Abelian anyons. In comparison to the three-dimensional system, the statistics can be described by the braid group, a mathematical group that characterizes how particles exchange positions in two dimensions.

Anyons can be described by a large theoretical and mathematical construct, which is called topological quantum field theory [94]. In the view of topological quantum computation, most of the mathematical details can be omitted and anyons are the central building block. This minimal model is referred to as the anyon model. It can be conveniently described by a diagrammatic notation, which we will introduce in the following [95]. If we assume that the low energy dynamics is only described by anyons the possible evolutions are limited to three scenarios:

- They can be created or annihilated in a pairwise fashion.
- They can be fused to form other types of anyons.
- They can be exchanged adiabatically.

The anyon model is spanned by some number of particles

$$M = 1, a, b, c, \ldots, \tag{5.2}$$

where the labels $a, b, c, \ldots$ correspond to topological charges carried by anyons, while 1 is a trivial label representing the vacuum state without anyons. These charges must adhere to conservation laws, which are referred to as fusion rules for anyons, and are defined as follows:

$$a \times b = \sum_{c \in M} N_{ab}^{c} c. \tag{5.3}$$

The fusion coefficients $N_{ab}^{c} = 0, 1, \ldots$ indicate the available topological charges (fusion channels) resulting from the combination of particles a and b. When different fusion outcomes occur for the anyons, it implies a state space spanned by various fusion possibilities. This means that a and b can fuse into multiple $c \in M$, allowing for the definition of orthonormal states $|ab; c\rangle$. In this space, we can then define a scalar product

$$\langle ab; c | ab; d\rangle = \delta_{cd}. \tag{5.4}$$

The states $|ab; c\rangle$ and $|ab; d\rangle$ belong to distinct charge sectors, denoted by c and d respectively. The scalar product between these states corresponds to a projective measurement in the fusion space, where we detect the outcome of fusing two anyons. In this fusion space, we have different operations that we can perform on the anyons: The F-matrices and braiding. To explain the F-matrices we have to imagine a system where we have three anyons a, b and c, which are constrained to fuse to d. There are now different possibilities for how to fuse these particles. Either a and b fuse to a particle e and then e fuses with c to d which would be denoted as $|(ab)c; ec; d\rangle$. Or b and c fuse to f and a and f fuse then to d which is denoted by $|a(bc); af; d\rangle$. The F-matrices are then used to connect these two choices by a unitary matrix F_{abc}^{d}

$$|(ab)c; ec; d\rangle = \sum_{f} (F_{abc}^{d})_{ef} \, |a(bc); af; d\rangle \tag{5.5}$$

where $(F_{abc}^{d})_{ef}$ are the matrix elements of F_{abc}^{d}, and f is summed over all the anyons that b and c can fuse to. The diagrammatic representation is shown in Figure 5.3. The braiding operation on the anyons is represented in a similar manner by a matrix R and they describe all possible unitary evolutions that can take place in the fusion space. If we take the basis $|(ab)c; ec; d\rangle$ and we exchange a and b it implements the unitary

$$|(ba)c; ec; d\rangle = \sum_{f} R_{ab}^{f} \delta_{e,f} \, |(ab)c; ec; d\rangle \,, \tag{5.6}$$

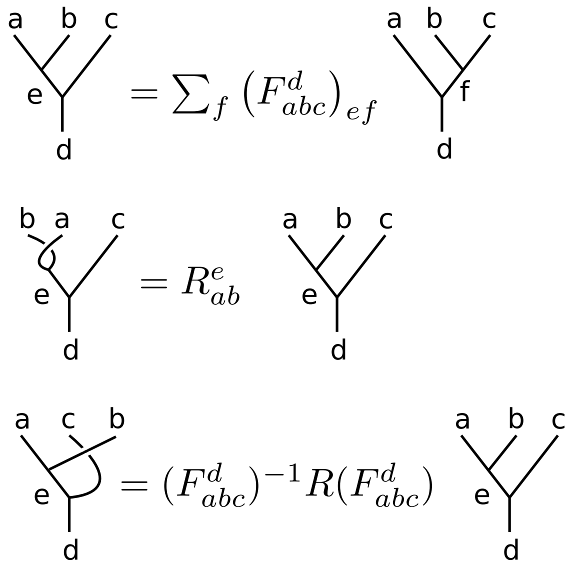
a b c
e
d
$= \sum_f \left(F^d_{abc}\right)_{ef}$
a b c
f
d
b a c
e
d
$= R^e_{ab}$
a b c
e
d
a c b
e
d
$= (F^d_{abc})^{-1}R(F^d_{abc})$
a b c
e
d

Figure 5.3.: Diagrammatic representation of Equation (5.5), (5.6) and (5.7). The lines correspond to the world lines of the anyons and two intersecting lines correspond to fusing two anyons.

where f spans all possible fusion outcomes of a and b and $\delta_{e,f}$ is the Kronecker delta function. However, if we want to exchange b and c in the same basis as before we first have to change the basis $|a(bc); af; d\rangle$, where we fuse b and c first with the matrix F^d_{abc}, then apply R and after that return to the original basis with $(F^d_{abc})^{-1}$. This can be formalised as

$$|(ac)b; ec; d\rangle = (F^d_{abc})^{-1} R F^d_{abc} \, |(ab)c; ec; d\rangle \tag{5.7}$$

The diagrammatic representation of these equations can be seen in Figure 5.3. With combinations of F and R matrices one is able to perform all unitary evolutions due to distinct exchanges of anyons.

After this general introduction of the formalism we will demonstrate it for the particular case of Ising anyons, which are most relevant in our work.

5.3.2. Ising Anyons

This model consists of two non-trivial particles ψ (Fermion) and σ (anyon), which satisfy the fusion rules

$$\begin{aligned} 1 \times 1 = 1, \quad 1 \times \psi = \psi, \quad 1 \times \sigma = \sigma, \\ \psi \times \psi = 1, \quad \psi \times \sigma = \sigma, \\ \sigma \times \sigma = 1 + \psi. \end{aligned} \tag{5.8}$$

According to these fusion rules, two Fermions combine into a trivial particle while a Fermion behaves like a trivial particle when brought together with an anyon. The last fusion rule embodies the non-Abelian nature of the canyon. When fusing two anyons, we can define two fusion channels denoted by $|\sigma\sigma; 1\rangle, |\sigma\sigma, \psi\rangle$, which belong to different charge sectors. In order to have a non-trivial fusion space in the same charge sector we need to consider at least three σ particles that can fuse to a single σ in two distinct ways (or equivalently four σ anyons that fuse to 1). With that, the constrained fusion space can be represented by the states

$$|(\sigma\sigma)\sigma; 1\sigma; \sigma\rangle\,, |(\sigma\sigma)\sigma; \psi\sigma; \sigma\rangle. \tag{5.9}$$

In this space we can then define the F- and R-matrices by [96]

$$\begin{aligned} F = F^{\sigma}_{\sigma\sigma\sigma} &= \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \\ R = \begin{pmatrix}R^{1}_{\sigma\sigma} & 0\\ 0 & R^{\psi}_{\sigma\sigma}\end{pmatrix} &= e^{-i\frac{\pi}{8}}\begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{2}}\end{pmatrix} \end{aligned} \tag{5.10}$$

Here the two left-most σ anyons were exchanged with the R-matrix. If we would define a qubit in this fusion space R would implement the phase gate up to an overall phase factor. If we exchange the right-most anyons we get

$$F^{-1}R^2F = e^{-i\frac{\pi}{4}}\begin{pmatrix}0 & 1\\ 1 & 0,\end{pmatrix} \tag{5.11}$$

which would correspond to a NOT-gate. Furthermore braiding the anyons changes the fusion channel of the two leftmost ones between 1 and ψ. Since one can only implement logical phase and NOT-gates on a single qubit, Ising anyons can only implement the Clifford group by braiding [97, 98]. This means that Ising's anyons, although non-Abelian, are not universal for quantum computation by entanglement. In order to overcome this shortcoming of the Ising anyons one has to introduce the use of non-topological operations. For example, can bring two anyons nearby which lifts the degeneracy of the fusion channels by ΔE and they will dephase in time according to

$$U = \begin{pmatrix}1 & 0\\ 0 & e^{i\Delta Et}.\end{pmatrix} \tag{5.12}$$

Assuming that bringing the anyons nearby and separating them again can be controlled precisely such that $\Delta Et = \pi/4$, then one would have implemented the desired $\frac{\pi}{8}$-phase gate, albeit in a non-topologically protected fashion. This dephasing we studied in the course of the Master thesis by Stephan Kessler [99] In this thesis, we will demonstrate the non-Abelian nature of anyons within a condensed matter system that prominently features a superconductor. In the upcoming chapter, we will introduce the fundamental concepts of superconductors.

6 Superconductivity

In 1911 H. K. Onnes managed to liquefy Helium. He used it to cool mercury down to cryogenic temperatures and performed resistivity measurements on the cooled mercury [100]. The resistivity measurements demonstrated that the resistance undergoes a sudden drop when the temperature falls below a critical value of $T_c = 4.2$ K, signalling the emergence of a superconducting phase characterized by the high conductivity of electrons in the material. [16]. Superconducting phases have also been observed in other metals like Lead (Pb) and Niobium (Nb), which are referred to as superconductors. This started a great effort to find different materials that become superconductors at low temperatures. In 1933 Meissner and Ochsenfeld discovered that superconductors also expel magnetic fields, which has become to be known as the Meissner effect [101]. The first theory that was able to give an explanation of why magnetic fields are expelled in superconductors was given by the London equations [102]. It was put forward by Fritz and Heinz London in 1935. Another phenomenological theory that is to explain the Meissner effect was the Ginzburg-Landau theory (1950) [103] of superconductivity. With the Ginzburg-Landau theory, one was able to divide the superconductors into two categories now referred to as Type I and Type II. In Type I superconductors, there exists a critical magnetic field that, once exceeded, causes the material to lose its superconducting properties. In contrast, Type II superconductors exhibit two critical magnetic fields: The first allows partial penetration of the magnetic field, while the second completely suppresses superconductivity. The microscopic theory of superconductors came in 1957 and was developed by Bardeen, Cooper and Schrieffer and is therefore known as the BCS theory [17, 104].

6.1. The BCS Model

More detailed studies of superconductors led to the observation that the critical temperature of the elemental superconductors is inversely proportional to the square root of the superconductor's elemental mass m, i.e. $T_{\mathrm{c}} \propto 1/\sqrt{m}$ [5]. The phonon frequencies ω_{ph} of an elemental crystal follow the same relation $\omega_{\mathrm{ph}} \propto 1/\sqrt{m}$, which suggests that superconductivity originates from electrons interacting with atoms of mass m.

The main mechanism behind BCS theory is the electron-phonon interaction causing an effective attractive electron-electron interaction. The illustrative example for such an attraction is often imagined as two electrons travelling inside a lattice of positively charged ions, shown in Figure 6.1. The first electron induces a slight displacement of the positively charged ions in its vicinity as it travels through the crystal. These displacements result in a weakly net positive charge density in the path travelled by the first electron. Due to the finite mass of the ions, they use some time to relax back to their original positions. The second electron is then attracted by this path of net positive charge, and hence there is an effective attraction between the electrons. This attraction has been mediated by atom displacements, that is by phonons.

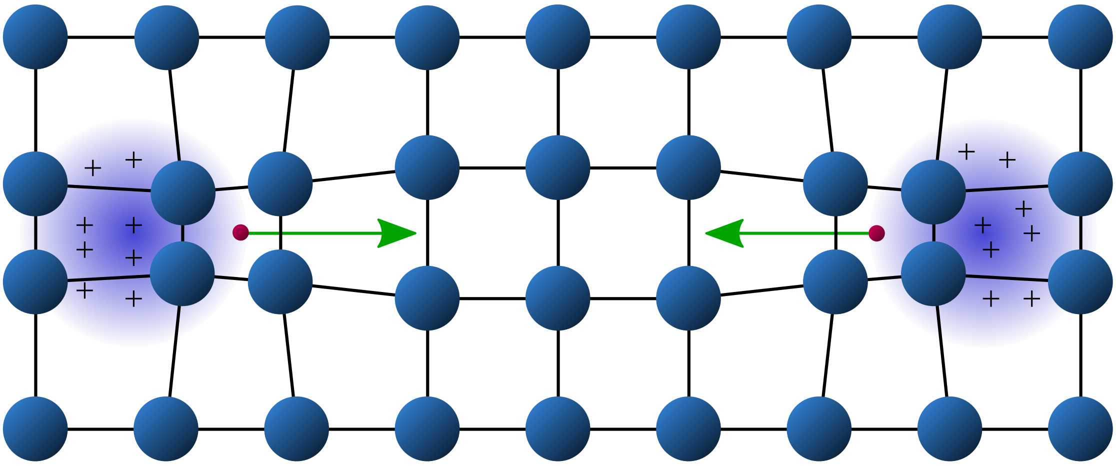

Figure 6.1.: Schematic picture of the electron-electron interaction described by the BCS theory. The movement of the right (left) electron causes lattice vibrations in the material resulting in a positively charged region. The left (right) electron is attracted by this positively charged region. Because of the attraction of the electrons to each other they move as a pair. Such pairs of electrons are called Cooper pairs.

This sketch gives a good view of the microscopics of the superconductor. The attractive potential can be derived properly from a Hamiltonian describing electron-phonon scattering. The resulting effective potential is attractive within a certain frequency range, and repulsive above these frequencies. The simplest version of the BCS theory is therefore built on a simple assumption: There is an attractive potential of strength V between electrons travelling in opposite directions within an energy interval $E_F \pm \hbar\omega_D$ about the Fermi level, where ω_D is the Debye frequency of the phonons.

6.2. BCS Hamiltonian

The BCS Hamiltonian in the continuum limit reads [105]

$$H = \int d\mathbf{r}\; c^\dagger_\alpha(\mathbf{r}) h_{\alpha\beta}(\mathbf{r}) c_\beta(\mathbf{r}) - \frac{V}{2} c^\dagger_\alpha(\mathbf{r}) c^\dagger_\beta(\mathbf{r}) c_\beta(\mathbf{r}) c_\alpha(\mathbf{r}) \tag{6.1}$$

where $c^\dagger_\sigma(\mathbf{r})$ creates an electron with spin σ at position $\mathbf{r}$, $h_{\alpha\beta}$ is the Hamiltonian for non-interacting electrons and V is the onsite potential term. Since it is quartic in the fermionic creation/annihilation operators this Hamiltonian is very difficult to handle. To make the second term quadratic we perform a mean-field approximation

$$AB = \langle A\rangle B + A\langle B\rangle - \langle A\rangle\langle B\rangle \tag{6.2}$$

where operators are partly replaced by their expectation values. To describe superconductivity we will choose the Cooper pair channel which gives us $A = c^\dagger_\uparrow c^\dagger_\downarrow$ and $B = c_\downarrow c_\uparrow$.

Applying Equation (6.2) to the second part of the Hamiltonian (6.1), while excluding the $\langle A\rangle\langle B\rangle$ term since it only adds an energy shift, one gets

$$H = \int d\mathbf{r}\; c_\alpha^\dagger(\mathbf{r}) h_{\alpha\beta}(\mathbf{r}) c_\beta(\mathbf{r}) + \Delta(\mathbf{r}) c_\uparrow^\dagger(\mathbf{r}) c_\downarrow^\dagger(\mathbf{r}) + \Delta^*(\mathbf{r}) c_\downarrow(\mathbf{r}) c_\uparrow(\mathbf{r}) \tag{6.3}$$

where

$$\Delta(\mathbf{r}) := -V\langle c_\downarrow(\mathbf{r}) c_\uparrow(\mathbf{r})\rangle = V\langle c_\uparrow(\mathbf{r}) c_\downarrow(\mathbf{r})\rangle = \frac{V}{2}\rho_{\alpha\beta}\langle c_\alpha(\mathbf{r}) c_\beta(\mathbf{r})\rangle \tag{6.4}$$

which uses $\rho_{\alpha\beta} = (i\sigma_y)_{\alpha\beta}$ with σ_y being a *Pauli matrix*. The quantity $\Delta(\mathbf{r})$, called the *pairing potential*, provides us information about the interaction strength of the Cooper pairs.

6.2.1. Bogoliubov-de Gennes Equation

In this section, we want to rewrite the Hamiltonian (6.3) with new operators γ such that it is diagonal with respect to these operators. Using the standard fermionic anticommutation relations we can calculate the useful commutator

$$[H, c_\sigma(\mathbf{r})] = -h_{\sigma\beta}(\mathbf{r}) c_\beta(\mathbf{r}) - \Delta(\mathbf{r})\rho_{\sigma\beta} c_\beta^\dagger(\mathbf{r}). \tag{6.5}$$

Then we use the *Bogoliubov transformation* to rewrite the c operators in terms of the new γ operators

$$c_\sigma(\mathbf{r}) = \sum_n u_{\sigma n}(\mathbf{r})\gamma_n + v_{\sigma n}^*(\mathbf{r})\gamma_n^\dagger \tag{6.6}$$

where u, v are complex functions. This leads to $H = \sum_n \epsilon_n \gamma_n^\dagger \gamma_n$ with ϵ_n being the eigenvalues in the diagonal basis. This enables us to rewrite Equation (6.5) so that

$$[H, c_\sigma(\mathbf{r})] = \sum_n (-h_{\sigma\beta} u_{\beta n} - \Delta(\mathbf{r})\rho_{\sigma\beta} v_{\beta n})\gamma_n + (-h_{\sigma\beta} v_{\beta n}^* - \Delta(\mathbf{r})\rho_{\sigma\beta} u_{\beta n}^*)\gamma_n^\dagger. \tag{6.7}$$

But we can also insert Equation (6.6) directly into the commutation relation so that

$$\begin{aligned} [H, c_\sigma] &= [H, \sum_n u_{\sigma n}(\mathbf{r})\gamma_n + v_{\sigma n}^*(\mathbf{r})\gamma_n^\dagger] \\ &= \sum_n u_{\sigma n}(\mathbf{r})[H, \gamma_n] + v_{\sigma n}^*(\mathbf{r})[H, \gamma_n^\dagger] \\ &= \sum_n -u_{\sigma n}(\mathbf{r})\epsilon_n \gamma_n + v_{\sigma n}^*(\mathbf{r})\epsilon_n \gamma_n^\dagger \end{aligned} \tag{6.8}$$

which uses the fact that H is diagonal in $\gamma_n^\dagger$, γ_n. Comparing Equation (6.7) and Equation (6.8) produces the following expressions:

$$\epsilon_n u_{\sigma n}(\mathbf{r}) = h_{\sigma\beta} u_{\beta n}(\mathbf{r}) + \Delta(\mathbf{r})\rho_{\sigma\beta} v_{\beta n}(\mathbf{r}) \tag{6.9}$$

and

$$\epsilon_n v_{\sigma n}(\mathbf{r}) = -h_{\sigma\beta}^* v_{\beta n}(\mathbf{r}) - \Delta^*(\mathbf{r})\rho_{\sigma\beta} u_{\beta n}(\mathbf{r}). \tag{6.10}$$

We can rewrite these equations in matrix form

$$\begin{pmatrix} \overset{\leftrightarrow}{h} & \Delta(\mathbf{r})\overset{\leftrightarrow}{\rho} \\ -\Delta^*(\mathbf{r})\overset{\leftrightarrow}{\rho} & \overset{\leftrightarrow}{h}^* \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} = \epsilon_n \begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} \tag{6.11}$$

with

$$\begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} = \begin{pmatrix} u_\uparrow \\ u_\downarrow \\ v_\uparrow \\ v_\downarrow \end{pmatrix}. \tag{6.12}$$

These equations are called the *Bogoliubov-de Gennes equation* (BdG equations). The BdG equations allow us to shift our perspective from creating Fermions to directly creating excitations on top of the superconductor's ground state, which are the Cooper pairs. By diagonalizing the Hamiltonian and solving the eigenvalue Equation (6.11), we gain direct access to the energy spectrum. In this scenario, the resulting particles are bosonic in nature, capable of carrying charge. Due to their bosonic nature, they can undergo condensation. Consequently, all electrons occupy the ground state, preventing scattering, and resulting in a vanishing resistivity.

7 Topological superconductivity

In this chapter, we aim to establish a connection between the preceding chapters by exploring the emergence of topological excitations within the context of superconductivity. These topological excitations can manifest as Majorana zero modes, which can be identified with the Ising anyons mentioned earlier. The conditions under which Majorana modes emerge can be best discussed in a minimal model of an electron chain. Since this model was first introduced by Kitaev [18] it is known as the Kitaev chain toy model. We will also discuss the braiding of Majorana modes within this analytically solvable model and learn how to deal with problems which appear in numerical calculations. After that, we connect two Kitaev chains to obtain a quasi-one dimensional T-shaped structure known as a T-junction, which was introduced by Alicea [24]. In such a T-junction we can study the braiding of Majorana modes more closely and recover their non-Abelian statistics.

7.1. Kitaev Chain

The Kitaev chain toy model can be written as a one-dimensional tight-binding model of a spinless p-wave superconductor, where p-wave superconductivity arises from the pairing between electrons with opposite momenta and opposite spins, and is described by the Hamiltonian

$$H_{\text{chain}} = -\mu \sum_{i=1}^{N} n_i - \sum_{i=1}^{N-1} (t c_i^\dagger c_{i+1} - \mathit{\Delta} c_i c_{i+1} + h.c.). \tag{7.1}$$

The system represents a chain of electrons, and each occupied electron contributes the chemical potential to the total energy. The electrons can hop from one site to another with the hopping energy t and they attract each other with the p-wave superconducting pairing $\mathit{\Delta}$. Here the p-wave type comes from the pairing potential acting on two lattice sites. This pairing has the strength $|\mathit{\Delta}|$ and the phase ϕ and N is the number of sites of the Kitaev chain. This model is schematically shown in Figure 7.1. The Kitaev chain toy model can be rewritten if one considers an electron as two half Fermions shown in the middle of Figure 7.1. Mathematically speaking this can be expressed by writing the electronic operator as a real and an imaginary part[1]:

$$c_i = \frac{1}{\sqrt{2}}(\gamma_{i,1} + i\gamma_{i,2}) \tag{7.2}$$

$$c_i^\dagger = \frac{1}{\sqrt{2}}(\gamma_{i,1} - i\gamma_{i,2}). \tag{7.3}$$

[1] which is a special form of the Bogoliubov transformation shown in Equation (6.6)

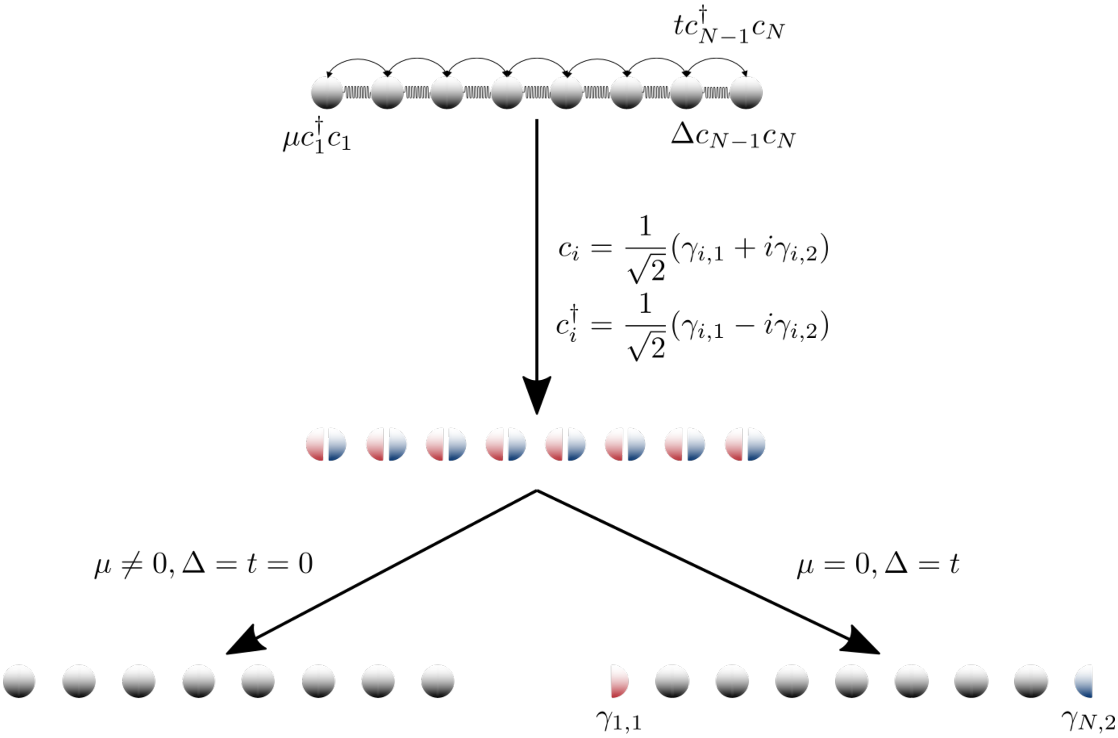
$tc^\dagger_{N-1}c_N$
$\mu c^\dagger_1 c_1$
$\Delta c_{N-1}c_N$
$c_i = \frac{1}{\sqrt{2}}(\gamma_{i,1} + i\gamma_{i,2})$
$c^\dagger_i = \frac{1}{\sqrt{2}}(\gamma_{i,1} - i\gamma_{i,2})$
$\mu \neq 0, \Delta = t = 0$
$\mu = 0, \Delta = t$
$\gamma_{1,1}$
$\gamma_{N,2}$

Figure 7.1.: In the top the Kitaev chain is shown with a graphical representation of the chemical potential, hopping and superconducting term. The arrow to the middle of the picture embodies the transformation in Equation (7.3) and leads to the chain in the middle of the figure, where the electrons are split in a red and blue half. These red and blue halves can then be combined differently depending on the parameters. On the left side, the halves are paired on the same site leading to a chain without any interaction between the sites. On the right side, the halves are paired on different sites leading to a chain without interactions between the electron halves on the same site and two unpaired electron halves.

Here we get two new operators $\gamma_{i,1}$ and $\gamma_{i,2}$, which are their own antiparticles $\gamma_{i,j} = \gamma^\dagger_{i,j}$. These particles are called Majorana particles [106] and obey the Fermion statistics $\{\gamma_{i,\alpha}, \gamma_{i,\beta}\} = 2\delta_{ij}\delta_{\alpha\beta}$. If we rewrite the Hamiltonian now with the new operators we get

$$\begin{aligned} H_{\text{chain}} = &-\frac{\mu}{2}\sum_{i=1}^{N} i\gamma_{i,1}\gamma_{i,2} - \frac{1}{2}\sum_{i=1}^{N-1} i[(t-\mathit{\Delta})\gamma_{i,1}\gamma_{i+1,2} \\ &- (t+\mathit{\Delta})\gamma_{i,2}\gamma_{i+1,1}] - \frac{\mu}{2}. \end{aligned} \tag{7.4}$$

The Hamiltonian in this form allows us to easily study the different limits of the model.

Limit $\mu \neq 0$ **and** $\mathit{\Delta} = t = 0$: In this scenario, the chain lacks interactions between its sites, resulting in a trivial case. However, there is a non-zero value for μ, which can be interpreted as the interaction between half electrons. This interaction is illustrated in the bottom left of Figure 7.1. The corresponding Hamiltonian describes this limit.

$$H_{\text{chain}} = -\frac{i}{2}\mu\sum_{i=1}^{N}\gamma_{i,1}\gamma_{i,2} = -\mu\sum_{i=1}^{N}c^\dagger_i c_i. \tag{7.5}$$

Limit $\mu = 0$ and $\Delta = t$: This is the topological limit where we have the Majorana zero modes $\gamma_{1,1}$ and $\gamma_{N,2}$ and the Hamiltonian transforms to

$$H_{\text{chain}} = -it \sum_{i=1}^{N-1} \gamma_{i,2}\gamma_{i+1,1}. \tag{7.6}$$

Since $\mu = 0$ we don't have any interaction between the electron halves on the same site, which is replaced by an interaction between the electron halves on different sides. When combining the electron halves on different sides, two-electron halves at the end of the chain remain unpaired, as depicted in Figure 7.1. These unpaired electron halves, known as Majorana modes, are of particular interest due to their non-Abelian statistics and potential applications in topological quantum computing. Another feature of this limit is that the Majorana modes at the ends of the chain $\gamma_{1,1}$ and $\gamma_{N,2}$ do not appear in the Hamiltonian (7.6) which means that they have zero energy.

7.2. Eliminating Numerical Phases in Zero Energy Eigenvalues

During the calculation of the Majorana zero modes, we encounter a challenge wherein the nearly degenerate zero energy eigenvalues result in multiple solutions, including linear combinations of the Majorana zero modes with arbitrary phase factors. To extract the Majorana zero modes we perform a two-step procedure, which we demonstrate in the following explicitly for two Majorana zero modes[2]. Note that numerically we are computing a system of finite size. Therefore, instead of having two identical zero eigenenergies, we have a small gap and thus we can define wave functions with energy $E \gtrsim 0$ and $E \lesssim 0$, because of particle-hole symmetry. As we are working in the BdG formalism every eigenstate χ can be decomposed into an electronic part, $\vec{u} = (u_{1\uparrow}, u_{1\downarrow}, \ldots, u_{N\uparrow}, u_{N\downarrow})^T$, and a hole part, $\vec{v} = (v_{1\uparrow}, v_{1\downarrow}, \ldots, v_{N\uparrow}, v_{N\downarrow})^T$, as

$$\chi = \begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} . \tag{7.7}$$

Additionally, the particle-hole symmetry comes with a gauge choice in the sense that any eigenstate with positive energy

$$\chi_{E\gtrsim 0} = e^{\mathrm{i}\kappa} \begin{pmatrix} \vec{u} \\ \vec{v} \end{pmatrix} , \tag{7.8}$$

has a corresponding eigenstate, its particle-hole partner, with negative energy,

$$\chi_{E\lesssim 0} = \mathcal{P}\chi_{E\gtrsim 0} = e^{-\mathrm{i}\kappa} \begin{pmatrix} \vec{v}^* \\ \vec{u}^* \end{pmatrix} . \tag{7.9}$$

Here $\mathcal{P} = \tau_x\mathcal{K}$ is the particle-hole operator, consisting of the operation of exchanging particles and holes combined with complex conjugation. We have explicitly taken into

[2] To disentangle multiple zero energy modes we iterate this procedure.

account an arbitrary, artificial phase factor $e^{\mathrm{i}\kappa}$, which can arise upon diagonalization. To determine the wave functions and get rid of the arbitrary phase factor, we first write

$$\chi_{E\gtrsim 0} = \eta\,\chi_A + \sqrt{1-\eta^2}\,\chi_B, \tag{7.10}$$

as a linear combination of the computed normalized eigenvectors χ_A and χ_B, where we optimize the parameter η to ensure that the wave function has equal weights at both vortex positions. Then we compute its particle-hole partner.

In a second step we compute the Majorana basis [107]

$$\begin{aligned}\chi_\mathrm{l} = \begin{pmatrix}\vec{u}_\mathrm{l}\\ \vec{v}_\mathrm{l}\end{pmatrix} &= \frac{1}{\sqrt{2}}(\tilde{\chi}_{E\gtrsim 0} + \tilde{\chi}_{E\lesssim 0})\\ &= \frac{1}{\sqrt{2}}\begin{pmatrix}e^{\mathrm{i}(\kappa-\beta)}\vec{u} + e^{-\mathrm{i}(\kappa-\beta)}\vec{v}^{*}\\ e^{\mathrm{i}(\kappa-\beta)}\vec{v} + e^{-\mathrm{i}(\kappa-\beta)}\vec{u}^{*}\end{pmatrix},\end{aligned} \tag{7.11}$$

$$\begin{aligned}\chi_\mathrm{r} = \begin{pmatrix}\vec{u}_\mathrm{r}\\ \vec{v}_\mathrm{r}\end{pmatrix} &= \frac{\mathrm{i}}{\sqrt{2}}(\tilde{\chi}_{E\gtrsim 0} - \tilde{\chi}_{E\lesssim 0})\\ &= \frac{\mathrm{i}}{\sqrt{2}}\begin{pmatrix}e^{\mathrm{i}(\kappa-\beta)}\vec{u} - e^{-\mathrm{i}(\kappa-\beta)}\vec{v}^{*}\\ e^{\mathrm{i}(\kappa-\beta)}\vec{v} - e^{-\mathrm{i}(\kappa-\beta)}\vec{u}^{*}\end{pmatrix},\end{aligned} \tag{7.12}$$

where we introduced $\tilde{\chi}_{E\gtrsim 0} = e^{-\mathrm{i}\beta}\chi_{E\gtrsim 0}$ and $\tilde{\chi}_{E\lesssim 0} = e^{\mathrm{i}\beta}\chi_{E\lesssim 0}$. We optimize β such that χ_l and χ_r are equally localized at the two ends of the chain. We chose the labels ‘l’ and ‘r’ for the initial left and right positions, respectively. Note that the left/right Majorana zero mode has also a small weight localized at the position of the right/left one, which is exponentially suppressed with the length of the chain.

In the next section, we will show that Majorana modes represent the Ising Anyons we discussed before and can therefore build a Majorana qubit.

7.2.1. Majorana Qubit

In the following, we will demonstrate that a topological superconductor has the capability to realize the Ising Anyons introduced in Section 5.3.2. For that, we need to associate the three particles 1, ψ and σ of the Ising Model to particles in topological superconductors. The localised Majorana modes can be seen as a pair of σ Anyons. Next, we will define the ground state of the topological phase in the absence of Majorana zero modes as 1. The fermionic particles correspond to ψ and the fusion rule $\psi \times \psi = 1$ means nothing else than that two Fermions can pair up to form a Cooper pair that vanishes into the condensate, thus becoming part of the ground state 1. This also means that only the parity of the ψ Fermions is conserved. When we consider a system of two Majorana zero modes corresponding to σ Anyons, then the state with eigenvalue of $d^\dagger d = (1 + i\gamma_1\gamma_2)/2$ being 0 can be identified with the fusion channel $\sigma \times \sigma \to 1$ and the state with eigenvalue 1 will then represent the fusion channel $\sigma \times \sigma \to \psi$. So the topological phase consists of the two fusion channels

$$i\gamma_1\gamma_2 \left|\sigma\sigma;1\right\rangle = -\left|\sigma\sigma;1\right\rangle, \quad i\gamma_1\gamma_2 \left|\sigma\sigma;\psi\right\rangle = +\left|\sigma\sigma;\psi\right\rangle \tag{7.13}$$

where $\left|\sigma\sigma;\psi\right\rangle = d^\dagger \left|\sigma\sigma;1\right\rangle$. With $\mathcal{P} = i\gamma_1\gamma_2$ as the parity operator, it comes apparent that these fusion channels belong to different parity sectors since they have different signs and can therefore not form a coherent superposition. This in turn means that they cannot form the basis of a qubit. To form a qubit we need four Majorana modes [108, 109, 110]. Here, we choose the even parity sector ($\mathcal{P} = +1$) to encode a physical qubit

$$|0\rangle \equiv |\sigma\sigma; 1\rangle\, |\sigma\sigma; 1\rangle\,, \quad |1\rangle \equiv |\sigma\sigma; \psi\rangle\, |\sigma\sigma; \psi\rangle\,. \tag{7.14}$$

With the qubit defined, we can discuss how braiding can be implemented and related to the braiding operator in Equation (5.11) for the Ising Anyons.

7.3. Braiding in a Kitaev Chain

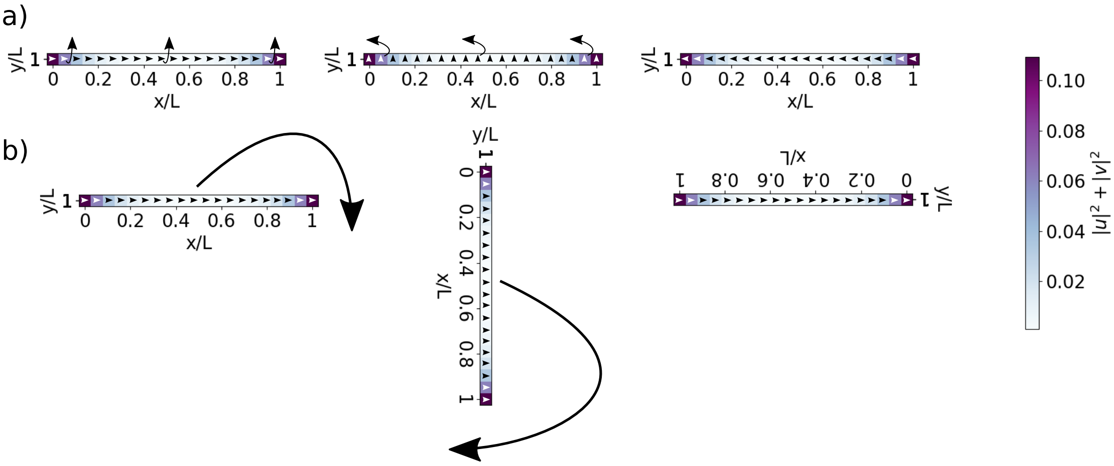

Figure 7.2.: a) Rotation of the superconducting phase of the Kitaev Chain from the view of the sample. b) Rotation of the sample while the superconducting phase is pointing to the right. The color bar is showing the probability density of the Majorana mode.

Generally, the operations of topological quantum computation can be carried out by braiding of Anyons. Any braiding operation can be split into a sequence of braids involving only two Majoranas. The problem with a completely one-dimensional model is that the notion of braiding is not directly imprinted in the model. This is because one cannot move the Majorana modes through each other without merging them back into an electron. So one has to think of a way that effectively changes the position of the Majoranas. The only thing in the model that gives the notion of a direction is the superconducting phase ϕ. So when we rotate this phase we effectively rotate the Majorana modes around each other. A visualization of that can be seen in Figure 7.2. In a) we keep the sample fixed and show how the superconducting phase turns from pointing to the right to pointing to the left. In scenario b), we maintain a fixed superconducting phase directed towards the right. This causes the sample to rotate, leading to a rotation of the Majorana modes at the ends around themselves. Consequently, this rotation results in the braiding of the Majorana modes within the Kitaev chain.

7.4. Analytical Braiding in the Kitaev Chain Toy Model

To justify the algorithm that we developed in Section 7.2 we study the effective braiding of Majoranas analytically in a Kitaev chain as we described it before. For that, we have to derive the Kitaev chain in the topological phase. In the sweet spot of the topological limit described by $\mu = 0$, $t = \Delta$ and $\Delta = |\Delta| \cdot e^{-i\phi}$, the Kitaev chain Hamiltonian reduces to

$$H_{\text{chain}}^{\text{top}} = -it \sum_{i=1}^{N-1} \gamma_{i,2}\gamma_{i+1,1}. \tag{7.15}$$

Two possible zero energy eigenmodes as a function of ϕ in the Bogoliubov-de Gennes formalism, are given by [111]

$$w_A(\phi) = \left\{e^{-i\phi}, 0, \cdots, 0; 1, 0, \cdots, 0\right\} / \sqrt{2} \tag{7.16}$$

$$w_B(\phi) = \left\{0, \cdots, 0, -e^{-i\phi}; 0, 0, \cdots, 1\right\} / \sqrt{2}, \tag{7.17}$$

where the first part in front of the semicolon represents the electronic part and the second part represents the hole part of the eigenvector. In the first step, we combine the two eigenvectors as in Equation (7.10). For $\eta = 1/\sqrt{2}$ we obtain a pure electronic wave function

$$w_{E\gtrsim 0}(\phi) = (w_A(\phi) + w_B(\phi))/\sqrt{2} = \left\{e^{-i\phi}, 0, \cdots, -e^{-i\phi}; 1, 0, \cdots, 1\right\} / 2 \tag{7.18}$$

with its particle-hole symmetric partner

$$w_{E\lesssim 0} = \mathcal{P} w_{E\gtrsim 0}(\phi) = \left\{1, 0, \cdots, 1; e^{i\phi}, 0, \cdots, -e^{i\phi}\right\} / 2. \tag{7.19}$$

With the ansatz $\tilde{w}_{E\gtrsim 0} = e^{-i\beta} w_{E\gtrsim 0}$ and $\tilde{w}_{E\lesssim 0} = e^{i\beta} w_{E\lesssim 0}$ we determine β from Equations (7.11) and (7.11) such that we have a left and a right localized mode. This yields a phase $\beta = \phi/2$ and the Majorana modes

$$w_{\mathrm{l}}(\phi) = \left\{-\frac{ie^{-i\phi/2}}{\sqrt{2}}, 0, \cdots, 0; \frac{ie^{i\phi/2}}{\sqrt{2}}, 0, \cdots 0\right\} \tag{7.20}$$

$$w_{\mathrm{r}}(\phi) = \left\{0, 0, \cdots, \frac{e^{-i\phi/2}}{\sqrt{2}}; 0, 0, \cdots, \frac{e^{i\phi/2}}{\sqrt{2}}\right\}. \tag{7.21}$$

The difference between $w_A(\phi), w_B(\phi)$ and $w_{\mathrm{r}}, w_{\mathrm{l}}$ gets apparent when we calculate the Berry matrix given by

$$\varGamma_{jk} = -i \int_0^{\varPhi} d\phi \, \langle w^{(j)}(\phi)| \, \partial_\phi |w^{(k)}(\phi)\rangle. \tag{7.22}$$

For the electronic part of $w_A(\phi), w_B(\phi)$ we get $\varGamma_{jk} = -\frac{\varPhi}{2}\delta_{jk}$ and for the electronic part of $w_{\mathrm{l}}, w_{\mathrm{r}}$ we get $\varGamma_{jk} = -\frac{\varPhi}{4}\delta_{j,k}$.

Next, we will discuss how we can calculate the exchange statistics of the Majorana zero modes in the Kitaev chain toy model. For that, we have to exchange the Majorana zero modes as we discussed before by changing ϕ from 0 to π. This means that our Majorana modes before the exchange with $\phi = 0$ have the following form

$$w_{\mathrm{l}}(0) = \left\{-\frac{i}{\sqrt{2}}, 0, \cdots, 0; \frac{i}{\sqrt{2}}, 0, \cdots 0\right\} \tag{7.23}$$

$$w_{\mathrm{r}}(0) = \left\{0, 0, \cdots, \frac{1}{\sqrt{2}}; 0, 0, \cdots, \frac{1}{\sqrt{2}}\right\}. \tag{7.24}$$

After the exchange, we have that $\phi = \pi$ resulting in the form

$$w_{\mathrm{l}}(\pi) = \left\{-\frac{1}{\sqrt{2}}, 0, \cdots, 0; -\frac{1}{\sqrt{2}}, 0, \cdots 0\right\} \tag{7.25}$$

$$w_{\mathrm{r}}(\pi) = \left\{0, 0, \cdots, -\frac{i}{\sqrt{2}}; 0, 0, \cdots, \frac{i}{\sqrt{2}}\right\}. \tag{7.26}$$

But the two edges are interchanged too. So overall the exchange resulted in a transformation of the coefficients $w_i^{(\mathrm{l})}(\pi) \to w_i^{(\mathrm{r})}(0)$ and $w_i^{(\mathrm{r})}(\pi) \to -w_i^{(\mathrm{l})}(0)$. Expressed as a matrix of overlaps we get

$$(M_{lr})_{jk} = \sum_i [w_i^{(j)}(\phi = \pi)]^* w_i^{(k)}(\phi = 0) = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{7.27}$$

We can express this matrix also in the operator form by remembering that the Majorana operator is given by the Bogoliubov-de Gennes formalism. The two zero energy Majorana operators are given by $\gamma_j = \sum_i (w_i^{(j)} c_i + [w_i^{(j)}]^* c_i^\dagger)$, where $j = \mathrm{l}, \mathrm{r}$. They satisfy the anticommutation relations $\{\gamma_j, \gamma_k\} = 2\delta_{jk}$, provided $\sum_i [w_i^{(j)}]^* w_i^{(k)} = \delta_{jk}$. Therefore the exchange leads to the transformation in operator form as $\gamma_{\mathrm{l}} \to \gamma_{\mathrm{r}}$ and $\gamma_{\mathrm{r}} \to -\gamma_{\mathrm{l}}$ which agrees with the work of Ivanov [29]. The exchange operator can then be defined as

$$\mathcal{R}_{lr} = e^{-i\frac{\pi}{4}\sigma_z} = e^{-\frac{\pi}{4}\gamma_l\gamma_r} = \frac{1}{\sqrt{2}}\Big(1 - \gamma_l\gamma_r\Big). \tag{7.28}$$

This exchange operator relates to the exchange operator of the Ising Anyons in Equation (5.10) by an overall phase factor (and the direction of the exchange).

7.4.1. Numerical Braiding in the Kitaev Chain Toy Model

To diagonalize Equation (7.1), it is very useful to write the Hamiltonian according to the Bogoliubov-de Gennes formalism [112], meaning that our Hamiltonian will be given by $H = \frac{1}{2} C^\dagger H_{\mathrm{BdG}} C$ with C a column vector containing all creation and annihilation operators, $C = (c_1, ..., c_N, c_1^\dagger, ..., c_N^\dagger)^T$, where the first part of the vector corresponds to holes, and the second part corresponds to electrons. The $2N \times 2N$ matrix H_{BdG} can be written in a compact way using Pauli matrices τ_i in particle-hole space, and denoting with $|n\rangle$ a column basis vector $(0, ..., 1, ..., 0)^T$ corresponding to the n-th site of the chain. With these definitions, the Bogoliubov-de Gennes Hamiltonian is given by

$$H_{\mathrm{BdG}} = -\mu \sum_{n=1}^{N} \tau_z \otimes |n\rangle\langle n| - \sum_{n=1}^{N-1} [(t\tau_z + i\Delta\tau_y) \otimes |n\rangle\langle n+1| + h.c.]. \tag{7.29}$$

In this form, we can get the energy spectrum by calculating the eigenvalues ϵ_n and eigenvectors $(\vec{u}, \vec{v})^T$ similarly to Equation (6.11). The phase of the zero energy modes

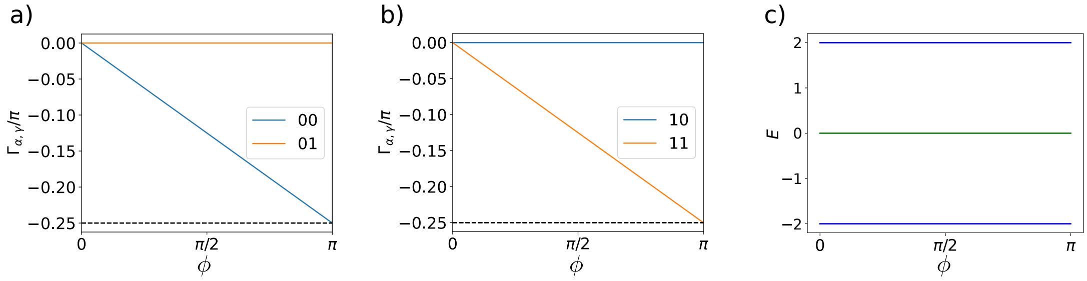

Figure 7.3.: a+b) Numerical calculation of Equation (7.22). We plot $j = 0, k = 0$ (blue), $j = 0, k = 1$ (orange) in the left figure and $j = 1, k = 0$ (blue), $j = 1, k = 1$ (orange) in the middle figure. ϕ characterizes the angle of the superconducting phase. c) Eigenspectrum of the Hamiltonian Equation (7.29) with $\mu = 1,7$, $t = |\Delta| = 1$ as a function of the superconducting phase ϕ.

$w_\mathrm{l}, w_\mathrm{r}$ is corrected by the algorithm presented in Section 7.2. We numerically calculated the Berry matrix and the energy spectrum for $\mu = 1,7$, $t = |\Delta| = 1$ and the results can be seen in Figure 7.3. The numerical calculations of the Berry matrix confirm the results obtained from the analytical calculation. The energy spectrum demonstrates that the zero-energy modes remain well-separated from the high-energy modes as the superconducting phase changes from 0 to π. Additionally, we computed the overlap of the eigenvectors before and after the process, which agrees with the matrix given in Equation (7.27).

In the next section, we extend the model of the Kitaev chain to a model, where one is also able to braid Majoranas in quasi-two dimensions. This also allows us to study the exchange of Majorana zero modes in a system with four Majorana zero modes.

7.5. T-Junction

Instead of braiding the Majorana modes effectively with the help of the superconducting phase, we can also combine two Kitaev chains such that they build a T-junction. The braiding is done in real space through a suitable manipulation of the chemical potential in this effectively one-dimensional wires [24]. Going beyond the quasi-one-dimensional system considered in the literature, we numerically study the T-junction in a full two-dimensional system. This is achieved by constructing a two-dimensional p-wave superconductor and selectively inducing the topological phase in specific regions, forming a topological T-junction structure within the superconductor, while keeping the remaining parts in the trivial phase. In the following, we will outline how the system shown in Figure 7.4 can be implemented with the software that was written in the course of this thesis. The software is based on a code that was written by Kjetil Hals [113]. We consider a two-dimensional superconductor with $p_x \pm \mathrm{i}p_y$-wave pairing described by the Hamiltonian

$$H = -\mu \sum_i c_i^\dagger c_i - t \sum_{\langle i,j \rangle} (c_i^\dagger c_j + h.c.) + \Delta_x \sum_i (c_i^\dagger c_{i+\hat{x}}^\dagger + h.c.) + \Delta_y \sum_i \left(c_i^\dagger c_{i+\hat{y}}^\dagger + h.c. \right). \tag{7.30}$$

Here μ is the chemical potential, t is the hopping parameter, Δ_x is the pairing in the x-direction and Δ_y is the pairing in the y-direction. In this two-dimensional system, every index for lattice sites corresponds to $i = (x, y)$ and the symbol $\langle i, j \rangle$ indicates a summation of the nearest-neighbour lattice sites. Since Majorana zero modes appear at the ends of a topological region in a superconductor, we control the phase transition in our system by changing the chemical potential μ. If $\mu < 2t$ the superconductor is in the topological phase and if $\mu > 2t$ the superconductor is in the trivial phase. To imprint the T-junction structure in our system we drive the chemical potential in the region outside of the T-junction to the trivial phase and make it a constant. Inside the T-junction we allow the chemical potential to change. This corresponds to the situation shown in Figure 7.4, where the electrodes will be able to change the chemical potential locally and define the T- Junction in the system. With the T-junction imprinted in the two-dimensional superconductor, we study how two and four Majoranas can be generated and exchanged to verify their statistics. In the case of two Majoranas, we drive the horizontal wire into the topological phase to generate Majorana modes on the left and the right side as shown in Figure 7.5 (a). The exchange procedure of the Majorana modes is as follows [111]:

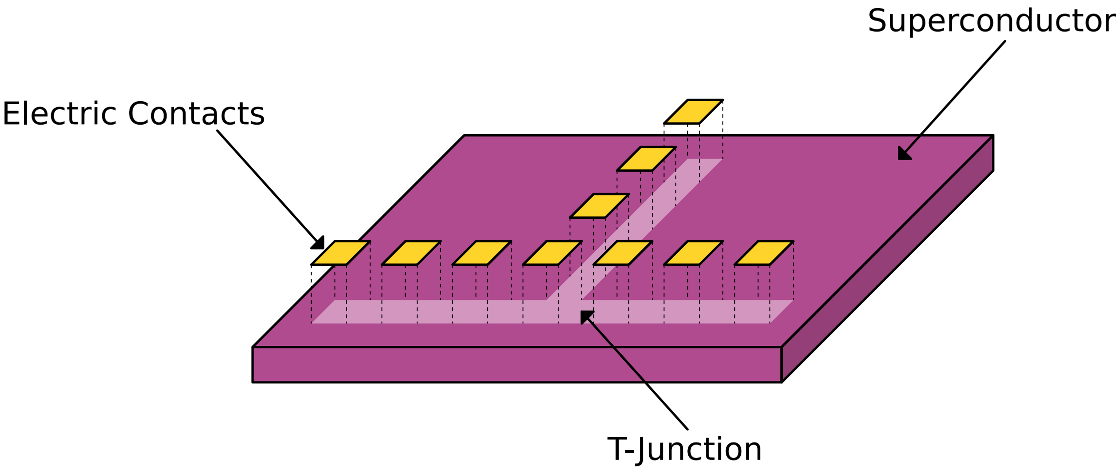

Figure 7.4.: Depiction of a quasi-two-dimensional superconductor with $(p_x \pm i\ p_y)$-wave pairing where the T-junction is controlled by electric contacts.

1. Move the left Majorana to the top of the vertical wire over the connection point of the vertical and horizontal wire.

2. Move the right Majorana to the left of the horizontal wire.

3. Move the Majorana in the vertical wire to the right of the horizontal wire.

In the case of four Majoranas, we drive a few sites of the horizontal wire on the left and the right in the topological phase as shown in Figure 7.5. With that, we have two Majoranas on the left side and two Majoranas on the right side of the horizontal wire. The exchange of the two central Majoranas follows the same procedure as for the case of two Majoranas. Besides that now the middle part of the bottom wire is in a trivial phase.

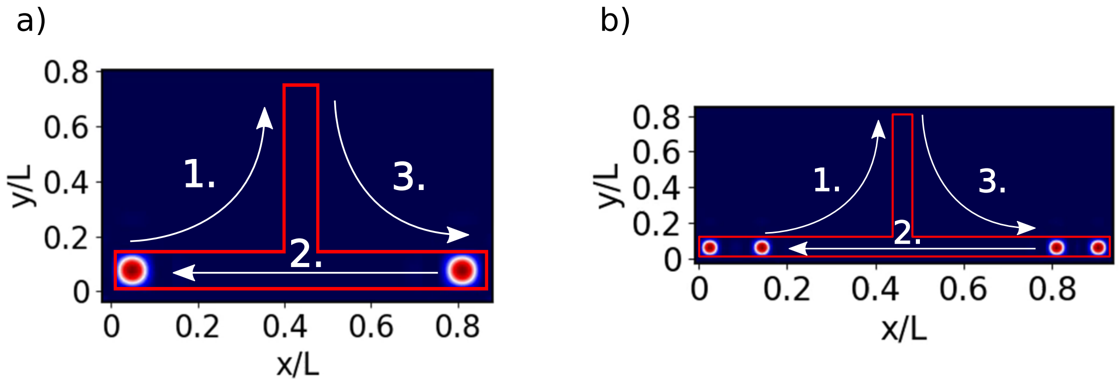

Figure 7.5.: Schematic picture of braiding two Majoranas (a) and four Majoranas (b) in a quasi-two-dimensional superconductor with a T-junction shown by a red box.

7.5.1. Numerical Implementation

Since the T-junction shape is much more difficult to calculate analytically we will calculate the Hamiltonian in the Equation (7.30) numerically by bringing it to the form of Equation (6.11) but without considering spin. This then results in the following form of the BDG Hamiltonian

$$\begin{pmatrix} \overset{\leftrightarrow}{h} & \Delta(\mathbf{r}) \\ -\Delta^*(\mathbf{r}) & \overset{\leftrightarrow}{h}^*, \end{pmatrix} \tag{7.31}$$

where $\overset{\leftrightarrow}{h}$ is given by the kinetic energy of free particles $\overset{\leftrightarrow}{h}_{\text{kin}} = -\frac{\hbar}{2m}(\partial_x^2 + \partial_y^2)$ and $\Delta(\mathbf{r})$ embodies the $p_x \pm i\ p_y$-wave pairing. We can relate the parameters in Equation (7.30) by discretizing $\overset{\leftrightarrow}{h}_{kin}$ and $\Delta(\mathbf{r})$. Here we choose to discretize the Hamiltonian on a square lattice where the neighbouring sites have a distance a to each other. Therefore, we use the *central differential approximation*

$$\begin{aligned} \frac{df}{dx} &\approx \frac{f_{i+a} - f_{i-a}}{2a} \\ \frac{d^2 f}{dx^2} &\approx \frac{f_{i+a} - 2f_i + f_{i-a}}{a^2} \end{aligned} \tag{7.32}$$

where i indicates a spatial site. By applying the central differential approximation to $\overset{\leftrightarrow}{h}_{\text{kin}}$ we then get

$$\overset{\leftrightarrow}{h}_{\text{kin}}\vec{u} = V_{i,j}\vec{u}_{i,j} + t(\vec{u}_{i+a,j} + \vec{u}_{i,j+a}) + t^\dagger(\vec{u}_{i-a,j} + \vec{u}_{i,j-a}), \tag{7.33}$$

with the onsite term $V_{i,j} = (\frac{2\hbar^2}{ma^2} - \mu)\mathbb{1}$ where $\mathbb{1}$ is the identity matrix and μ is the chemical potential. And the hopping parameter $t = -\frac{\hbar^2}{2ma^2}\mathbb{1}$. The superconducting pairing gives us

$$\Delta(\mathbf{r})\vec{u} = \Delta_x\vec{u}_{i+a,j} + \Delta_x^\dagger\vec{u}_{i-a,j} + \Delta_y\vec{u}_{i,j+a} + \Delta_y^\dagger\vec{u}_{i,j-a}. \tag{7.34}$$

Having the numerics set up we can study the braiding of Majorana zero modes in the T-junction.

7.5.2. Numerical Braiding in the T-Junction

To braid the Majorana zero modes in a numerical setup starts with the situation in Figure 7.6 where both Majorana zero modes are localized in the bottom wire. The region between the Majoranas is driven in the topological phase and the rest of the system is in the trivial phase. We can then move one of the Majoranas by driving the site where the left Majorana is placed to the trivial phase. This results in the Majorana localizing one site more to the right. With this procedure of moving the Majorana modes, we can implement the exchange of the Majoranas described before. When the two Majoranas are right above each other we perform the transformation $\Delta_y \to -\Delta_y$ to make sure that we don't get any spurious zero-energy modes as mentioned in the work of Alicea et al [24]. For each of these adiabatic steps t [3] we then solve the eigensystem. The resulting exchange of two Majorana modes is shown in Figure 7.6 a). In Figure 7.6 b) we show the absolute overlap of the initial electronic wavefunction with the electronic wavefunction after different adiabatic steps. After part one, only half of the wavefunction stayed at the same places since we moved one of the Majorana modes to the vertical wire. During the second part, the overlap goes to zero meaning that no Majorana is in the same position as before. But at the end of the second part, we are again at an overlap of $\langle\psi_i|\psi_f\rangle = 0.5$. When we proceed with the third part, we reproduce the situation of the beginning apart from a non-abelian phase which results again in a perfect absolute overlap. In c) we show how the energy of the different modes around zero energy transform. The important observation here is, that the two zero-energy Majorana modes stay well separated from the high energy modes during the whole process. This justifies our transformation from the electronic wavefunctions to the Majorana wavefunction later on. Only the high-energy modes are changing under the exchange of the Majorana modes.

If we numerically calculate the overlap matrix (7.27) we get

$$M_{12} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} , \tag{7.35}$$

where we numbered the Majoranas from left to right. This recovers the same exchange operator we had for the Kitaev chain. In the T-junction, we can now also implement a system with four Majoranas numerically. Therefore we drive some sites on the left and right side in the topological phase and leave the rest of the system in the trivial phase. The movement of the Majoranas can then be done by driving the site on the right to the second Majorana in the topological phase. The second Majorana will then localize on the site to the right resulting in a movement to the right. The exchange of four Majorana modes is shown in Figure 7.7

The numerical calculation of the overlap matrix for the second and third Majorana then leads to the matrix

$$M_{23} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} . \tag{7.36}$$

Expressed in the Majorana operators we get $\gamma_2 \to \gamma_3$ and $\gamma_3 \to -\gamma_2$. Since the braiding involves the pair γ_2 and γ_3 it is convenient to consider the two qubits composed out of the Majorana modes (γ_2, γ_3) and (γ_1, γ_4). Therefore we introduce the basis [114] $|n_+ n_-\}$,

[3] We don't do a full time evolution here.

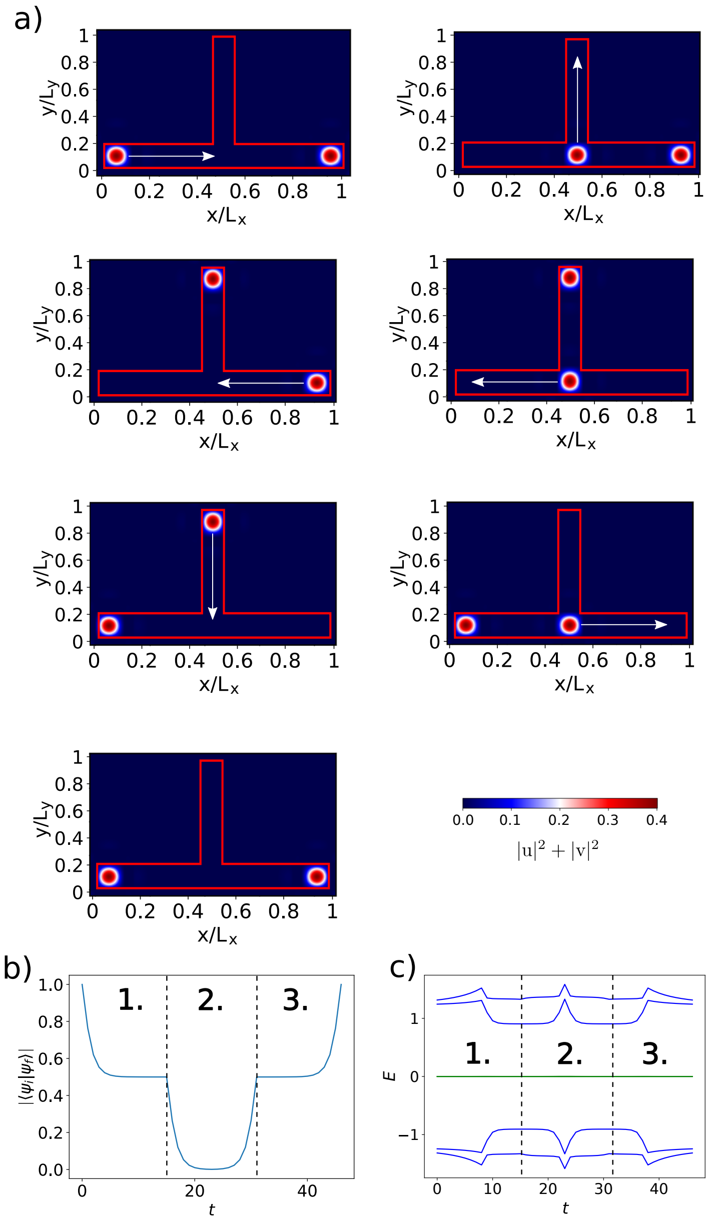

Figure 7.6.: a) Numerical braiding of two Majorana modes with $L_x = 19$ and $L_y = 11$. b) Absolute overlap of the initial electronic wavefunction with the electronic wavefunction after the adiabatic step t. c) Energies of the modes around zero during the exchange of the Majorana modes. The color bar is showing the probability density of the Majorana mode.

where $n_+ = 0, 1$ is a qubit formed by (γ_2 and γ_3), while $n_- = 0, 1$ is a qubit formed by (γ_1 and γ_4). However, it is more convenient to define the exchange operators in the basis $|n_1 n_2\rangle$ where the first qubit n_1 is formed by (γ_1 and γ_2), while the second qubit

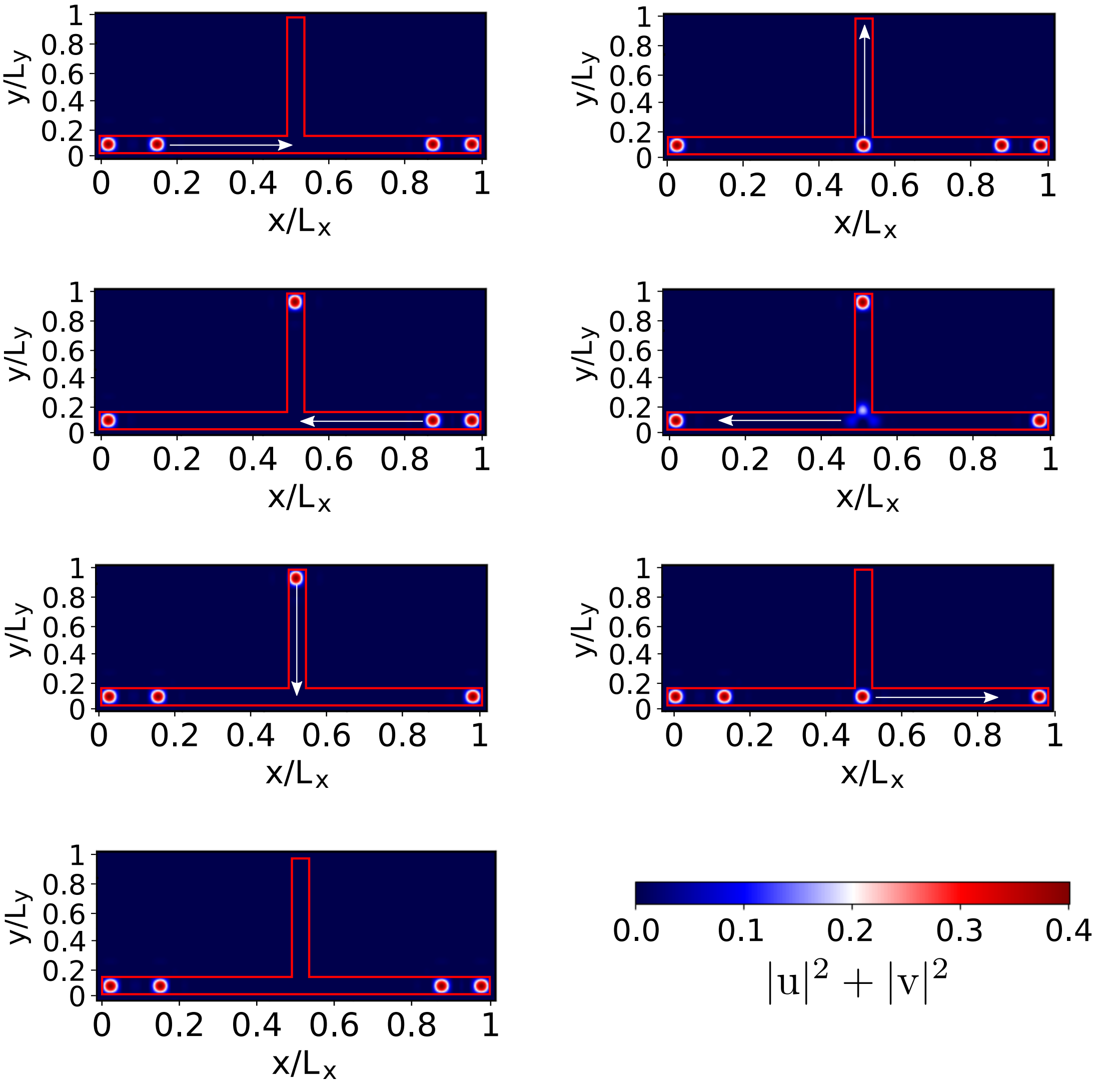

Figure 7.7.: Numerical braiding of four Majoranas with $L_x = 40$ and $L_y = 15$. When the Majorana mode is in the middle of the bottom wire it forms a triangle mode since it is changing from one parity sector to the other parity sector.

is formed by (γ_3 and γ_4). These two bases can be related by the F-matrix we discussed with the Ising Anyons given by [114]

$$\begin{pmatrix} |00\} \\ |11\} \\ |01\} \\ |10\} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} |00\rangle \\ |11\rangle \\ |01\rangle \\ |10\rangle \end{pmatrix}. \tag{7.37}$$

The exchange operator R_{23} in the basis $|n_+n_-\}$ is given by as in Equation (7.28)

$$R_{23}\begin{pmatrix}|00\}\\|11\}\end{pmatrix} = e^{-i\frac{\pi}{4}\sigma_z}\begin{pmatrix}|00\}\\|11\}\end{pmatrix} = \begin{pmatrix}e^{-i\frac{\pi}{4}} & 0\\0 & e^{i\frac{\pi}{4}}\end{pmatrix}\begin{pmatrix}|00\}\\|11\}\end{pmatrix} \tag{7.38}$$

$$R_{23}\begin{pmatrix}|01\}\\|10\}\end{pmatrix} = e^{-i\frac{\pi}{4}\sigma_z}\begin{pmatrix}|01\}\\|10\}\end{pmatrix} = \begin{pmatrix}e^{-i\frac{\pi}{4}} & 0\\0 & e^{i\frac{\pi}{4}}\end{pmatrix}\begin{pmatrix}|01\}\\|10\}\end{pmatrix}. \tag{7.39}$$

With these three equations, we can then also write the exchange operator U_{23} in the basis $|n_1n_2\rangle$

$$R_{23}\begin{pmatrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 0 & -i\\0 & 1 & -i & 0\\0 & -i & 1 & 0\\-i & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{pmatrix}, \tag{7.40}$$

which relates to the exchange operator in a four Majorana system shown in [29].

To summarize, our work focused on the intersection of superconductors and topological quantum computation, aiming to understand the essential components required for implementing topological quantum operations. We started by investigating the Kitaev chain toy model, a minimal system that undergoes a phase transition from a trivial superconductor to a topological superconductor, leading to the emergence of Majorana zero modes at its ends. These Majorana zero modes were then identified as Ising Anyons, crucial for implementing quantum gates through their braiding statistics. We demonstrated that even in this one-dimensional Kitaev chain model, the braiding of Majorana zero modes can be achieved by varying the superconducting phase. Our calculations verified that the braiding statistics of the Majorana zero modes align with those of the Ising Anyons.

A key part of this chapter was the development of an algorithm to eliminate the numerical phases associated with zero energy modes in the calculations. This algorithm allowed us to accurately and reliably analyze the properties and behaviour of the Majorana zero modes without being influenced by numerical artefacts. By implementing this algorithm, we ensured that our results were robust and precise, enhancing the credibility of our findings. With this novel approach, we were able to address the challenges posed by numerical phases and obtain meaningful and accurate results in our study of topological quantum computation with Majorana zero modes.

To provide a more comprehensive exploration, we extended our study to a two-dimensional system involving four Majorana zero modes, formed by connecting two Kitaev chains at a T-junction structure. Through this complex setup, we examined the braiding statistics of four Majorana zero modes. While our previous models were minimal and challenging to implement experimentally, our aim is to build a framework that enables the discussion of systems that are more feasible to study in real-world experiments.

Part III.

Towards Quantum Computation with Superconductor-Magnet Heterostructures

8

Superconductor-Magnet Heterostructure Model

In the previous chapter, we explored that p-wave superconductivity and Majorana zero modes are relatively rare in materials. To overcome this limitation, researchers have proposed combining different materials to create an effective p-wave superconductor, thus creating an experimental system that hosts Majorana zero modes. Kane and Fu were among the first to predict that topological insulators could achieve $p_x \pm ip_y$-wave-like pairing with an s-wave superconductor, leading to new theoretical ideas for realizing Majorana modes in physical systems. Some experimental studies have been conducted on a few systems, revealing features that hint at the braiding of these quasiparticles, yet the conclusive "smoking gun" experiment, demonstrating the braiding of two Majoranas and their non-abelian statistics, remains elusive.

In our work, we study composite systems of thin ferromagnet layers and thin superconducting layers. Such systems are of special interest since they are in a superconducting and ferromagnetic state at the same time in the region where the superconductor is in contact with the magnet. This region can be driven into a phase where it is an effective p-wave superconductor and can therefore host Majorana zero modes. In the following, we show that superconducting vortices, which are able to bind the Majorana zero modes, can be moved by domain walls in the magnet. Furthermore, we show that the braiding of Majorana zero modes in two-dimensional real space can be done with Skyrmion-Vortex pairs [31, 115].

8.1. Hamiltonian of a Superconductor-Magnet Heterostructure

To study such a superconductor-magnet heterostructure we refine the Hamiltonian in Equation (6.3):

$$\mathcal{H} = \int d\vec{r} \left[\psi^\dagger_{\alpha r} \left(\frac{\hat{p}^2}{2m} - \mu \delta_{\alpha\beta} + h^{so,i}_{\alpha\beta} \hat{p}_i + h^{ex}_{\alpha\beta} \right) \psi_{\beta r} + (\mathit{\Delta}\ \psi_{\downarrow r} \psi_{\uparrow r} + h.c.) \right]. \tag{8.1}$$

Here, ψ_α is the real-space creation operator for a spin α, μ is the chemical potential, and $\mathit{\Delta}$ is the pairing strength of the electrons in the superconductor. The first term of the Hamiltonian represents the kinetic energy of the electrons in the superconductor, the second term represents the spin-orbit coupling, the third term represents the proximity coupling of the magnet with the superconductor and the last term represents the pairing of electrons in an s-wave superconductor. In the following, we will discuss the different terms in more detail.

8.1.1. Spin-Orbit Coupling at Interfaces

Considering a heterostructure where the spin-orbit coupling arises from the breaking of inversion symmetry at the interface, we focus on Rashba spin-orbit coupling. In accordance with [116], we summarize the origin and important properties of spin-orbit coupling. This effect arises from the motion of an electron in an electric field, leading to an effective magnetic field $\boldsymbol{B}_{\text{eff}} \sim \boldsymbol{E} \times \boldsymbol{p}/mc^2$, which then couples to the electron spin as $\mu_B\, \boldsymbol{\sigma} \cdot \boldsymbol{B}$. This results in the equation

$$h^{so,i}_{\alpha\beta}\hat{p}_i \sim \mu_B/mc^2(\boldsymbol{\sigma} \times \boldsymbol{E}) \cdot \boldsymbol{p}. \tag{8.2}$$

The corresponding electric field arises in materials where the structural inversion symmetry is broken along the growth direction. This is a fact that also results from the time-reversal symmetry of the spin-orbit coupling. With the interfacial electric field $\boldsymbol{E} = E_z$ this results in a spin-orbit coupling of the form

$$h^{so,i}_{\alpha\beta}\hat{p}_i = -i\hbar\alpha_{\text{R}}(\boldsymbol{\sigma} \times \boldsymbol{z}) \cdot \boldsymbol{p}. \tag{8.3}$$

where α_{R} is known as the Rashba parameter. Another important property of the Rashba spin-orbit coupling is that it locks spin to the linear momentum and split the sub-bands in energy as shown in Figure 8.1.

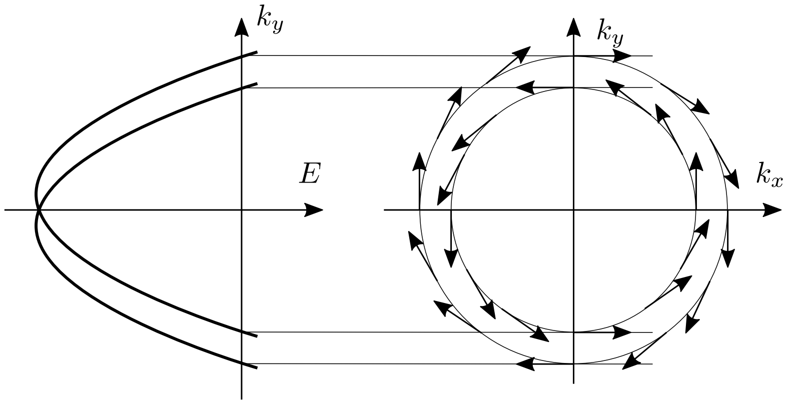

Figure 8.1.: Schematic picture that shows how the spin-orbit coupling splits the energy subbands. The arrows in the right panel represent the spin of the electrons with the corresponding momentum in the x- and y-directions. Since the spin has to be orthogonal to the momentum the picture shows the consequences of the spin locking.

8.1.2. Proximity Effect

When two materials are in close contact the properties of their interfaces change. In the case of a superconductor-metal interface, the Cooper pairs can penetrate the normal metal up to some distance. This means that superconducting-like properties may be induced in the normal metal. This effect is called proximity effect [27]. Additionally, the leakage of Cooper pairs weakens the superconductivity near the interface, which is then called the inverse proximity effect [27].

If we now replace the normal metal with a ferromagnet this effect gets dependent on the spin of the electrons. A Cooper pair consists of two electrons with opposite spins and momenta. In the ferromagnetic case, the spin-up electron, defined as having the orientation of the exchange field, experiences a decrease in energy, while the spin-down electron experiences an increase by the same value. This means that at the interface between the superconductor and the ferromagnet the exchange field couples to the spin of the electron which can be described by the term

$$h^{ex}_{\alpha\beta} = \mathbf{h} \cdot \boldsymbol{\sigma}. \tag{8.4}$$

It is important to note here that this term is only valid at the interface between the superconductor and the magnet.

8.2. Majorana Zero Modes in Superconductor-Magnet Heterostructures

Moving forward, our objective is to examine the phases where topological superconductivity emerges. To accomplish this, we will review relevant aspects of the work by Alicea et al. [22]. We commence this analysis by considering the one-dimensional Hamiltonian described in Equation(8.1).

$$\mathcal{H} = \int dx \left[\psi^{\dagger}_{\alpha x} \left(-\frac{\hbar^2 \partial_x}{2m} - \mu - i\hbar\alpha_{\mathrm{R}}(\hat{\boldsymbol{e}} \cdot \boldsymbol{\sigma})\partial_x - h_z\sigma_z \right) \psi_{\beta x} + \left(|\Delta| e^{i\phi} \psi_{\downarrow x}\psi_{\uparrow x} + h.c. \right) \right]. \tag{8.5}$$

The magnetic field here points in the z-direction and the vector $\boldsymbol{e}$ lies in the x-y plane. The topological phase can be reached by going into a phase of the system where the condition $|h_z| \gg m\alpha_{\mathrm{R}}^2$ is fulfilled. This condition ensures that the bands are separated far enough so that the spins are nearly polarized and we can map the equation to a one-band problem by writing $\psi_{\uparrow x} \sim (\alpha_{\mathrm{R}}(e_y + ie_x)/|h_z|)\partial_x \boldsymbol{\Psi}_x$ and $\psi_{\downarrow x} \sim \boldsymbol{\Psi}_x$, with $\boldsymbol{\Psi}_x$ the lower-band Fermion operator. We then have a spinless system as we have it in the Kitaev chain. To leading order, one obtains

$$\mathcal{H}_{\mathrm{eff}} = \int dx \left[\boldsymbol{\Psi}^{\dagger}_x \left(-\frac{\hbar^2 \partial_x}{2m} - \mu_{\mathrm{eff}} \right) \boldsymbol{\Psi}_x + |\Delta_{\mathrm{eff}}| e^{i\phi_{\mathrm{eff}}} \boldsymbol{\Psi}_x \partial_x \boldsymbol{\Psi}_x + h.c. \right] \tag{8.6}$$

where $\mu_{\mathrm{eff}} = \mu + |h_z|$ and the effective p-wave pairing is

$$|\Delta_{\mathrm{eff}}| e^{i\phi_{\mathrm{eff}}} \approx \frac{\alpha_{\mathrm{R}}|\Delta|}{|h_z|} e^{i\phi}(e_y + ie_x). \tag{8.7}$$

In the low-density limit, the effective Hamiltonian represents the Kitaev chain. The presence of effective p-wave pairing is evident from the combination of spin-orbit coupling, magnetic field, and s-wave pairing terms. This intriguing combination gives rise to the same physics as observed in the Kitaev chain and strongly suggests the existence of Majorana zero modes in such systems.

8.3. Numerical Implementation

Next, we will discuss how to discretize the Hamiltonian in Equation (8.1). For that, we use again the central differential approximation from Equation (7.32). If we combine the spin-orbit coupling, the proximity coupling and the kinetic term we get

$$\overleftrightarrow{h} = -\frac{\hbar}{2m}(\partial_x^2 + \partial_y^2) - i\hbar\alpha_{\mathrm{R}}(\sigma_x\partial_y - \sigma_y\partial_x) + \mathbf{h}\cdot\boldsymbol{\sigma} \tag{8.8}$$

and using the central differential approximation this results in

$$\begin{aligned} \overleftrightarrow{h}\vec{u} &= V_{i,j}\vec{u}_{i,j} + t_x\vec{u}_{i+a,j} + t_x^\dagger\vec{u}_{i-a,j} + t_y\vec{u}_{i,j+a} + t_y^\dagger\vec{u}_{i,j-a} \\ \varDelta\overleftrightarrow{\rho}\vec{u} &= i\varDelta_{i,j}\sigma_y\vec{u}_{i,j} \end{aligned} \tag{8.9}$$

where i, j indicate spatial site in x- and y-direction respectively and $\vec{u}_{i,j}$ are the spinors at each spatial site. A few new terms are introduced:

$$V_{i,j} = (\frac{2\hbar^2}{ma^2} - \mu)\mathbb{1} + \mathbf{h}_{i,j}\cdot\boldsymbol{\sigma}. \tag{8.10}$$

Here $\mathbb{1}$ is the identity matrix and μ is the chemical potential. $V_{i,j}$ are called the *onsite terms*. The *hopping parameters* in the x- and y-direction are given by

$$\begin{aligned} t_x &= -\frac{\hbar^2}{2ma^2}\mathbb{1} + i\frac{\hbar\alpha_{\mathrm{R}}}{2a}\sigma_y \\ t_y &= -\frac{\hbar^2}{2ma^2}\mathbb{1} - i\frac{\hbar\alpha_{\mathrm{R}}}{2a}\sigma_x. \end{aligned} \tag{8.11}$$

Here we see, that the spin-orbit coupling manifests as an additional term to the hopping parameter. From these terms, we finally obtain the desired Hamiltonian written in terms of the creation operators $\mathbf{c}_i^\dagger = (c_{i\uparrow}^\dagger, c_{i\downarrow}^\dagger)$ that create an electron a site i with spin ↑ or ↓. Rescaling these equations gives us

$$\begin{aligned} V_{i,j}^* &= -\mu^*\mathbb{1} + \mathbf{h}_{i,j}^*\cdot\boldsymbol{\sigma} \\ t_x^* &= -\mathbb{1} + i\alpha_{\mathrm{R}}^*\sigma_y \\ t_y^* &= -\mathbb{1} - i\alpha_{\mathrm{R}}^*\sigma_x, \end{aligned} \tag{8.12}$$

where $\mu^* := \frac{\mu}{t}$, $\mathbf{h}_{i,j}^* := \frac{1}{t}\mathbf{h}_{i,j}$, and $\alpha_{\mathrm{R}}^* := \frac{\alpha_{\mathrm{R}}ma}{\hbar}$ with m being the electron mass. Putting everything together results in the Hamiltonian

$$\begin{aligned} H = &- t\sum_{\langle i,j\rangle}\boldsymbol{c}_i^\dagger\boldsymbol{c}_j - \mu\sum_i\boldsymbol{c}_i^\dagger\boldsymbol{c}_i - \sum_i\boldsymbol{c}_i^\dagger(\boldsymbol{h}_i\cdot\boldsymbol{\sigma})\boldsymbol{c}_i \\ &+ \mathrm{i}\,\alpha_{\mathrm{R}}\sum_{\langle i,j\rangle}\boldsymbol{c}_i^\dagger\hat{\boldsymbol{z}}\cdot(\hat{\boldsymbol{d}}_{ij}\times\boldsymbol{\sigma})\boldsymbol{c}_j + \sum_i(\varDelta_i c_{i\uparrow}^\dagger c_{i\downarrow}^\dagger + h.c.)\,. \end{aligned} \tag{8.13}$$

At site i of the square lattice, on the x-y plane and with lattice constant a, itinerant electrons with spin α are generated (annihilated) by the operator $c_{i\alpha}^\dagger(c_{i\alpha})$, and $\boldsymbol{c}_i = (c_{i\uparrow}, c_{i\downarrow})^T$. The symbol $\langle i, j\rangle$ indicates a summation over nearest-neighbour lattice sites. Here, t is the hopping parameter, μ is the chemical potential, and $\boldsymbol{h}_i$ is the exchange field resulting from the magnetic texture in the proximity-coupled ferromagnet. Rashba spin-orbit coupling enters through the fourth term, where α_{R} is the coupling strength, $\hat{\boldsymbol{d}}_{ij}$ is the unit vector oriented along the nearest-neighbour link from j to i, $\boldsymbol{\sigma}$ is the vector of Pauli matrices, and i is the imaginary unit coefficient. The complex-valued and spatially dependent superconducting order parameter is $\varDelta_i$.

8.3.1. Self-Consistency Condition of the Pairing Potential

Having derived the Hamiltonian in equation (8.1) using mean-field theory, we need to take the self-consistency condition into account, as shown in (6.4). Upon inserting the Bogoliubov transformation (6.6) into the definition of the pairing potential, we observe its dependence on the eigenvalues and eigenvectors of the Hamiltonian:

$$\begin{aligned}\mathit{\Delta}(\mathbf{r}) &= \frac{V}{2}\langle c_\uparrow(\mathbf{r})c_\downarrow(\mathbf{r})\rangle - \langle c_\downarrow(\mathbf{r})c_\uparrow(\mathbf{r})\rangle \\ &= \frac{V}{2}\sum_n (v^*_{n\uparrow}(\mathbf{r})u_{n\downarrow}(\mathbf{r}) - v^*_{n\downarrow}(\mathbf{r})u_{n\uparrow}(\mathbf{r}))2n_F(\epsilon_n) + (u_{n\uparrow}(\mathbf{r})v^*_{n\downarrow}(\mathbf{r}) - u_{n\downarrow}(\mathbf{r})v^*_{n\uparrow}(\mathbf{r})).\end{aligned} \tag{8.14}$$

Here we used that $\langle\gamma_n^\dagger\gamma_n\rangle = 1 - \langle\gamma_n\gamma_n^\dagger\rangle = n_F(\epsilon_n)$, where $n_F(\epsilon_n)$ is the Fermi-Dirac distribution and ϵ_n is an eigenvalue of the system. Therefore, we have to solve the Hamiltonian and the Pairing potential within an iterative procedure. A flow chart of this procedure is shown in Figure 8.2. Here, we start by guessing a pairing potential. Subsequently, we proceed to compute the eigenvalues and eigenvectors of the resulting Hamiltonian. Utilizing these solutions, we can then evaluate the updated pairing potential using Equation (8.14). The newly obtained pairing potential is inserted into the Hamiltonian matrix. This process loops until the accumulated absolute value of the old and new $\mathit{\Delta}_{i,j}$ differs by less than a value of $\xi = 10^{-4}$. The calculation of the eigenvectors and eigenvalues is done on a graphic card. While we implemented the same functions also for the CPU, the GPU speeds up our calculations by a factor of three. Additionally, we implemented a monitoring function, allowing us to carefully track the convergence of the iterative procedure and troubleshoot any potential issues that may arise during the relaxation process.

8.3.2. Relaxation of Superconducting Vortices

In the following, we will show the relaxation of superconducting vortices using the algorithm above. For a superconducting Vortex centred at the origin we evaluate the following continuous ansatz

$$\mathit{\Delta}_i^{\text{Vortex}} = \mathit{\Delta}_0(r_i)e^{im\theta_i+\theta_0}, \tag{8.15}$$

at the lattice sites i with spatial coordinates characterized by radius r_i and azimuthal angle θ_i. Here, the angle θ_0 (chirality), as well as $m = \pm 1$ (vorticity), define the type of the Vortex, e.g. $\theta_0 = 0$ and $m = 1$ yields a standard outward pointing Vortex. We choose the magnet to be a ferromagnet. To facilitate the distinction between different chiralities, we devised a color wheel, as depicted in Figure 8.3. To identify and represent the various types of vortices, we compute the curl and divergence of the superconducting order parameter and map them to specific colors according to the color wheel.

This results in an inward pointing Vortex having a red colour in the centre, an outward pointing Vortex having a blue colour, an anticlockwise Vortex having a purple colour and a clockwise Vortex having a green colour in the centre as shown in Figure 8.4 a). If we choose the vorticity to be negative we can also introduce an Antivortex in the system as shown in Figure 8.4 b). In our numerical simulations, we introduce multiple vortices by

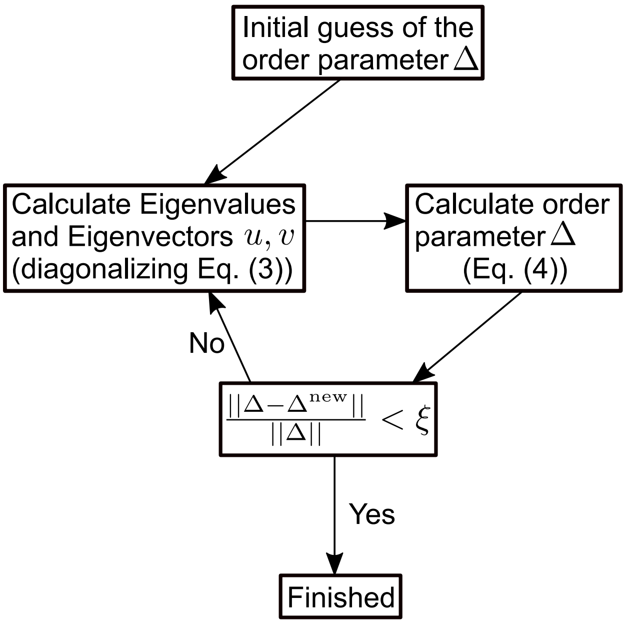
Initial guess of the order parameter Δ
Calculate Eigenvalues and Eigenvectors u, v (diagonalizing Eq. (3))
Calculate order parameter Δ (Eq. (4))
$\frac{||\Delta-\Delta^{\rm new}||}{||\Delta||} < \xi$
No
Yes
Finished

Figure 8.2.: Flow chart illustrating the calculation algorithm for the pairing potential using a diagonalization method. A convergence threshold $\xi = 10^{-4}$ is chosen in the simulation [117].

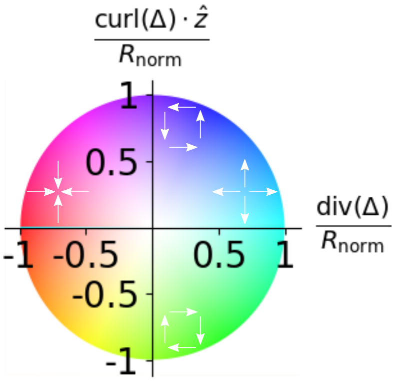
$\frac{\mathrm{curl}(\Delta)\cdot\hat{z}}{R_{\rm norm}}$
1
0.5
-0.5
-1
-1
-0.5
0.5
1
$\frac{\mathrm{div}(\Delta)}{R_{\rm norm}}$

Figure 8.3.: We show the colour wheel used in the following chapters to assign different colours to the chiralities of the vortices. The white arrows sketch what the Vortex of the corresponding colour looks like. To distinguish the chiralities we calculate the curl and the divergence of the superconducting order parameter.

convolving equation (8.15) with a Gaussian function, effectively localizing each vortex. By adding two vortices with opposite chirality (as shown in Figure 8.4 c), we observe the presence of two vortices in the sample, with the superconducting order parameter connecting them. However, when two vortices of the same chirality are added (as shown in Figure 8.4 d), an antivortex forms in the middle to compensate for the discontinuity in the superconducting phase between the vortices.

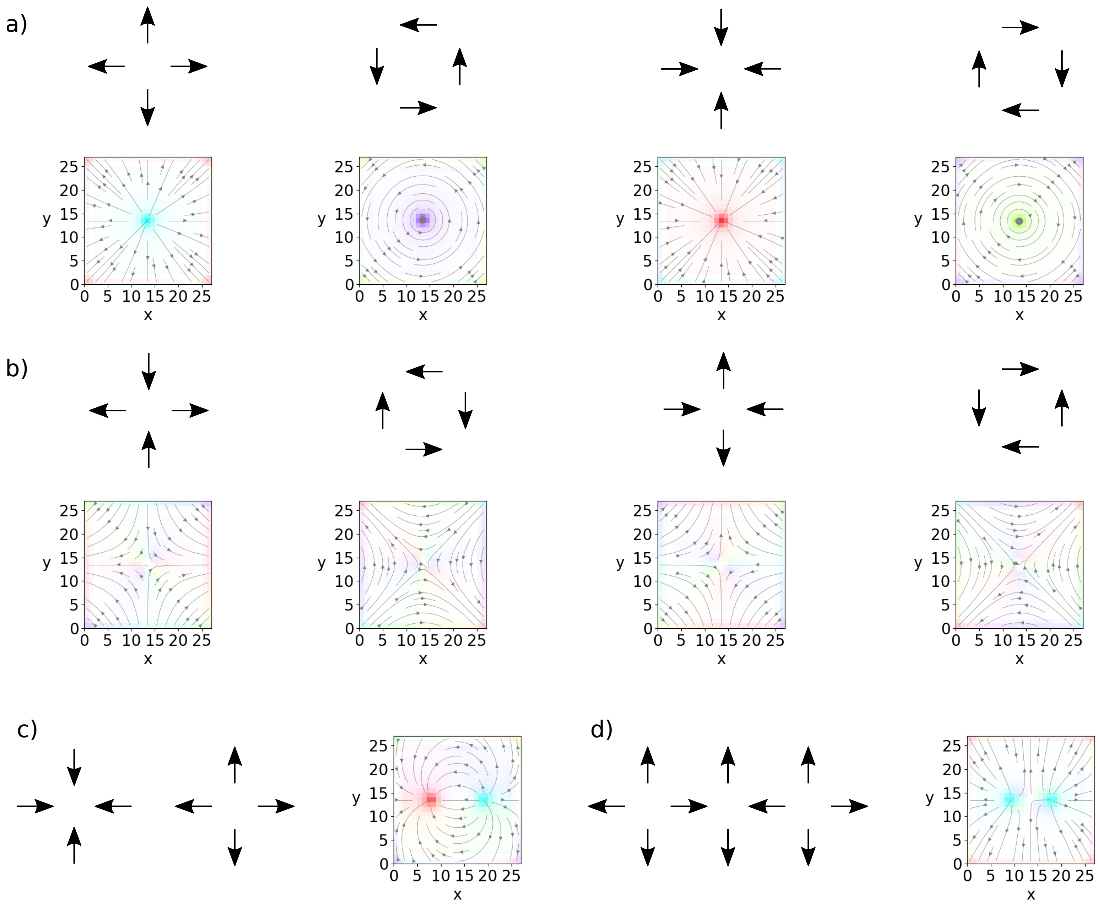

Figure 8.4.: Here we show a depiction of the superconducting order parameter with arrows and the corresponding pictures from the simulation. In panel a) we show vortices of different chiralities $\sigma_0 = 0, \pi/2, \pi, 3\pi/2$ (from left to right), in panel b) we show antivortices of different chiralities $\sigma_0 = 0, \pi/2, \pi, 3\pi/2$ (from left to right), in panel c) there are two vortices having the same chirality and in panel d) two vortices having different chiralities. Note that for panel d) there forms an additional Antivortex between the two vortices.

9

Domain Wall Vortex Interaction

In this chapter, we want to explore how magnetic domain walls and vortices interact and how these interactions can be used. In nature, there are many examples of how interactions of different media can lead to ordered media. For example, the RNA polymerase glides along a DNA strand during the transcription process [118, 119]. In superconductor-ferromagnet heterostructure, one employs different order parameters to achieve ordering [120, 27, 121]. By using a thin insulating spacer between the superconducting and the magnetic layers to avoid the mutual suppression of superconductivity and magnetism, magnetic domain walls have been shown to interact with superconducting vortices via stray fields [122, 123]. In our approach, we use spin-orbit coupling to achieve the coexistence of superconductivity and ferromagnetism [124, 125] and is known to enhance proximity-induced superconductivity [126, 127]. To combine the different order parameters we will drive the superconductor-ferromagnet heterostructure in a phase where domain walls are present in the magnet and vortices are present in the superconductor as shown in Figure 9.1. The interactions between the two order parameters then show their nature when moving the domain wall towards the vortices. In the figure, we depicted that the domain wall pushes the vortices. Depending on the domain wall helicity and the Vortex vorticity the vortices can also be pulled by the domain wall or even glide along the domain wall. The rich interplay between magnetic domain walls and

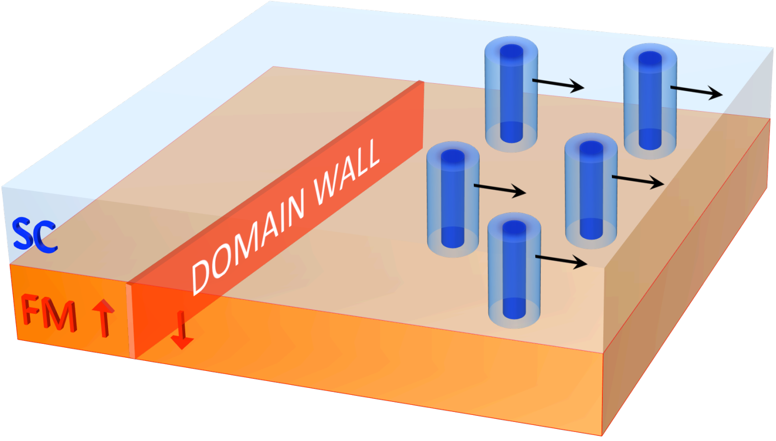

Figure 9.1.: Superconducting vortices manipulated by magnetic domain walls in superconductor-ferromagnet heterostructures. Due to the magnetoelectric effect induced by spin-orbit coupling, a magnetic domain wall in a ferromagnetic thin film (FM) that is close to a superconducting layer (SC) interacts with superconducting vortices (blue tubes). The domain wall can push –as seen above–, pull, or trap vortices depending on the helicity of the domain wall and the vorticity of the Vortex.

superconducting vortices in proximity-coupled superconductor-ferromagnet heterostructures induced by Rashba spin-orbit coupling is highlighted in the following.

9.1. Magnetic Domain Wall - Superconducting Vortex Interactions

To study the interaction between a superconducting Vortex and a magnetic domain wall we have to embed their profiles in the Hamiltonian shown in Equation (8.13) we described before. The exchange field induced by the magnetic domain wall, aligned parallel to the y-axis and centred at $x = 0$, is given by

$$\mathbf{h}^{(\mathrm{dw})} = h_0[\sin \Theta(x)(\cos\gamma\,\hat{\boldsymbol{x}} + \sin\gamma\,\hat{\boldsymbol{y}}) + \cos\Theta(x)\hat{\boldsymbol{z}}], \tag{9.1}$$

where h_0 is the exchange field strength and γ is the domain wall *helicity*. We choose $\cos\Theta(x) = -\tanh(x/d)$, which satisfies $\Theta(0) = \pi/2$, $\Theta(-\infty) \to 0$, and $\Theta(+\infty) \to \pi$, i.e., with the magnetic moments pointing asymptotically upward (downward) to the left (right) of the domain wall. Here, d characterizes the domain wall width and we use $d = 3a$ in our numerical calculations. An example of a Néel domain wall is shown in Figure 9.2. The superconducting Vortex is defined as in Equation (8.15) and we use $q_v = 1$ in the following.

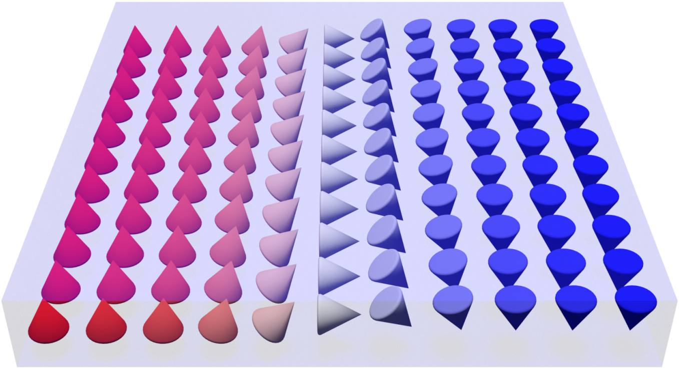

Figure 9.2.: Depiction of a Néel domain wall.

The numerical calculation is done by the following procedure. We place a single superconducting Vortex sufficiently far to the right of the domain wall (at a distance of 19 lattice sites) where the magnetic texture is effectively field polarized. We choose the profile of this initial Vortex such that it is close to an actual Vortex profile. Then we calculate the superconducting order parameter, with this initial Vortex as a seed, self-consistently according to the algorithm shown in Figure 8.2. The vorticity of the Vortex core, characterized by the winding number of θ_s, along with the chirality of the background exchange field, plays a crucial role in determining the resulting Vortex profile at its location and the strength of the Rashba spin-orbit coupling α_{R}. We then move the location of the domain wall one lattice site to the right and repeat the self-consistent calculation with the Vortex configuration from the previous step as the seed. We simulate four distinct helicity cases: Néel domain walls with $\gamma = 0, \pi$ and Bloch domain walls with $\gamma = \pi/2, 3\pi/2$. Implementing a rightward domain wall motion yields strikingly different superconducting Vortex responses.

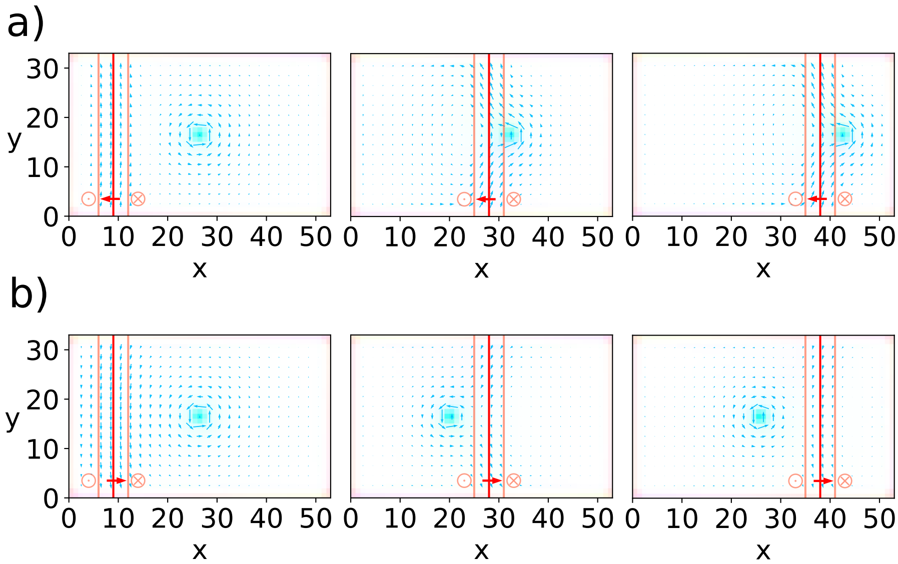

Figure 9.3.: Repulsive and attractive interaction between a Néel magnetic domain wall and a superconducting Vortex. The blue arrows show the superconducting current, the red line corresponds to the centre of the domain wall with the red arrow showing the chirality. The orange lines depict the domain wall width with the magnetisation direction. The initial Vortex-domain wall separation amounts to $15a$. a) Néel domain wall with $\gamma = \pi$ and b) Néel domain wall with $\gamma = 0$.

A Néel domain wall with $\gamma = \pi$, when sufficiently close, constantly pushes the superconducting Vortex to the right as shown in Figure 9.3 a). However, if the Néel domain wall has helicity $\gamma = 0$, the behaviour is almost the opposite. Initially, the superconducting Vortex is attracted towards the domain wall until it crosses to the left side of the domain wall where it finds a sweet spot and it follows the motion of the domain wall at a constant distance. This looks like the superconducting Vortex is being dragged by the domain wall (Figure 9.3 c)).

For a Bloch domain wall, the superconducting Vortex is attracted to the domain wall for both chiralities. When the Vortex comes closer to the domain wall it gets deflected depending on the chirality and glides along the domain wall. For $\gamma = 3\pi/2$ as shown in Figure 9.4 a) it glides along the $+\hat{\boldsymbol{y}}$-direction and for $\gamma = \pi/2$ it glides in the $-\hat{\boldsymbol{y}}$-direction (Figure 9.4 c)).

The numerical results described above can be understood using the well-established Ginzburg-Landau theory of superconductors, for which Kjetil Hals performed the analytical calculations. The spatially asymmetric Rashba spin-orbit coupling (SOC) introduces magnetoelectric coupling mechanisms between the superconducting condensate and the spin system. These interactions are incorporated into the Ginzburg-Landau phenomenology through the following magnetoelectric free energy functional

$$F_{\text{me}}\left[\psi^*, \psi, \mathbf{A}, \mathbf{h}\right] = \int \mathrm{d}\mathbf{r} \left[\mathcal{F}^{(1)}_{\text{me}} + \mathcal{F}^{(2)}_{\text{me}}\right], \tag{9.2}$$

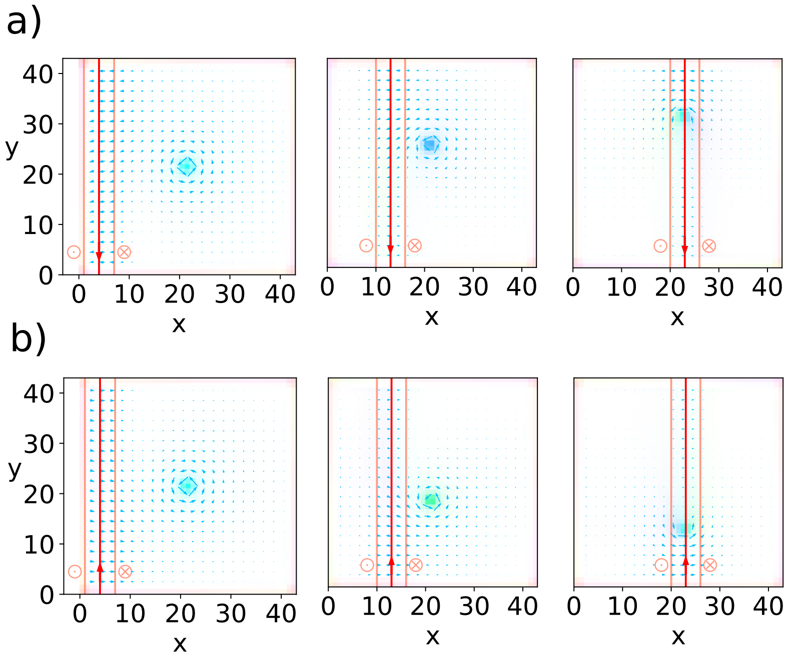

Figure 9.4.: Attractive interaction and gliding motion of a superconducting Vortex along a Bloch magnetic domain wall. The initial Vortex-domain wall separation amounts to $15a$ a) Néel domain wall with $\gamma = 3\pi/2$ and b) Néel domain wall with $\gamma = \pi/2$.

where the two free energy densities $\mathcal{F}_{\mathrm{me}}^{(1)}$ and $\mathcal{F}_{\mathrm{me}}^{(2)}$ are SOC-induced Lifshitz invariants given by

$$\mathcal{F}_{\mathrm{me}}^{(1)} = \kappa^{(1)}(\hat{\mathbf{z}} \times \mathbf{h}) \cdot \boldsymbol{\mathcal{P}}, \tag{9.3}$$

$$\mathcal{F}_{\mathrm{me}}^{(2)} = \kappa^{(2)}(\hat{\mathbf{z}} \times \boldsymbol{\nabla} h_z) \cdot \boldsymbol{\mathcal{P}}. \tag{9.4}$$

Here, $\mathbf{h}$ is the exchange field induced in the superconductor by the adjacent ferromagnetic layer via the exchange proximity effect, $\boldsymbol{\mathcal{P}} = \mathrm{Re}\,[\psi^*(-i\hbar\boldsymbol{\nabla} - 2e\mathbf{A}/c)\psi]$ is the momentum density of the superconducting condensate, $\psi(\mathbf{r}) = |\psi(\mathbf{r})| \exp(i\phi(\mathbf{r}))$ is the complex order parameter field of the superconductor, 2e is the charge of a Cooper pair, c is the speed of light, and $\mathbf{A}$ is the magnetic vector potential, which is connected to the magnetic induction $\mathbf{B}$ via the relationship $\mathbf{B} = \boldsymbol{\nabla} \times \mathbf{A}$. In the subsequent discussion, we concentrate on the case of a thin film type-II superconductor where the penetration depth is significantly larger than the typical length scales that characterize the spatial variations of magnetic domain walls and superconducting vortices. In this case, we can disregard the magnetic vector potential in Equation (9.2). The constant $\kappa^{(1)}$ ($\kappa^{(2)}$) parametrizes the magnetoelectric coupling and is first (second) order in the SOC.

From the magnetoelectric free energy functional one can now calculate the emergence of an anomalous supercurrent density $\mathbf{j}_{as}$ which builds next to the conventional supercurrent density $\mathbf{j}_{cs}$ the net supercurrent density $\mathbf{j}_s = \mathbf{j}_{cs} + \mathbf{j}_{as}$. The specific form of these supercurrents is calculated in Appendix B.2,A and results . From these supercurrents, one can build a simple physical understanding:

When a heterostructure consisting of a superconductor and a ferromagnet contains both a domain wall and a Vortex, the superconductor aims to minimize its free energy by displacing the Vortex to reduce the overall supercurrent density $\mathbf{j}_s = \mathbf{j}_{cs} + \mathbf{j}_{as}$ to a minimum. Consequently, the Vortex experiences a magnetoelectric force $\mathbf{f}_{\mathrm{me}}$ that influences its equilibrium position and dynamics. In the case of a Néel domain wall, it is anticipated that the Vortex will settle at an equilibrium position approximately $|R_x| \sim \xi$ away from the centre of the wall. This arrangement ensures that the region with the highest current intensity around the Vortex core partially counteracts the significant anomalous supercurrent density generated by the magnetoelectric interactions $\mathcal{F}^{(1)}_{\mathrm{me}}$ and $\mathcal{F}^{(2)}_{\mathrm{me}}$ along the wall. Conversely, for Bloch domain walls, the effects of $\mathcal{F}^{(1)}_{\mathrm{me}}$ and $\mathcal{F}^{(2)}_{\mathrm{me}}$ differ. The influence of $\mathcal{F}^{(2)}_{\mathrm{me}}$ is not dependent on $\hat{\mathbf{n}}^{(\mathrm{dw})}$ and consequently results in the same equilibrium position for the Vortex as observed in the case of a Néel domain wall. Conversely, the $\mathcal{F}^{(1)}_{\mathrm{me}}$ term is expected to pin the Vortex to one of the edges of the Bloch wall (depending on the helicity of the spin texture) in order to counteract the supercurrent induced by the $\kappa^{(1)}$-term in the interior of the sample.

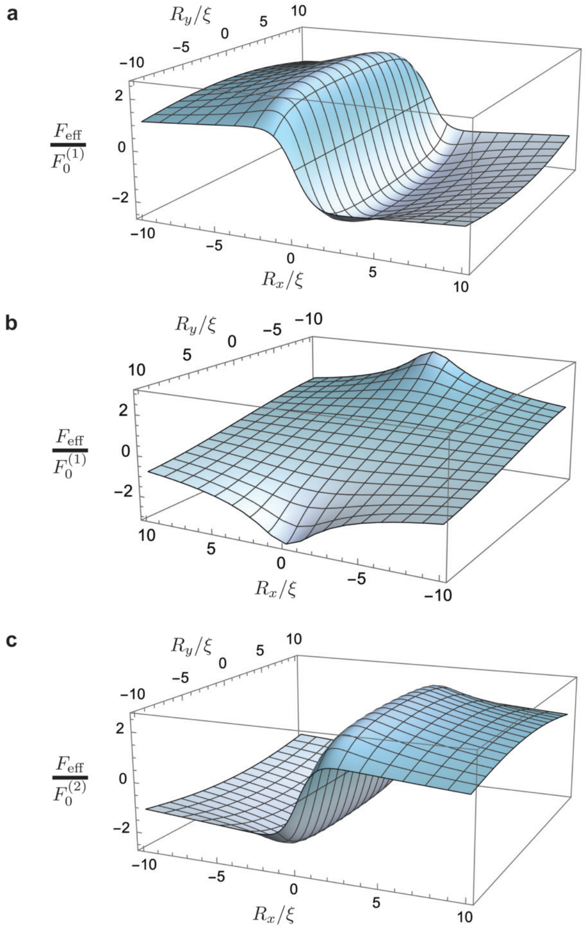

Figure 9.5.: Numerical calculations of the effective free energy F_{eff} in Equation (9.5) under the following conditions: **a.** F_{eff} in systems containing a Néel wall, where $\kappa^{(2)} = 0$ and $\gamma = 0$. **b.** F_{eff} in systems containing a Bloch wall, where $\kappa^{(2)} = 0$ and $\gamma = \pi/2$. **c.** F_{eff} when $\kappa^{(1)} = 0$. In this case, the magnetoelectric free energy is independent of the domain wall structure. In all figures, the length scales are measured in units of ξ, and we have used that $\lambda/\xi = 0.4$. The sample size considered is $[-\infty, +\infty] \times [-L_y, L_y]$, where $L_y/\xi = 10$. The free energy in cases **a** and **b** (case **c**) is measured in units of $F_0^{(1)} = \pi\hbar\lambda\kappa^{(1)}h_0\psi_0^2$ ($F_0^{(2)} = 2\hbar\kappa^{(2)}h_0\psi_0^2$).

The magnetoelectric force $\mathbf{f}_{\mathrm{me}}$ is determined by the effective free energy $F_{\mathrm{eff}}(\mathbf{R})$, given by

$$F_{\mathrm{eff}}(\mathbf{R}) = F_{\mathrm{me}}[\psi^{(v)*}, \psi^{(v)}, \mathbf{h}^{(\mathrm{dw})}], \tag{9.5}$$

This free energy is obtained by substituting the exchange field $\mathbf{h}^{(\mathrm{dw})}$ and Vortex profile $\psi^{(v)}(\mathbf{r}) = \psi_0 \tanh\left(\frac{|\mathbf{r}-\mathbf{R}|}{\xi}\right) \exp(i\phi(\mathbf{r}))$ into Equation (9.2), and then integrating over the spatial coordinate $\mathbf{r}$. The expression F_{eff} depends solely on the collective coordinate $\mathbf{R}$ of the Vortex and determines how the free energy of the superconductor changes as the Vortex's position varies relative to a domain wall centred at $x = 0$.

Using Equation (9.5), we can derive the force acting on the Vortex as

$$\mathbf{f}_{\mathrm{me}} = -\frac{\partial F_{\mathrm{eff}}}{\partial \mathbf{R}}, \tag{9.6}$$

This force drives the Vortex towards the energy minimum of the effective free energy described in Equation (9.5). Figure 9.5(a) (Figure 9.5(b)) illustrates a numerical computation of the effective free energy F_{eff} in the vicinity of a Néel wall (Bloch wall). In order to highlight the influence of the domain wall profile, we have set $\kappa^{(2)} = 0$ in Figures 9.5(a) and 9.5(b), focusing solely on the impact of the $\mathcal{F}^{(1)}_{\mathrm{me}}$ term. Figure 9.5(c) presents the helicity-independent component generated by $\mathcal{F}^{(2)}_{\mathrm{me}}$. As expected, for Néel walls, the helicity-dependent part of F_{eff} displays a global minimum at a distance $|R_x| \sim \xi$ from the centre of the wall (Figure 1(a)). In contrast, Bloch walls exhibit a minimum at one of the edges of the wall at the boundary of the sample (Figure 1(b)).

In Appendix B.2,B we derive functional form of the effective magnetoelectric free energy (9.5) for vortices located near Néel and Bloch walls, which are consistent with the numerical investigations shown in Figure 9.5:

$$\begin{aligned} F^{(nw)}_{\mathrm{eff}} &= (\epsilon F^{(1)}_0 - F^{(2)}_0)\left[c_2 \tilde{R}^3_x - c_1 \tilde{R}_x + c_3 \tilde{R}_x \tilde{R}^2_y\right] \\ F^{(bw)}_{\mathrm{eff}} &= -\epsilon c_4 F^{(1)}_0 \tilde{R}_y - F^{(2)}_0 \left[c_2 \tilde{R}^3_x - c_1 \tilde{R}_x + c_3 \tilde{R}_x \tilde{R}^2_y\right]. \end{aligned} \tag{9.7}$$

In the case of Néel walls, the series expansions of the terms $F^{(1)}_0$ and $F^{(2)}_0$ are equivalent, resulting in an energy minimum at $\mathbf{R}_0 = \mathrm{sign}(\epsilon F^{(1)}_0 - F^{(2)}_0)[\xi\sqrt{c_1/3c_2}, 0, 0]$. In contrast, for heterostructures containing Bloch walls, the term $F^{(1)}_0$ exhibits an energy minimum at $R_y = \mathrm{sign}(\epsilon F^{(1)}_0) L_y$. It is important to note that this phenomenon is solely influenced by finite-size effects originating from the c_4 term in $F\mathrm{eff}^{(bw)}$, which diminishes as L_y/ξ approaches infinity. On the other hand, the term $F^{(2)}_0$ still possesses a minimum at $\mathbf{R}_0 = -\mathrm{sign}(F^{(2)}_0)[\xi\sqrt{c_1/3c_2}, 0, 0]$.

The magnetoelectric force $\mathbf{f}_{\mathrm{me}}$ associated with the free energies in Equation (9.7) can be obtained from Equation (9.6). This force can be incorporated into a description of the Vortex's collective motion using the equation of motion (see Appendix B.2,C):

$$m_v \ddot{\mathbf{R}} = \dot{\mathbf{R}} \times \mathbf{G} + \mathbf{f}_{\mathrm{me}} - \alpha \dot{\mathbf{R}}. \tag{9.8}$$

In this equation, m_v represents the effective mass of the Vortex, α parameterizes the friction, and $\mathbf{G} = 2\pi n_s q_v \hat{\mathbf{z}}$ determines the Magnus force arising from the Vortex's topologically nontrivial structure. Here, n_s corresponds to the density of Cooper pairs, while q_v denotes the vorticity, which, in our specific case, is equal to 1.

The dynamics of the Vortex are governed by the interplay between the Magnus force and the magnetoelectric force described in Equation (9.8). Our numerical simulations confirm the presence of the force originating from the $F_0^{(1)}$ term in Equation (9.7), assuming that $F_0^{(1)}$ is negative. In the case of a Néel wall with $\gamma = 0$, the $F_0^{(1)}$ term in Equation (9.7) exhibits a minimum at $\mathbf{R} = [-\xi\sqrt{c_1/3c_2}, 0, 0]$. This explains why the Vortex crosses into the left domain and lags behind the moving wall. Conversely, for a Néel wall with $\gamma = \pi$, the Vortex reaches an equilibrium position at $\mathbf{R} = [\xi\sqrt{c_1/3c_2}, 0, 0]$ and is pushed in front of the domain wall as it moves to the right.

In the case of Bloch walls, the $F_0^{(1)}$ term has a minimum at the edge of the wall, specifically at $R_y = L_y$ ($R_y = -L_y$) for $\gamma = 3\pi/2$ ($\gamma = \pi/2$). This energy landscape induces a motion of the Vortex along the y-direction, explaining its gliding movement along the wall. Regardless of the specific structure of the domain wall, the Magnus force introduces a slight deflection in the motion of the Vortex.

To evaluate the relevance of the $\kappa^{(2)}$ term (i.e., $F_0^{(2)}$ in Equation (9.7)), we conducted a numerical analysis comparing the magnitudes of $j_{as,x}^{(\mathrm{dw})}$ and $j_{as,y}^{(\mathrm{dw})}$ in Equation (A-12) for a system featuring a Bloch wall (without a Vortex). In this scenario, the $\kappa^{(1)}$ term generates a current along the x-axis, while the $\kappa^{(2)}$ term produces a current parallel to the y-axis. Consequently, the ratio $j_{as,y}^{(\mathrm{dw})}/j_{as,x}^{(\mathrm{dw})}$ serves as a direct indicator of $\kappa^{(2)}/\kappa^{(1)}$. Our numerical investigation reveals that the value of $j_{as,y}^{(\mathrm{dw})}/j_{as,x}^{(\mathrm{dw})}$ is significantly lower than the machine precision, suggesting that the contribution from $\kappa^{(2)}$ is negligible.

9.2. Manipulating Superconducting Vortices with Magnetic Domain Walls

By taking advantage of the propulsion experienced by superconducting vortices due to the $\gamma = \pi$ Néel magnetic domain wall, enhanced functionality can be gained. For example, as shown in Figure 9.6, a single domain wall can be used to push away and thereby remove superconducting vortices from a specific spot in the sample. Alternatively, the same procedure can be used to enhance the density of vortices in another sample area or to fill an otherwise empty region with vortices. Furthermore, this method shows that magnetic domain walls have the ability to prevent or induce superconducting vortex avalanches [128, 129, 130].

In Figure 9.7, two domain walls are used to assemble a chain of superconducting vortices, illustrating another potential application. Vortex chains have previously been described in anisotropic [131, 132, 133] and nanopatterned superconductors [134]. Because of their mobility, magnetic domain walls are an appealing non-invasive way to build reconfigurable superconducting Vortex channels.

Apart from manipulating superconducting vortices with magnetic domain walls, they can also be manipulated with different magnetic textures. In the next chapter, we discuss the interaction between magnetic Skyrmions and superconducting vortices. In comparison to the interaction with domain walls, the superconducting Vortex is not pushed by the domain wall but they stick to the Skyrmion enabling a more controlled movement of the superconducting Vortex. For example, a braiding motion of the superconducting

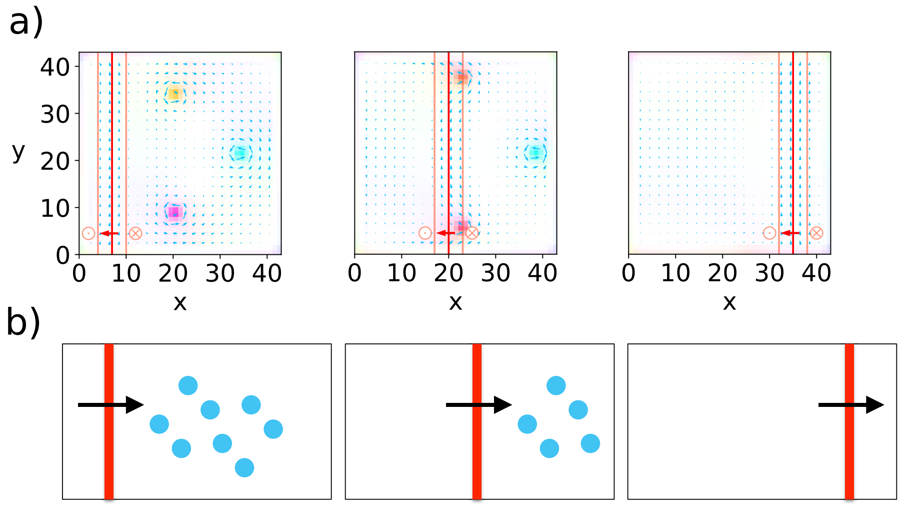

Figure 9.6.: Wiping the slate clean of superconducting vortices. Sequential simulation snapshots of a rightward-moving Néel magnetic domain wall with helicity $\gamma = \pi$ that pushes vortices away from the sample area, effectively acting as a Vortex "eraser" or "rake". The initial Vortex-domain wall separation amounts to $12a$.

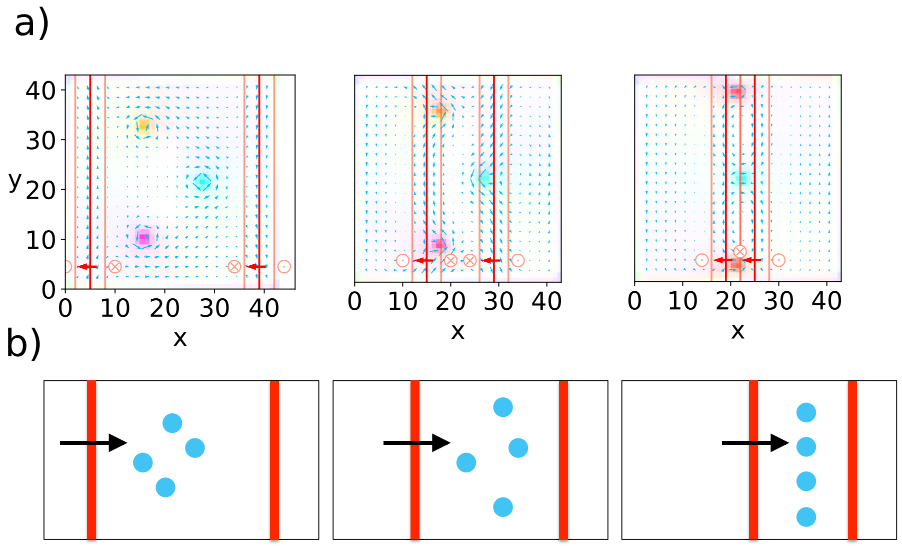

Figure 9.7.: Assembling a superconducting Vortex chain using two magnetic domain walls. Sequential simulation snapshots of superconducting vortices confined between two Néel magnetic domain wall with helicity $\gamma = \pi$. As the separation between the domain walls decreases the superconducting vortices rearrange and form a chain parallel to the domain walls. The initial Vortex-domain wall separation amounts to $7a$ to the left and right domain walls.

vortices can be implemented which opens the path to topological quantum computation with superconducting vortices.

10 Skyrmion-Vortex Pair Braiding

Until now there are only a few experiments that reported features alluding to the finding of Majorana zero modes and similar quasiparticles [135, 136, 137], but the "smoking gun" experiment, i.e. the braiding of two Majorana zero modes revealing their non-Abelian statistics, is still missing.

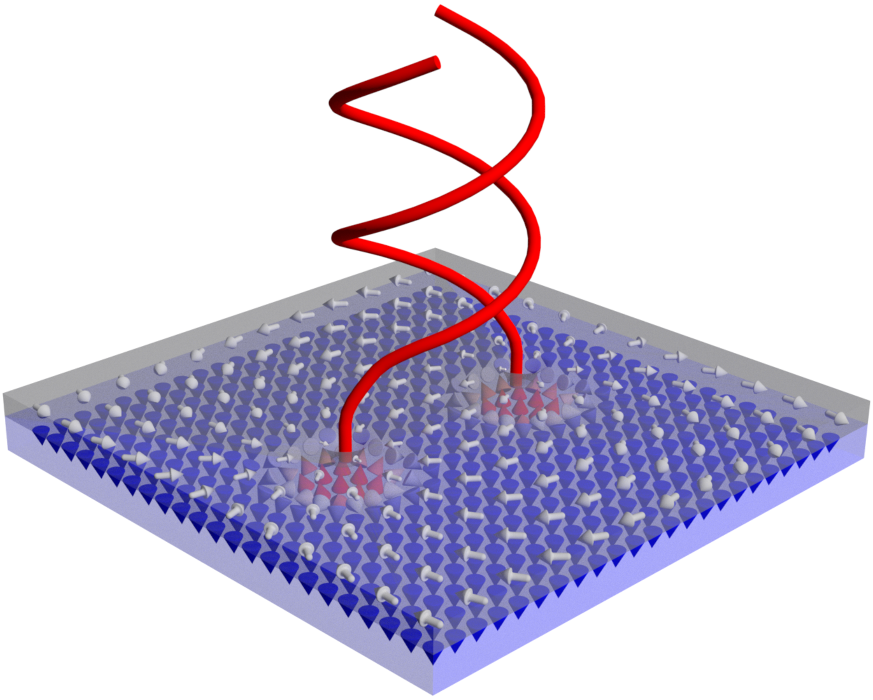

Figure 10.1.: Sketch of the braiding of two Skyrmion-Vortex pairs in a superconductor-ferromagnet heterostructure. The white arrows represent the superconducting order parameter and the magnetic structure is depicted by the cones underneath. Upward (downwards) pointing magnetic moments are plotted in red (blue). The red tubes indicate the world lines that emerge when braiding Majorana modes. Picture published in [117].

We show, that Majorana modes can be braided in a two-dimensional system consisting of a ferromagnet with Skyrmions proximity coupled to an s-wave superconductor with vortices [31, 115]. We consider a superconductor-ferromagnet heterostructure modelling a 2d s-wave superconductor by the two-dimensional tight binding Hamiltonian in the Bogoliubov-de Gennes formalism [113, 138]. We adiabatically braid the hybrid topological structures and explicitly confirm the non-Abelian statistics of the Majorana zero modes they carry. For that, we numerically solve for the superconducting order parameter self consistently. Together with a proper structuring of these materials, we propose a scalable topological quantum computing platform [117, 30]. Additionally, this platform provides that the Majorana zero modes can be initialized, manipulated, and read out. A sketch of the braiding of two Skyrmion-Vortex pairs (SVPs) in a superconductor-ferromagnetic heterostructure is shown in Figure 10.1. Furthermore, we show that our system also provides techniques to fuse and create Majorana modes. In the end, we give an outlook on how to define a qubit out of the Majorana modes and initialize this qubit to a certain state.

10.1. Skyrmions and Vortices in a Superconductor-Magnet Heterostructure

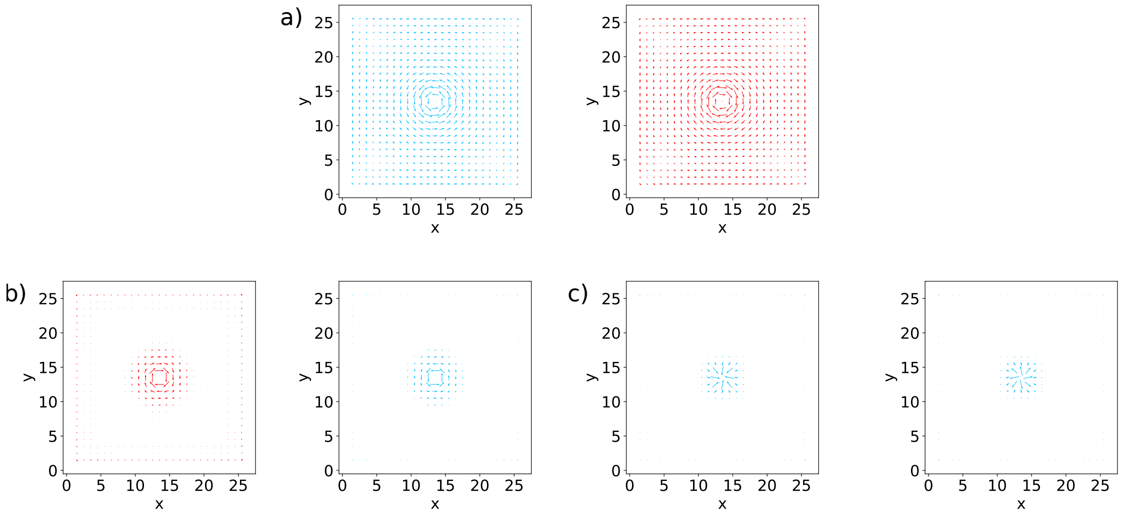

Figure 10.2.: Supercurrent of a) Vortex and Antivortex, b) Néel Skyrmions ($\phi = 0, \pi$) and c) Bloch Skyrmions ($\phi = \pi/2, 3\pi/2$). The colour depicts if the supercurrent is anticlockwise (blue) or clockwise (red)

In the following model, we study the properties of Skyrmion-Vortex pairs in superconductor-ferromagnetic heterostructures [113, 115]. To incorporate the Skyrmion in the Hamiltonian shown in (8.13) we describe it by an external magnetic field $\boldsymbol{h}_i$. The magnetic field for a Skyrmion centred at the origin will be represented by

$$\boldsymbol{h}(r, \phi) = h_0 \cdot (\cos \Phi \, \sin f(r), \sin \Phi \, \sin f(r), \cos f(r)), \tag{10.1}$$

with

$$\Phi = \Phi(\phi) = \mathcal{W}_{\text{sk}} \, \phi + \varphi \tag{10.2}$$

at the corresponding lattice site i with spatial coordinates characterized by radius r_i and azimuthal angle ϕ_i. Here, h_0 is the strength of the magnetization, $\mathcal{W}_{\text{sk}}$ is the Skyrmion winding number, and φ is the helicity of the Skyrmion. For the approximated profile function of the Skyrmion we choose one that is valid in thin films [139] and can be written as

$$f(r) = 4 \arctan(\exp(-r/\eta)), \tag{10.3}$$

where η is a parameter describing uniform compression or expansions of the profile. The superconducting Vortex is defined as in Equation (8.15).

Since the magnetic field $\boldsymbol{h}_i$ couples to the electrons in the superconductor, it induces a supercurrent. Depending on the supercurrent that is induced by the Skyrmion it can couple to a Vortex or Antivortex. When the supercurrent induced by the Skyrmion opposes the supercurrent of the Vortex or Antivortex, an attractive interaction occurs. This is because the supercurrent is minimized, leading to a reduction in the interaction term in the Hamiltonian. In Figure 10.2, we have plotted the supercurrent of a Vortex, an Antivortex, and various types of Skyrmions. Our findings reveal that the Néel Skyrmion with $\phi = 0$ exhibits an attractive interaction with the Vortex, while all other types of Skyrmions show an attractive interaction with the Antivortex. This is in agreement with the results found in [113].

In our study of Skyrmion-Vortex pairs, we focus on Néel Skyrmions, where $\varphi = 0$, as they exhibit an attractive interaction with superconducting vortices. Our main interest lies in investigating the Majorana zero modes hosted by these Skyrmion-Vortex pairs, which we will explore in the upcoming section.

10.2. Majorana Zero Modes at Skyrmion-Vortex Pairs

By calculating the energy spectrum of the system we reveal the regions where Majorana modes are present. Using the Bogoliubov de Gennes formalism we compute the eigenvalues ε_n and the eigenvectors of the excitations $u_{n,\beta}, v_{n,\alpha}$ of the Hamiltonian Equation (8.13) while self-consistently calculating the superconducting order parameter [138, 113]

$$\varDelta(v_{n,\alpha}, u_{n,\beta}) = i\frac{V}{2} \sum_{\substack{n>0 \\ 0<\varepsilon_n<\hbar\omega_D}} (\sigma_y)_{\alpha\beta} v^*_{n,\alpha} u_{n,\beta} \left[2n_F(\varepsilon_n) - 1\right]. \tag{10.4}$$

Under the influence of spin-orbit coupling and magnetic field V characterizes the effective attractive interaction strength between the Cooper pairs and $n_F(\varepsilon_n) = (\exp\left(\frac{\varepsilon_n}{k_B T}\right)+1)^{-1}$ is the Fermi-Dirac distribution. The superconducting state including the relaxed Vortex structures are encoded in $\varDelta(v_{n,\alpha}, u_{n,\beta})$. The energy summation is only carried out in a small shell around the Fermi energy, i.e. $0 < \varepsilon_n < \hbar\omega_D$, where ω_D is the Debye frequency.

Figure 10.3 shows the 20 lowest modes of the energy spectrum and the corresponding wavefunctions of the red-marked modes for a) one, b) two, c) three, d) four and e) six Skyrmion-Vortex pairs as a function of applied magnetic field strengths h_0. These red-marked modes experience a gap closing at $h_0^{\mathrm{gap}} \approx 0.55$ which is invariant of the number of Skyrmion-Vortex pairs. Since these modes are separated in energy from the other modes for field values closely above h_0^{gap}, it is an indication that we are in a topological regime and these modes are Majorana modes. Furthermore, they are localized at the Skyrmion-Vortex pairs as shown in Figure 10.3. For an even number of Skyrmion-Vortex pairs, we find that the Majorana zero modes are localized at the Skyrmion-Vortex pairs, whereas for an odd number of Skyrmion-Vortex pairs the Majoranas localizes at the Skyrmion-Vortex pair and the other one extends over the boundary of the sample. We listed all system parameters that are not varying in Table 10.1 which were inspired by Refs. [113] and [138].

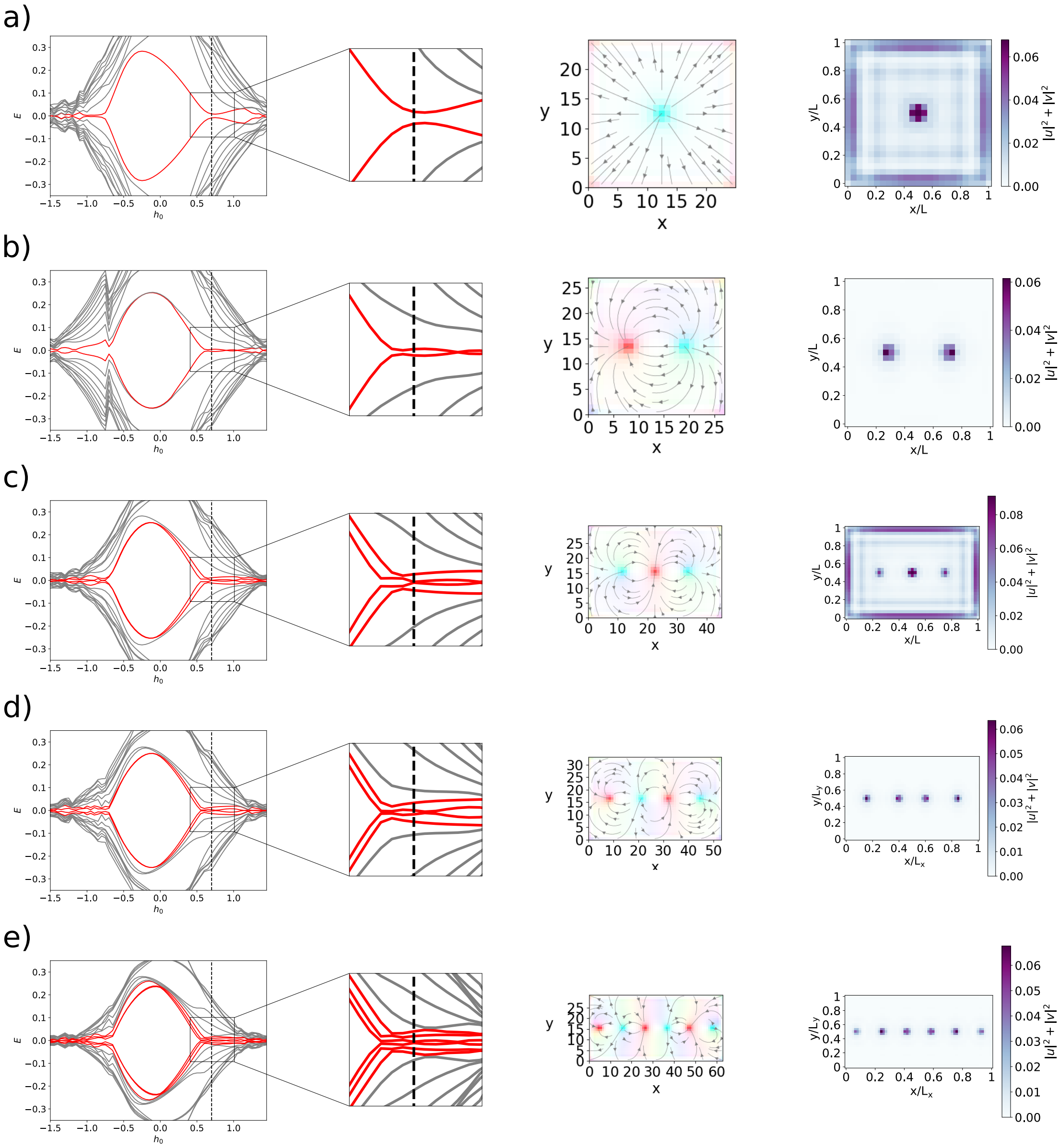

Figure 10.3.: Low energy spectrum and the corresponding wavefunctions of the red marked modes of a superconductor-ferromagnet heterostructure with a) one, b) two, c) three, d) four and e) six Skyrmion-Vortex Pairs as a function of magnetic field strength h_0. The lowest modes which could be Majorana modes after the gap closing are shown in red for better visibility. The horizontal line at $h_0 = 0.55$ is intended to clarify where the gap closing takes place. We show a zoom of the gap closing as well a Plot of the superconducting order parameter and a Plot of the Majorana modes (from left to right). The sample has dimensions of a) L=25a b) L=27a c) L_x=45a L_y=31a d) L_x=53a L_y=33a e)L_x=63a L_y=31a. To increase the visibility of the edge modes in a) and b) we used a scaling factor of a) 10, b) 20.

The topological region of the model of two Skyrmion-Vortex pairs can be seen in Figure 10.4a). We calculated the difference between the lowest energy mode E_1, shown in Figure 10.4b), and the next higher energy mode E_2, Figure 10.4c). To obtain Majorana zero mode the lowest energy mode has to be close to zero and there should be

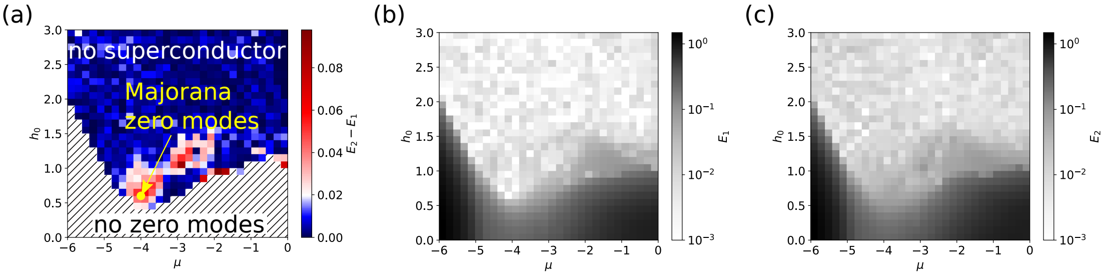

Figure 10.4.: In a) we show the energy difference of the two lowest energy modes. The colour determines how far the two energies are separated where red means they are separated and form a gap and blue means they are not separated. We cut out the part, where the lowest energy mode is not close to zero. For reference we plotted the lowest energy mode E_1 in b) and the second lowest energy mode E_2 in c). The conditions for energy modes to be Majorana modes are $E_1 \to 0$ and to be in a well-defined topological region with a finite superconducting gap $E_1 - E_0 > 0$. The yellow dot in a) marks such a region for a chemical potential of $\mu = -4$ and a field value of about $h_0 = 0.7$. These parameters we will use in further numerical calculations.

a gap to the next higher energy mode. Therefore we excluded the part of the plot in Figure 10.4a), where the lowest energy mode is far away from zero and the difference between the modes is then given by the colour code. With that, we find Majorana zero modes in the red regions of the Plot. We marked the optimal spot in the phase diagram with a yellow dot which corresponds to a chemical potential of $\mu = -4$ and a field value of about $h_0 = 0.7$. These parameters will be used in the following numerical calculations.

Parameters	Symbol	Value
Hopping parameter	t	1
Chemical potential	μ	-4
Spin-Orbit-Coupling	α_R	0.75
Thermal energy	k_BT	0.001
Debye frequency	ω_D	100
Effective attractive interaction	V	5

Table 10.1.: Chosen system parameters.

10.3. Braiding of Skyrmion-Vortex Pairs

Having a system with Majorana modes we will braid them and see if they obey the braiding statistics predicted by theory. To reach a motion of the Majorana modes we have to move the superconducting Vortex to which the Majorana mode is bound. With a Skyrmion present in the system, we have two mutually supportive mechanisms with which we can move the superconducting Vortex with the help of the Skyrmion. The first mechanism is based on magnetic stray fields. Superconducting vortices, being holes in the superconductor, enable the penetration of magnetic fields at their positions. In a superconductor-magnet heterostructure, the field lines of the Skyrmions in the magnet must encircle the magnet once, due to the distinct polarity of the Skyrmion core compared to the rest of the magnet. This interaction is most effective when the field lines can penetrate the superconductor. For this reason, it is energetically more favourable for Skyrmions and superconducting vortices to lie on top of each other. The second mechanism is based on the proximity coupling we discussed in this thesis. A Skyrmion induces a supercurrent in the superconductor [113], which in the case of a Néel induces a supercurrent which is opposite to the supercurrent of a superconducting Vortex. Since the system wants to minimise its supercurrent a Néel Skyrmion has an attractive interaction with a superconducting Vortex. The same is true for a Bloch Skyrmion and an Antivortex.

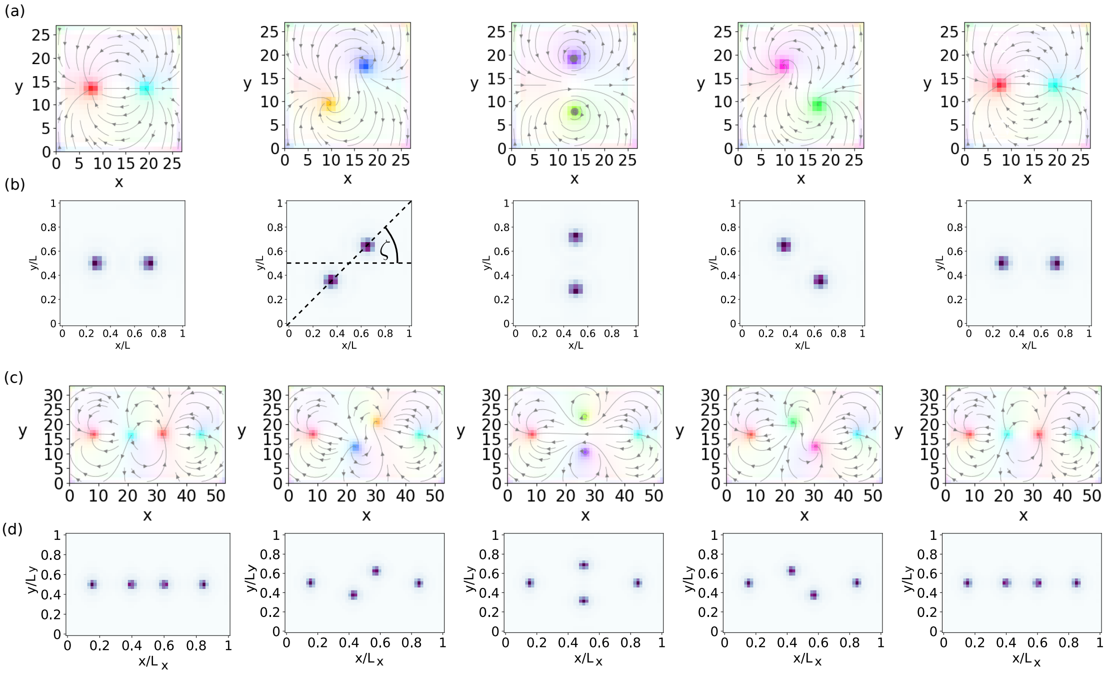

Figure 10.5.: Braiding of two Skyrmion-Vortex pairs. (a) Evolution of the superconducting order parameter revealing the braiding of the vortices. The colour code visualises the type of Vortex characterised by its divergence and curl as shown in Figure 8.3. (b) Spatial probability density of the wave functions corresponding to the Majorana zero modes during braiding (L=27a). With the angle ζ, shown in the second panel, we parameterise the braiding. (c) Braiding of two Skyrmion-Vortex pairs in a four Skyrmion-Vortex pair set-up. Here, the two Skyrmion-Vortex pairs in the middle are braided, leaving the outer ones unchanged. Colour codes are as in (a). (d) Spatial probability density of the wave functions corresponding to the Majorana zero modes (L_x=53a L_y=33a).

In the numerical calculation, we implement this motion by adiabatically changing the position of the Néel Skyrmions and numerically relaxing the system in each step according to Equations (8.13) and (10.4). By doing this the Vortex follows the motion as expected since it is energetically more favourable for the Vortex to be close to the Skyrmion. During this braiding motion, the vortices change their chirality. An inward (outward)-pointing Vortex (red and cyan) transforms to a right (left)-handed Vortex (green and purple) and then to an outward (inward)-pointing Vortex as shown in Figure 10.5 a) for two SVPs and c) for four SVPs. The braiding of the SVPs also results in the adiabatic motion of the Majorana zero mode bound to the SVPs whose evolution is shown in Figure 10.5 b) for two SVPs and in d) for four SVPs. While we exchange the

Majorana modes in this way they stay well localised at the superconducting vortices. To represent an infinitely large system in our numeric calculations, we fixed the phase of the superconducting order parameter at the sample boundary.

When we calculate the Braiding statistics with the concepts described in Section 7.4, where in this case we use ζ instead of ϕ we get

$$M_{12} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} , \quad M_{23} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} , \tag{10.5}$$

for two and four Skyrmion-Vortex pairs respectively by using our self-consistent calculation. As discussed in Section 7.5 this directly reveals the non-abelian statistics of the Majorana zero modes. We also tried to do this analysis for Skyrmion-Antivortex pairs; the results can be seen in Section B.3. However, further research is necessary to investigate the conditions under which Skyrmion-Antivortex pairs can host Majorana modes.

10.4. Creation and Fusing of Majoranas

In the realm of topological quantum computation, measurements can be performed by fusing two Anyons [91]. To achieve the fusion of Anyons bound to the Skyrmion-Vortex pairs, a possible approach is to pin one of the Skyrmions and move the other one into it. However, due to the typically lower coupling of the Skyrmions to the vortices compared to the repulsion between the vortices, this can lead to challenges, as the vortex may detach from the skyrmion before the Anyons fuse. To overcome this issue, an alternative strategy involves driving the Skyrmion-Vortex pair out of the sample, thereby localizing one Anyon on the edge of the sample. Subsequently, the second Skyrmion-Vortex pair can be driven out of the sample, leading to the fusion of the Anyons. This fusion process grants access to the logical information stored in the state of the system. In this context, we explore the creation and fusion of the Majorana zero modes, which are further discussed with the aid of the results presented in Figure 10.6.

In Figure 10.6, the upper panel displays the process of moving a Majorana mode out of the sample for the Skyrmion-Vortex pairs (a) and the wavefunctions of the Majorana zero modes (b). This involves moving the right Skyrmion out of the sample, along with the Vortex and the associated Majorana zero mode. As the Majorana mode approaches the edge, its localization weakens until it completely localizes on the edge. To enhance the visibility of the edge-localized mode, we increased the amplitude of its wavefunction. Subsequently, by moving the left Skyrmion to the left, the Majorana mode also shifts to the left, as depicted in the lower panel of Figure 10.6 (a) and (b). In this scenario, the Majorana mode located at the edge redistributes the probability of the electrons toward the left side, where the distance to the other Majorana is smaller. As the localized Majorana approaches the other, they prepare to fuse with each other. Ultimately, in the last image, the Majorana modes fuse, and the wavefunction of the lowest energy mode becomes delocalized, leading to a uniform value of the wavefunction throughout the entire sample.

By introducing an additional Skyrmion-Vortex pair (SVP), a new pair of Majorana zero modes emerges, as illustrated by reversing the time of the lower panel in Figure 10.6 b).

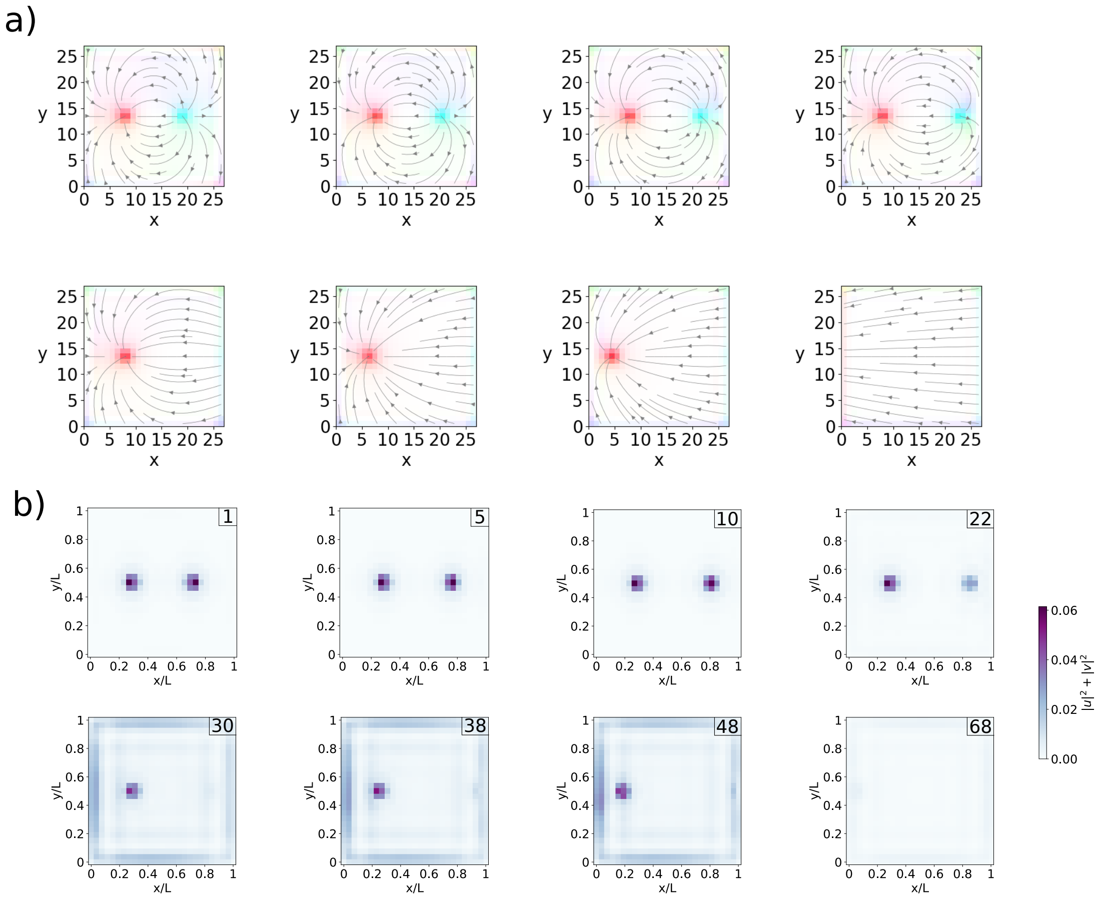

Figure 10.6.: Here, we show how fusing of two Majorana zero modes can be done by manipulating the superconducting vortices with Skyrmion-Vortex pairs. In a) we show the motion of the vortices during the fusing and in b) we show the corresponding wavefunction of the Majorana modes. First, we move the right Vortex out of the sample and after that, we move the left Vortex out of the sample. When the right Vortex is moved out it distributes over the edge of the sample. When the left Vortex is moved out the two Majoranas fuse on the edge leading to a completely extended state over the sample. For better visibility, we increased the amplitude of the wavefunction on the edge by a factor of 10.

When a new SVP is added, one of the Majorana modes localizes at the new SVP, while its companion extends over the edge of the sample. This indicates that the creation of a Skyrmion-Vortex pair from the edge can separate one Majorana from its counterpart.

To be sure, that the Majorana modes are really fused after this procedure we plotted the energies of the modes of the system in Figure 10.7. In the first part of the plot, the energies are really near zero. After one Majorana localises on the edge there is a jump in the higher energy modes but the low energy modes stay near zero. By moving the second Majorana to the edge the gap extends until it completely opens up when the two Majoranas fuse. After that, we do not have any Majorana in the system anymore.

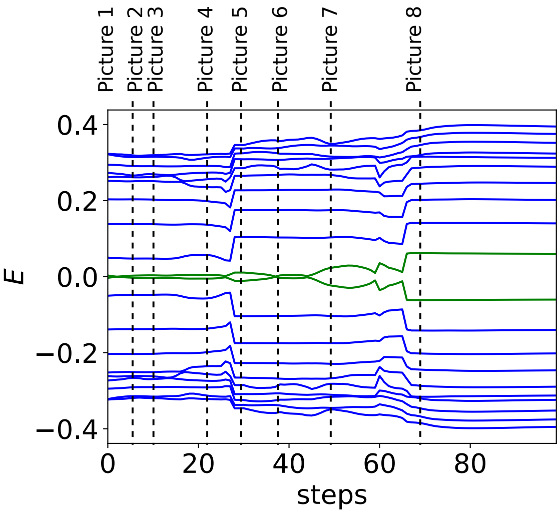

Figure 10.7.: In this figure the energy evolution of the modes shown in Figure 10.6 are shown. The Majorana zero modes are marked in green. While the Skyrmion-Vortex pairs, and therefore also the Majorana zero modes are pushed out of the sample, the energies change. After both Majorana zero modes are pushed out of the sample an energy gap forms.

10.5. Advancing Scalability in Topological Quantum Computing

In this section, we demonstrate the scalability of our platform. As previously shown, we can create multiple Majorana zero modes by increasing the number of Skyrmion-Vortex pairs (SVPs) in the sample, and we have also demonstrated the braiding of multiple Majorana zero modes using the SVPs. However, to achieve a scalable platform, we need a method for performing braiding of SVPs in a controlled and efficient manner. To address this, we propose a device structure, as depicted in Figure 10.8, consisting of a superconductor (grey), a magnet (blue), an insulating layer (white), and electric contacts. The magnet is patterned in a grid structure to create paths for the Skyrmions, allowing specific pairs of SVPs to be braided while leaving others unaffected. We present two architectures for this grid structure: one with a square pattern, which is simpler to fabricate, and another with a triangular pattern, allowing for a denser configuration of SVPs due to its diagonal paths.

The electric contacts are used to generate a controlled current flow, moving the Skyrmions along the magnetic paths. To perform the braiding, two electrodes are activated, sequentially moving one SVP to its target position and then the other SVP, leading to the desired movement of the two SVPs. To protect the Majorana modes in the superconductor from the current flow, we propose the addition of an insulating layer between the magnet and the superconductor. The thickness of this insulating layer should be chosen to ensure that the current flow in the superconductor is negligible, while still maintaining a significant exchange field coupling between Skyrmions and vortices. This is achievable because the exchange field and the current scale differently as a function of layer thickness d (exchange field $\sim d^2$ and current $\sim d^4$), allowing us to effectively isolate the Majorana modes while retaining the desired coupling between SVPs and vortices.

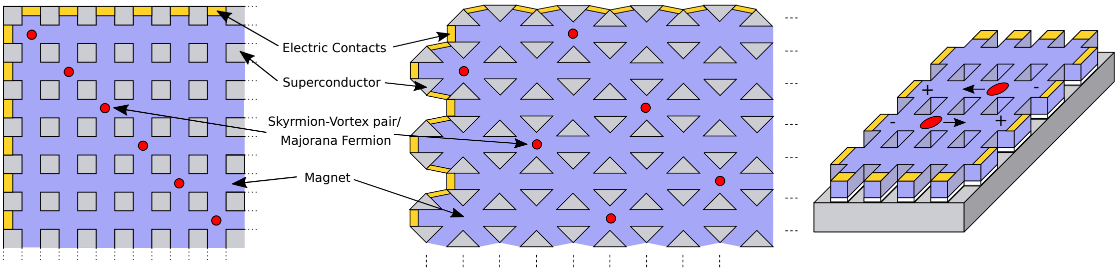

Figure 10.8.: Scalable topological quantum computing platforms based on Skyrmion-Vortex pairs. Left: top view of a square grid architecture. Middle: top view of a triangular grid architecture. Right: 3d heterostructure comprising a thin film superconductor (grey) and a square-grid-patterned magnet (blue), separated by an insulating layer (white), where electric contacts (yellow) are used to manipulate and braid SVPs (red). Adapted from Ref. [30].

10.6. Initialisation of a Qubit State and Quantum Gates

As discussed for the Ising Anyons in Section 5.3.2 we need at least three Anyons to define a qubit. To discuss the quantum gates for Skyrmion-Vortex pairs we will consider two qubits and therefore six Majorana zero modes denoted by σ. This leaves us with a computational basis of [96]

$$\begin{aligned} |00\rangle &= |\sigma\sigma;1\rangle\,|\sigma\sigma;1\rangle\,|\sigma\sigma;1\rangle\,, \\ |10\rangle &= |\sigma\sigma;\psi\rangle\,|\sigma\sigma;\psi\rangle\,|\sigma\sigma;1\rangle\,, \\ |01\rangle &= |\sigma\sigma;1\rangle\,|\sigma\sigma;\psi\rangle\,|\sigma\sigma;\psi\rangle\,, \\ |11\rangle &= |\sigma\sigma;\psi\rangle\,|\sigma\sigma;1\rangle\,|\sigma\sigma;\psi\rangle\,, \end{aligned} \tag{10.6}$$

Manipulations of the Majoranas by braiding shown before will perform quantum operations on this state [140].

These manipulations are called quantum gates and it is important to note here, that every quantum gate can be split up into some single-qubit gates and one two-qubit gate. The realisation of common quantum gates is shown in Figure 10.9. The Clifford gates can therefore be implemented for example by the Hadamard, the phase gate, and the CNOT gate. The Hadamard gate is represented by the following exchange sequence

$$R_{12}R_{23}R_{12} = R_{23}R_{12}R_{23} = e^{i\pi/4}H. \tag{10.7}$$

The phase gate can be easily implemented by

$$R_{12} = S. \tag{10.8}$$

The CNOT gate can be implemented by six Majorana zero modes [141]

$$CNOT = R_{34}^{-1}R_{45}R_{34}R_{12}R_{56}R_{45}R_{34}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{10.9}$$

Together with the $\frac{\pi}{8}$-gate this would form a universal set of quantum gates. However a $\frac{\pi}{8}$ gate can not be implemented with Majorana zero modes, which is a basic limitation

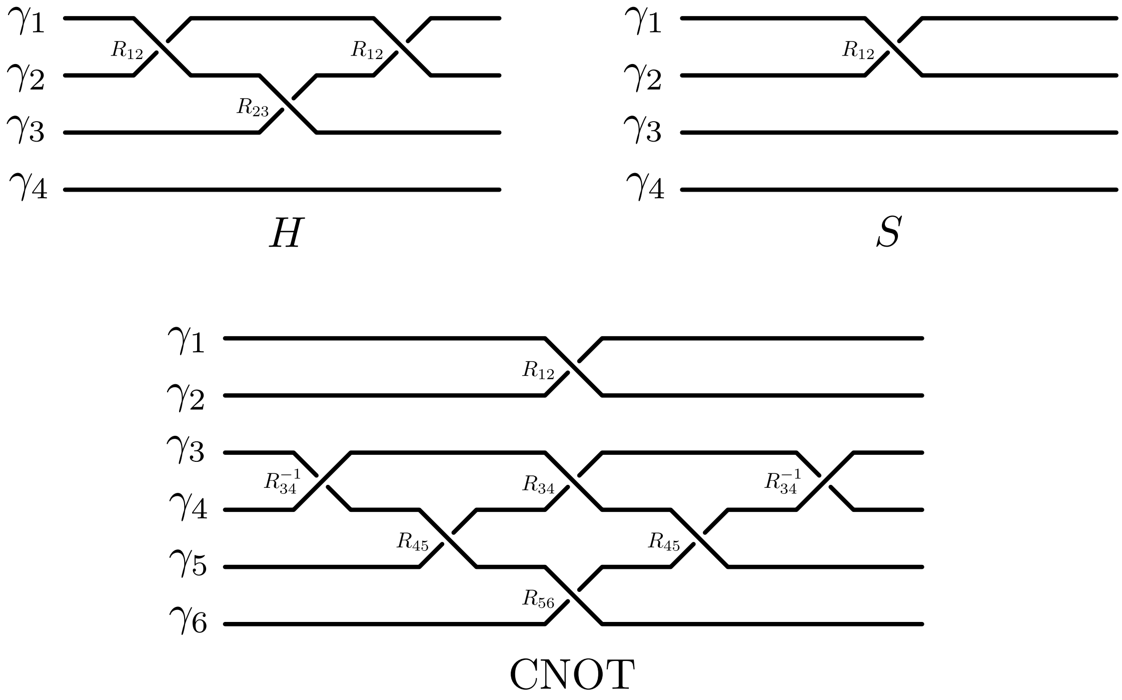

Figure 10.9.: Clifford quantum gates consisting out of the Hadamard gate `H`, the phase gate `S` and the `CNOT` gate. The lines correspond to the worldlines that the Majorana follows. An exchange is represented by two crossing lines, which can move counterclockwise or clockwise. We illustrate this with one line being solid whereas the other line is interrupted.

of Majorana zero modes. In Section 5.3.2 we already discussed how to overcome this shortcoming by using non-topological gates, where we bring two Majorana zero modes close to each other. There are even more ideas for universal sets of quantum gates discussed in [141] that circumvents measuring the qubits as suggested by [140]. Also, some of the gates stated in [141] have been carried out in [142] with a system consisting of a superconductor-semiconductor heterostructure providing superconducting vortices with Majorana-zero modes. Here the braiding is done by manipulating the position of the vortices with STM tips. This also makes our proposal experimental plausible.

10.7. Advancements in Quantum Computing Using Skyrmion-Vortex Pairs

In this chapter, we proposed a system which could be used as a quantum computing platform for Majorana zero modes. Since these topological qubits are less susceptible to fluctuations from the environment and then also less susceptible to errors this opens up the next step in quantum computing by reducing the amount of quantum error correction. Because of this logical qubits are made of fewer physical qubits and therefore making quantum computers less hardware-consuming. This simplifies the path to a robust quantum computer that is able to exceed the computation space of classical computers.

We discussed the details of the platform by presenting the model that we use consisting of a superconductor-ferromagnet heterostructure where we introduce Skyrmions in the magnet and vortices in the superconductor. Next, we analysed this model with respect to the existence of Majorana zero modes, where we found that such modes are present in systems with Skyrmion-Vortex Pairs. We also show a phase diagram for which parameters these modes are present. As a next step, we looked into the mechanism of braiding these Majorana modes by moving the Skyrmion and dragging the superconducting Vortex along. We also showed the Majorana modes have the statistics, which was proposed by Ivanov [29]. Furthermore, we discussed the possibility of creating and fusing Majorana modes in our quantum computing platform. Here, the Majorana modes can be created from the edge and also fused by successively driving the Majorana modes one after the other to the edge. A very important topic is the scalability of such a system. Here we proposed a specific setup which tackles this problem and makes it possible to control the movement of the Majorana zero modes by patterning the magnetic material in a grid to predefine paths that the Majorana modes can use and by using electrodes from the edge of the sample. Finally, we also discuss how a complete set of quantum gates could be realised in our system and how the qubit states could be initialised.

This last part is only a sketch of how the initialisation of the qubits can be carried out. It is not clear, that adding Skyrmion-Vortex pairs always gives us Majorana zero modes in the right state. One could use the leads at the edge of the sample and makes a conductive measurement on the Majorana modes. This projects the Majorana mode into the empty fermionic states resulting also in a projection of the qubit state. In order to verify that it is indeed possible to set a well-defined initial state with this technique one has to add a term for the electric contacts to the Hamiltonian. This will then give rise to new insights into how electric currents will influence the proposed system. To extend our model away from the adiabatic regime we will have to solve the Schrödinger equation and study how fast the braiding can be done without destroying the coherence of the system.

11 Conclusion

In conclusion, this thesis explores the fascinating realms of Skyrmion excitations, topological superconductors, and superconductor-magnet heterostructures.

During our exploration of Skyrmion excitation, we discovered that the eigenmodes of Skyrmions can be utilized for low-energy frequency multiplication. Through micromagnetic simulations, we demonstrate the exciting possibility of exciting the breathing mode and elliptical mode of Skyrmions at fractions of their eigenfrequencies. This opens up new avenues for developing low-energy frequency multipliers, which can have significant implications for various technological applications. Additionally, the tunability of these frequencies through temperature changes and external magnetic or electric fields enhances their versatility. Importantly, we develop a theory that extends beyond specific microscopic features, providing a robust framework for future exploration and utilisation of topological excitations in magnets.

In the realm of quantum computation with topological superconductors, we delve into the principles of braiding in a Kitaev chain. Furthermore, we present a novel technique for eliminating numerical phases in zero energy eigenvalues, providing a pathway for precise analysis of the braiding in more complex systems. Furthermore, we outline the implementation of braiding in a T-junction by controlling the topological and trivial phases of the superconductor through the chemical potential. By employing numerical simulations, we validate the obtained braiding statistics, as initially proposed by Ivanov [29].

In the investigation of superconductor-magnet heterostructures towards quantum computation, we establish a fundamental connection between these heterostructures and the Kitaev chain, uncovering shared physics that facilitates the utilisation of the heterostructures for quantum computing applications. We introduced a Hamiltonian that accurately represents the superconductor-magnet heterostructure. We develop a powerful numerical framework capable of computing the system's eigensystem and solving the self-consistency condition of the BCS theory, enabling in-depth investigations. Within this framework, we uncover significant insights, including the manipulation and ordering of superconducting vortices through interactions with magnetic domain walls. Additionally, we discover the localisation of Majorana modes at Skyrmion-Vortex pairs, providing a promising avenue for braiding Majoranas. Furthermore, we demonstrate the creation, fusion, and scalability of Majoranas using Skyrmion-Vortex pairs. Moreover, we outline how the utilisation of Skyrmion-Vortex pairs enables the performance of Clifford gates, paving the way for quantum computation.

Collectively, this thesis presents a comprehensive exploration of Skyrmion excitations, topological superconductors, and superconductor-magnet heterostructures. The findings and methodologies presented here contribute to the development of low-energy frequency multipliers, robust braiding techniques, and the realization of quantum computing architectures. By harnessing the unique properties of the superconductor-heterostructures, we can advance the field of quantum computation and pave the way for technologies.

Part IV.

Appendix

A

Program Code for Frequency Multiplication of Skyrmion Modes

A.1. Relaxation and Excitation of a Skyrmion

The simulations are carried out with **mumax**^{3} [42]. In the following, we show a sample script which was used to relax a material hosting a Skyrmion to its energy minimum. Therefore we employ a finite difference discretization of space as defined in Equation (2.7) used in chapter 2.

```
// Define sample dimensions
lx := 600e-9
ly := 600e-9
lz := 1e-9
nx := 600
ny := 600
nz := 1
dx := lx / nx
dy := ly / ny
dz := lz / nz

// Define material parameters
satmag := 1.1e6        //  A/m
jex    := 1.6e-11      //  J/m
dm     := 2.5e-3      //  J/m^2
ani    := 0.51e6       //  J/m^3
dampinga := 1.0e-3

// Set geometry
SetGridSize(nx, ny, nz)
SetCellSize(dx, dy, dz)

SetPBC(3, 3, 0)         // Periodic boundary conditions
EnableDemag = False     // turn off stray field


// Set material parameters
Msat  = satmag
Aex   = jex
AnisU = vector(0, 0, 1)
Ku1   = ani
Dind  = dm
alpha = dampinga

//Relax Skyrmion
m = NeelSkyrmion(1,-1)
relax()
```

```
save(m)    // relaxed state
```

After we have relaxed the material to its energy minimum we can load the relaxed state into a new script shown below. In this script we excite the Skyrmion within the material by applying a magnetic field pulse. In order to excite all modes of the Skyrmion we chose magnetic field pulses with different frequency ranges. After that we let the system relax for 20 ns so that non-eigenmodes decay and we are left with the eigenmodes of the Skyrmion. We collected data about the eigenmodes for another 10 ns.

```
lx := 600e-9
ly := 600e-9
lz := 1e-9
nx := 600
ny := 600
nz := 1
dx := lx / nx
dy := ly / ny
dz := lz / nz

// Define material parameters
satmag := 1.1e6        //  A/m
jex    := 1.6e-11        //  J/m
dm     := 2.5e-3      //  J/m^2
ani    := 0.51e6       //  J/m^3
dampinga := 1.0e-3

t0 := 0.1e-9 // The centre of the sinc pulse
f1 := 25e9
f2 := 50e9
f3 := 100e9
f4 := 200e9
f5 := 500e9
h_max := 0.05

totaltime :=10e-9 //total time of simulation
deltatime :=1e-11 //interval of time to print off

// Set geometry
SetGridSize(nx, ny, nz)
SetCellSize(dx, dy, dz)

SetPBC(3, 3, 0)         // Periodic boundary conditions
EnableDemag = False     // turn off stray field

// Set material parameters
Msat  = satmag
Aex   = jex
AnisU = vector(0, 0, 1)
Ku1   = ani
Dind  = dm
alpha = dampinga

//Initialize Skyrmion
m.LoadFile("Relax.out/m000000.ovf")

//Run Pulse
B_ext = vector(h_max * sin(2 * pi * f1 * (t - t0)) / (2 * pi * f1 * (t -
    t0)), 0, 0)
run(2*t0)
```

```
B_ext = vector(h_max * sin(2 * pi * f2 * (t - 3*t0)) / (2 * pi * f2 * (t -
     3*t0)), 0, 0)
run(2*t0)
B_ext = vector(h_max * sin(2 * pi * f3 * (t - 5*t0)) / (2 * pi * f3 * (t -
     5*t0)), 0, 0)
run(2*t0)
B_ext = vector(h_max * sin(2 * pi * f4 * (t - 7*t0)) / (2 * pi * f4 * (t -
     7*t0)), 0, 0)
run(2*t0)
B_ext = vector(h_max * sin(2 * pi * f5 * (t - 9*t0)) / (2 * pi * f5 * (t -
     9*t0)), 0, 0)
run(2*t0)

B_ext = vector(0, 0, 0)
run(20e-9)  // Let it run for 20 ns to suppress non eigenmodes

autosave(m, deltatime)
run(totaltime)
```

A.2. Excitation of Skyrmion Eigenmodes

We address separately the eigenmodes of the Skyrmion with the following script. Here we apply a magnetic field with the corresponding frequency, let it run for 40 ns so that the eigenmode adjusts to the excitation, and collect data for 30 ns.

```
lx := 600e-9
ly := 600e-9
lz := 1e-9
nx := 600
ny := 600
nz := 1
dx := lx / nx
dy := ly / ny
dz := lz / nz

//Kappa parameter
kappa:= 1.3 //initial kappa parameter

//Reduced current
u:= 0.0 //inital reduced current


satmag := 1.1e6        //  A/m
jex    := 1.6e-11        //  J/m
dm     := 2.8e-3       //  J/m^2
ani    := 0.51e6        //  J/m^3
dampinga := 1.0e-3

frequency:= 3.59564344e+09  // frequency of the applied in-plane field
h0 := 0.10263157894736842   // amplitude of the applied in-plane field

totaltime :=30e-9 //total time of simulation
deltatime :=0.01e-9 //interval of time to print off

// Set geometry
SetGridSize(nx, ny, nz)
SetCellSize(dx, dy, dz)
```

```

SetPBC(3, 3, 0)          // Periodic boundary conditions
EnableDemag = False      // turn off stray field


// Set material parameters
Msat  = satmag
Aex   = jex
AnisU = vector(0, 0, 1)
Ku1   = ani
Dind  = dm
alpha = dampinga

//Initialize Skyrmion
m.LoadFile("Relax.out/m000000.ovf")
//SaveAs(m,sprintf("skyrmion_relaxed_j=%.3f_k=%.3f_d=%.3f.ovf",jex,ani,dm))


//Run Excitation
MaxErr=1e-6
B_ext = vector(h0*Sin(2*Pi*frequency*t), 0, 0)
autosave(m, deltatime)
run(40e-9)
run(totaltime)
```

A.3. Scripts to Analyze the Skyrmion Excitation

With the following script, we are able to calculate the power spectral density (PSD) as defined in Equation (4.2). For that, we have to read the simulation time and the spatial dimension of the material from our **mumax**^{3} simulations defined in the functions `getTime` and `getSpatialDimensions`. In the function `Psd` we read in the relaxed state of the Skyrmion and the Skyrmion states during the excitation. We then subtract the relaxed Skyrmion state from the Skyrmion states during the excitation and then calculate the PSD according to Equation (4.2).

```
import numpy as np
import sys
import discretisedfield as df
import os
import time
import cupy as cp

def getTime(file):
    with open(file, "rb") as ovffile:
        f = ovffile.read()
        lines = f.split(b"\n")

        mdatalines = filter(lambda s: s.startswith(bytes("#", "utf-8")), lines)
        for line in mdatalines:
            if b"Total simulation time" in line:
                return float(line.split()[5])


def getSpatialDimensions(file):
    with open(file, "rb") as ovffile:
        f = ovffile.read()
        lines = f.split(b"\n")

        xNodes = 0
        yNodes = 0
        zNodes = 0

        mdatalines = filter(lambda s: s.startswith(bytes("#", "utf-8")), lines)
        for line in mdatalines:
            if b"xnodes" in line:
                xNodes = int(line.split()[2])
            elif b"ynodes" in line:
                yNodes = int(line.split()[2])
            elif b"znodes" in line:
                zNodes = int(line.split()[2])

            if xNodes != 0 and yNodes != 0 and zNodes != 0:
                return xNodes, yNodes, zNodes


def Psd(directory, relaxedStateFile):

    """ Spatially-resolved power spectral density as a function of frequency f """

    relaxedStateArray = df.Field.fromfile(relaxedStateFile).array

    xNodes, yNodes, zNodes = getSpatialDimensions(relaxedStateFile)

    relaxedStateArray = relaxedStateArray[int(xNodes/2)-50:int(xNodes/2)+50,int(yNodes/2)-50:int(yNodes/2)+50,:,:]

    dimx = relaxedStateArray.shape[0]
    dimy = relaxedStateArray.shape[1]

    N = dimx * dimy * zNodes

    filesToScan = []
    for file in os.listdir(directory):
        if file.endswith(".ovf") and file.startswith("m"):
            filesToScan.append(directory + file)

    filesToScan = sorted(filesToScan)

    allFiles = np.zeros((len(filesToScan), len(relaxedStateArray), len(relaxedStateArray[0]), len(relaxedStateArray[0][0]), len(relaxedStateArray[0][0][0])))
    timesArray = np.zeros(len(filesToScan))

    start = time.time()

    for i, file in enumerate(filesToScan):
        print("Loading file", i+1, "of", len(filesToScan))
        allFiles[i] = df.Field.fromfile(file).array[int(xNodes/2)-50:int(xNodes/2)+50,int(yNodes/2)-50:int(yNodes/2)+50,:,:]
        timesArray[i] = getTime(file)

    diffArray = allFiles - relaxedStateArray

    diffArray = diffArray.reshape((len(allFiles), dimx * dimy * zNodes, 3)
        )

    middle = time.time()

    Psd=np.array([np.fft.rfftfreq(diffArray.shape[0],d=(timesArray[1]-
        timesArray[0]))/1e9,np.sum(1/N*np.sum(np.abs(np.fft.rfft(diffArray
        ,axis=0))**2,axis=1),axis=1)])

    end = time.time()

    print(’loading time:’)
    print(middle-start)
    print(’fft time:’)
    print(end - middle)

    return Psd

directory = "Sinc_Pulse.out/"
initialFile = "Relax.out/m000000.ovf"
Psd = Psd(directory, initialFile)
np.save("Psd", Psd)
```

To analyze a single mode of a Skyrmion we read in the z component of the simulation data where we excited the Skyrmion with one of its eigenfrequency. We then extract the boundary of the Skyrmion by taking the elementwise absolute of the gathered data. By employing this procedure, the boundary of the Skyrmion looks like a white donut as shown in figure 4.4 a). We then use the *skimage* tool [72] to extract the contours, which gives us an inner and an outer contour. Next, we calculate the centroid of these contours so that we can get the radius at every point of the contour. After that, we cut the contour into different sections where we measure the radius so that we have a skeleton of the Skyrmion which we follow during its excitation. The lists of radii we then export for the corresponding Plot later. A sample picture of the contour extracted Skyrmion can best be seen in figure 4.4 a).The subsequent code demonstrates the implementation of this procedure.

```
import numpy as np
import matplotlib.pyplot as plt
from glob import glob
from os import getcwd, chdir
from skimage import measure
import discretisedfield as df
import sys

folder = str(sys.argv[1])

saved=getcwd()
chdir(folder)
n= sorted(glob(’*.ovf’))
chdir(saved)

radiusnumber=1000+1
radiuslist = np.zeros(shape=(len(n),radiusnumber))
radiuslist2 = np.zeros(shape=(len(n),radiusnumber))


```

```python
i=0

for number in n:
    print(i)
    #Import Skyrmion data
    read_field = df.Field.fromfile(folder+number)

    vecztmp = read_field.array[:,:,0,2]

    vecz=abs(vecztmp)

    contours = measure.find_contours(vecz, 0.8)

    centroid_x = np.median(contours[0][:,1])
    centroid_y = np.median(contours[0][:,0])

    x=contours[0][:,1]-centroid_x
    y=contours[0][:,0]-centroid_y

    r=np.sqrt(x**2 + y**2)
    a=np.arctan2(y, x)

    angle = np.linspace(-np.pi,np.pi,radiusnumber)

    radius = np.interp(angle,a,r,period=360)

    radiuslist[i]=radius

    centroid_x2 = np.median(contours[1][:,1])
    centroid_y2 = np.median(contours[1][:,0])

    x2=contours[1][:,1]-centroid_x2
    y2=contours[1][:,0]-centroid_y2

    r2=np.sqrt(x2**2 + y2**2)
    a2=np.arctan2(y2, x2)

    angle2 = np.linspace(-np.pi,np.pi,radiusnumber)

    radius2 = np.interp(angle2,a2,r2,period=360)

    radiuslist2[i]=radius2

    fig = plt.figure(frameon=False)
    fig.set_size_inches(vecz.shape[0]*0.01,vecz.shape[1]*0.01)
    ax = plt.Axes(fig, [0., 0., 1., 1.])
    ax.set_axis_off()
    fig.add_axes(ax)
    plt.imshow(-vecz, cmap=plt.cm.gray)
    plt.plot(contours[0][:,1],contours[0][:,0],'r')
    plt.plot(contours[1][:,1],contours[1][:,0],'g')
    plt.gca().set_aspect("equal")
    plt.savefig(folder+number.replace('.ovf', '.png'))
    plt.close(fig)
    plt.gcf().clear()

    print('')
    i=i+1

np.save('Plot_'+folder.replace('.out','')+"data",[radiuslist,radiuslist2])
```

A.4. Scripts for the Frequency Multiplication of Skyrmion Modes

The following code loads the data of our script that calculates the Psr. Here we use the *scipy.signal* package [143] to find the peaks in our spectrum. We then save the corresponding frequencies of the peaks for further processing. A sample plot of the Psr can be seen in the right panel of figure 4.3.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks

data = np.load("Psr.npy")

frequencies=data[0]
Psr= data[1]

peaks, _ = find_peaks(Psr, prominence=1e-3)

plt.plot(frequencies[2:-1] , Psr[2:-1])
plt.xlim(0,30)
plt.xlabel("Frequency (GHz)")
plt.ylabel(r"$P_{\mathrm{sr}}$ (A.U.)")
plt.plot(frequencies[peaks], Psr[peaks], "r.")
plt.semilogy()
plt.savefig("Psr.png")

print(frequencies[peaks])
print(Psr[peaks])
np.save(’Psr_freq’,frequencies[peaks])
```

We plot the different frequencies with respect to the rescaled DMI as a scatter plot and assign the different modes with different colours. The corresponding plot can be seen in the left panel of figure 4.3.

```
import matplotlib.pyplot as plt
import matplotlib
import numpy as np
import scipy
from numpy import genfromtxt

data = genfromtxt(’FrequencyvsDMI2.csv’, delimiter=’;’,names=True,dtype=
    None,encoding=’utf-8-sig’)

print(data.dtype.names)

font = {’size’   : 15}

matplotlib.rc(’font’, **font)

plt.scatter(data[’DMI’],data[’Breathing_frequency’],20,color=’red’,
    linestyle=’-’,marker=’o’,alpha=0.5,label=r"$\omega_0$, breathing")
plt.scatter(data[’DMI’],data[’Extra_peak_3_frequency’],20,color=’tomato’,
    linestyle=’-’,marker=’o’,alpha=0.5,label=r"$2 \omega_0$")
plt.scatter(data[’DMI’],data[’Gyrotropic_frequency’],20,color=’royalblue’,
    linestyle=’-’,marker=’o’,alpha=0.5,label=r"$\omega_1$, gyrotropic")
plt.scatter(data[’DMI’],data[’Elliptic_frequency’],20,color=’forestgreen’,
    linestyle=’-’,marker=’o’,alpha=0.5, label=r"$\omega_2$, elliptical")
```

```
plt.scatter(data['DMI'],data['Triangular_frequency'],20,color='gold',
    linestyle='-',marker='o',alpha=0.5,label=r"$\omega_3$, triangular")
plt.axhline(y=25, color='k', linestyle='-',linewidth='1')
plt.fill_between(data['DMI'], 25, 30,color="lightgray")
plt.xlim(min(data['DMI']),max(data['DMI']))
plt.ylim(0,30)
plt.xlabel(r"$D/D_c$")
plt.ylabel("Frequency (GHz)")
plt.annotate('Magnon Continuum',(2.95,27),style='italic')
plt.legend(loc=(0.02,0.35), prop={'size': 12})
plt.tight_layout()
plt.savefig('Frequency vs rescaled DMI_v2.pdf')
plt.gcf().clear()
```

From the analysis of the excitation of a single eigenmode of the Skyrmion, we can plot the radius with respect to time and the corresponding frequency by applying a Fourier transformation to the radius data. Considering that we have two contours for the Skyrmion, resulting in two radii, we calculate the average of the two values. This results in a Skyrmion radius defined as the distance from the centre to the boundary where the z component is zero. A sample picture of the radius against the time can be seen in figure 4.4 b), where one can also see, that the eigenmode starts in a transient phase until it adjusts to the applied frequency. In figure 4.4 c) we show a sample picture of the Fourier transformation of the radius data.

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.signal import find_peaks
from skimage import measure
import discretisedfield as df
import copy
import sys

def getTimeLength(file):
    with open(file, "rb") as ovffile:
        f = ovffile.read()
        lines = f.split(b"\n")
        time=length=0

        mdatalines = filter(lambda s: s.startswith(bytes("#", "utf-8")),
            lines)
        for line in mdatalines:
            if b"Total simulation time" in line:
                time=float(line.split()[5])
            elif b"zmax" in line:
                length=float(line.split()[2])
        return [time,length]

def fourier_transform(dt,fkt):

    Fkt = np.fft.rfft(fkt)/len(fkt)*2

    return Fkt

folder = str(sys.argv[1])

datatmp = np.load(folder+'data.npy')

```

```
dt=getTimeLength(folder.replace('Plot_','').replace('/','')+".out/m000001.
    ovf")[0]-getTimeLength(folder.replace('Plot_','').replace('/','')+".
    out/m000000.ovf")[0]
ds=getTimeLength(folder.replace('Plot_','').replace('/','')+".out/m000000.
    ovf")[1]
t = np.linspace(dt,dt*datatmp[0].shape[0],datatmp[0].shape[0])
n = t.size
f=np.linspace(0, 1/(2*dt), int(n/2+2))[:-1]

#read in relaxed radii
read_field = df.Field.fromfile("Relax.out/m000000.ovf")

vecztmp = read_field.array[:,:,0,2]

vecz=abs(vecztmp)

contours = measure.find_contours(vecz, 0.8)

centroid_x0 = np.median(contours[0][:,1])
centroid_y0 = np.median(contours[0][:,0])

x0=contours[0][:,1]-centroid_x0
y0=contours[0][:,0]-centroid_y0

r0=np.sqrt(x0**2 + y0**2)
a0=np.arctan2(y0, x0)

angle0 = np.linspace(-np.pi,np.pi,datatmp[0].shape[0])

radius0 = np.interp(angle0,a0,r0,period=360)

centroid_x1 = np.median(contours[1][:,1])
centroid_y1 = np.median(contours[1][:,0])

x1=contours[1][:,1]-centroid_x1
y1=contours[1][:,0]-centroid_y1

r1=np.sqrt(x1**2 + y1**2)
a1=np.arctan2(y1, x1)

angle1 = np.linspace(-np.pi,np.pi,datatmp[0].shape[0])

radius1 = np.interp(angle1,a1,r1,period=360)

meanradius0=np.sum(radius0)/len(radius0)
meanradius1=np.sum(radius1)/len(radius1)

meandws=abs(meanradius0-meanradius1) #dws is domainwallsize

print("Mean Radius of relaxed state in nm:",(meanradius0*ds)/1e-9)
print("Mean Domain wall size of relaxed state in nm:",(meandws*ds)/1e-9)
print("")

#fourier transform all modes
data=copy.deepcopy(datatmp[0])

meanall = 0
fftall=0

for i in range(data.shape[1]):
    meanall+= data[:,i]
    data[:,i]= data[:,i]-radius0

for i in range(data.shape[1]):
    fftall+= abs(fourier_transform(dt,data[:,i]))

meanall=meanall/data.shape[1]
fftall=fftall/data.shape[1]


plt.plot(t/1e-9,(meanall*ds)/1e-9)
plt.xlabel("Time (ns)")
plt.ylabel("Radius (nm)")
plt.xlim(0,max(t/1e-9))
plt.savefig(folder+’Analyseradius.svg’)
plt.gcf().clear()

plt.plot(t/1e-9,(meanall*ds)/1e-9)
plt.xlabel("Time (ns)")
plt.ylabel("Radius (nm)")
plt.xlim(68,max(t/1e-9))
plt.savefig(folder+’Analyseradius2.svg’)
plt.gcf().clear()

peaks, _ = find_peaks(abs(fftall),prominence=1)

plt.plot(f[2:-1]/1e9,abs(fftall[2:-1]))
plt.xlim(0,30)
plt.xlabel("Frequency (GHz)")
plt.ylabel("Amplitude (A.u.)")
plt.savefig(folder+’Analysefftradius.svg’)
plt.gcf().clear()

print(f[peaks])
np.save(folder+’Contourdata’,[f/1e9,abs(fftall)])
np.save(folder+’/Frequencies_Contour’,f[peaks])
np.save(folder+’/data_Relaxed’,[(meanradius0*ds)/1e-9,(meandws*ds)/1e-9])
np.save(folder+’/data_min-max-Amp’,[(min(meanall)*ds)/1e-9,(max(meanall)*
    ds)/1e-9])
```

We show the code which plots the radius of the Skyrmion with respect to the applied frequency. Within this code, we have to read the amplitude and the frequency that we calculated before. Then we have to select the frequency we want to observe where we choose a certain range in which we expect the peak. We then plot the amplitude of the peak with respect to the applied frequency. We showed two plots using this code in the main text. First, we analyzed the breathing mode in figure 4.5 and then we also analyzed the elliptical mode in figure 4.8.

```
import matplotlib.pyplot as plt
import matplotlib
import numpy as np
import os

font = {’size’   : 15}

matplotlib.rc(’font’, **font)

AmplitudeList=[]
```

```
frequency=[7.19128688]
frequencylist=np.linspace(0,1.0333333333333333,102)[1:-1]*frequency

for f in frequencylist:

  data = np.load(’Plot_Excitation_inplane_’+str(round(f,2))+’/Contourdata.npy’)
  [relaxed_radius, relaxed_dw] = np.load(’Plot_Excitation_inplane_’+str(round(f,2))+’/data_Relaxed.npy’)
  f = np.load(’Plot_Excitation_inplane_’+str(round(f,2))+’/Frequencies_Contour.npy’)
  newf=f[np.argmin(abs(f-7.10981627e+09), axis=None)]*1e-9
  if (7.1 < newf < 7.2):
    Amplitude=data[1][np.unravel_index(np.argmin(abs(data[0]-newf), axis=None),data[0].shape)]
  else:
    Amplitude=data[1][np.unravel_index(np.argmin(abs(data[0]-7.10981627), axis=None),data[0].shape)]
  AmplitudeList.append(Amplitude/(relaxed_radius))

plt.plot(frequencylist,AmplitudeList,color=’red’,marker=’o’)
plt.axvline(7.19128688,color=’k’,label=’f’)
plt.axvline(1/2*7.19128688,color=’k’,linestyle=’dashed’,label=’f/2’)
plt.axvline(1/3*7.19128688,color=’k’,linestyle=’dotted’,label=’f/3’)
plt.axvline(1/4*7.19128688,color=’k’,linestyle=’dashdot’,label=’f/4’)
plt.xlabel("Frequency (GHz)")
plt.ylabel("Amplitude/relaxed Radius")
plt.xlim(min(frequencylist),max(frequencylist))
plt.legend(loc=2, prop={’size’: 10})
plt.tight_layout()
plt.savefig(’Analysefftradius_Amplitudes.pdf’)
plt.gcf().clear()
```

B

Numerical and Analytical Methods for Topological Superconductors

This chapter provides a comprehensive elucidation of the analytical and numerical methodologies employed throughout this thesis. In the following, we will show a derivation of the braiding operator in a tight binding model, the free energy calculation to describe the domain wall interaction with a superconducting Vortex, the Python module to model a superconductor-magnet heterostructure and the scripts that were used to study the superconductor-magnet heterostructure.

B.1. Derivation of the Braiding Operator in a Tight Binding Model

Apart from being topological zero-energy bound states, the Majorana zero modes in the Kitaev chain are fascinating because they exhibit non-Abelian braiding. This means that the system's state can change when a Majorana is braided around another. Consider a system that combines four of these Majorana zero modes on some tracks and enables the interchange of two nearby Majoranas in the manner shown below to demonstrate this idea:

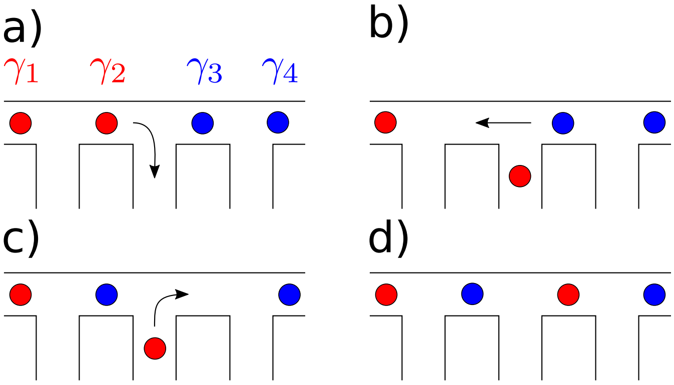

Figure B.1.: Braiding of Majorana zero modes in a T-Junction.

The following calculation was done by Tobias Meng and we show this derivation in this thesis. While the particle number in a Majorana system is not conserved, the fermionic parity is (even or odd numbers of Fermions in the system correspond to parity eigenvalues +1 and -1, respectively). However, proper fermionic states are required to define parity.

The properties of the fermionic modes $c = (\gamma_1 + i\gamma_2)/2$ and $d = (\gamma_3 + i\gamma_4)/2$ are given by

$$P_c = 2c^\dagger c - 1 = i\gamma_1\gamma_2 \quad \text{and} \quad P_d = 2d^\dagger d - 1 = i\gamma_3\gamma_4. \tag{A-1}$$

The total parity is given by $P_{\text{tot}} = P_d P_e$, and hence by

$$P_{\text{tot}} = -\gamma_1\gamma_2\gamma_3\gamma_4. \tag{A-2}$$

The exchange of Majoranas 2 and 3 (see Figure B.1) can be understood as an adiabatic evolution of the system as some microscopic parameters are changed and are thus represented by a unitary matrix U_{23}. Only the Majorana Fermion operators γ_2 and γ_3 can be involved in the evolution operator U_{23} because only these Majoranas are impacted by the braiding, taking into account that different Majoranas do not interact. Due to the fact that $\gamma_i\gamma_i = 1$, the most general operator involving only γ_2 and γ_3 can be written as

$$U_{23} = a + b\gamma_2 + c\gamma_3 + d\gamma_2\gamma_3 \tag{A-3}$$

with complex numbers a, b, c, d. Since the parity is not affected by the evolution, we have $[P_{\text{tot}}, U_{23}] = 0$. This commutator implies $b = 0 = c$. Unitary finally means $U_{23}^\dagger U_{23} = 1$, and hence $|a|^2 + |d|^2 + (a^*d - ad^*)\gamma_2\gamma_3 = 1$. Writing $a = |a|e^{i\varphi}$ and $d = |d|e^{i\theta}$, we find $\varphi = \theta$ and $|a|^2 + |d|^2 = 1$. The most general form of U_{23} is thus

$$U_{23} = e^{i\varphi}(\sin x + \cos x\ \gamma_2\gamma_3) \tag{A-4}$$

Using this form to execute the exchange we find

$$U_{23}\gamma_2 U_{23}^\dagger = -\cos 2x\ \gamma_2 - \sin 2x\ \gamma_3 \tag{A-5}$$

$$U_{23}\gamma_3 U_{23}^\dagger = -\cos 2x\ \gamma_3 + \sin 2x\ \gamma_2. \tag{A-6}$$

Since the two Majoranas have switched places, the evolution should result in $\gamma_2 \to \gamma_3$ (up to a phase) and $\gamma_3 \to \gamma_2$ (up to a phase). This translates to $2x = (n + 1/2)\pi$. The two x values that result in different operators U_{23} can then be chosen as $x = \pm\pi/4$. The ±-sign represents the braiding's rotation sense: one can braid as shown in the image or do the reverse process in which γ_3 is moved down in the first step. Inserting x into the equation before then yields the result we expect from [29]

$$U_{23}\gamma_2 U_{23}^\dagger = -\gamma_3 \tag{A-7}$$

$$U_{23}\gamma_3 U_{23}^\dagger = \gamma_2. \tag{A-8}$$

B.2. Domain Wall Interaction with a Superconducting Vortex

A. Ginzburg-Landau Theory of Noncentrosymmetric Superconductors

For noncentrosymmetric superconductors, the Ginzburg-Landau free energy takes the form of

$$F\left[\psi^*, \psi, \mathbf{A}, \mathbf{h}\right] = \int \mathrm{d}\mathbf{r} \left[\mathcal{F}_0 + \mathcal{F}_{me}^{(1)} + \mathcal{F}_{me}^{(2)}\right], \tag{A-9}$$

where $\mathcal{F}_{me}^{(1)}$ and $\mathcal{F}_{me}^{(1)}$ are defined below Equation (9.2) and $\mathcal{F}_0$ is given by

$$\mathcal{F}_0 = \frac{1}{4m}(\boldsymbol{\Pi}\psi)^* \cdot \boldsymbol{\Pi}\psi + a|\psi|^2 + \frac{b}{2}|\psi|^4 + \frac{|\mathbf{B}|^2}{8\pi}. \tag{A-10}$$

Here, $\boldsymbol{\Pi} = -i\hbar\boldsymbol{\nabla} - 2e\mathbf{A}/c$, and a and b determine the absolute value of the order parameter field ψ. Varying the free-energy functional in Equation (A-9) with respect to $\mathbf{A}$ and using that $\mathbf{j}_s = (c/4\pi)(\boldsymbol{\nabla} \times \mathbf{B})$ lead to the supercurrent density $\mathbf{j}_s = \mathbf{j}_{cs} + \mathbf{j}_{as}$. The anomalous supercurrent density $\mathbf{j}_{as}$ is then given by the equation

$$\mathbf{j}_{as} = 2e|\psi|^2 \left[\kappa^{(1)}(\hat{\mathbf{z}} \times \mathbf{h}) + \kappa^{(2)}(\hat{\mathbf{z}} \times \boldsymbol{\nabla} h_z)\right]. \tag{A-11}$$

Besides the conventional contribution of $\mathbf{j}_{cs} = e\boldsymbol{\mathcal{P}}/m$ generated by phase gradients in ψ ($2m$ is the mass of a Cooper pair), Equation (A-11) produces an additional contribution to the supercurrent, resulting in a net supercurrent density of $\mathbf{j}_s = \mathbf{j}_{cs} + \mathbf{j}_{as}$. According to Equation (A-11), magnetic textures, such as ferromagnetic domain walls, produce anomalous supercurrents in proximity-coupled superconductor-ferromagnet heterostructures. By incorporating the exchange field $\mathbf{h}^{(dw)}$ defined in Equation 9.1, we find the following supercurrent density associated with a domain wall of general helicity

$$\mathbf{j}_{as}^{(dw)}(x) = g^{(1)}(x)\, \hat{\mathbf{z}} \times \hat{\mathbf{n}}^{(dw)} - g^{(2)}(x)\, \hat{\mathbf{y}}, \tag{A-12}$$

where $g^{(1)} = \kappa^{(1)} 2eh_0|\psi|^2 \operatorname{sech}(x/\lambda)$ ($g^{(2)} = \kappa^{(2)} 2eh_0|\psi|^2 \operatorname{sech}^2(x/\lambda)/\lambda$) represents the current originating from $\mathcal{F}_{me}^{(1)}$ ($\mathcal{F}_{me}^{(2)}$), and $\hat{\mathbf{n}}^{(dw)} = [\cos(\gamma), \sin(\gamma), 0]$ is determined by the helicity of the domain wall. Specifically, for the $\kappa^{(1)}$-term, Néel domain walls give rise to supercurrents along the wall, while Bloch domain walls generate currents across the wall. Furthermore, the direction of the current switches sign under $\hat{\mathbf{n}}^{(dw)} \mapsto -\hat{\mathbf{n}}^{(dw)}$.The current density generated by the $\kappa^{(2)}$-term is independent of the domain wall's helicity and always flows along the wall.

By examining the interplay between $\mathbf{j}_{as}^{(dw)}$ and the supercurrent around a Vortex, we can obtain a simple physical picture of the interaction mechanism between domain walls and superconducting vortices. A superconducting Vortex, situated at the position $\mathbf{R} = [R_x, R_y, 0]$, can be characterized by an order parameter field of the form

$$\psi^{(v)}(\mathbf{r}) = \psi_0 \tanh\left(\frac{|\mathbf{r} - \mathbf{R}|}{\xi}\right) \exp(i\phi(\mathbf{r})). \tag{A-13}$$

In this equation, ϕ represents the phase of $\psi^{(v)}$, which varies by 2π in making a complete circuit counterclockwise about the centre $\mathbf{R}$ of the Vortex, ψ_0 denotes the absolute value of the order parameter far from the Vortex core, and ξ represents the coherence length. $\psi^{(v)}$ yields the following (conventional) supercurrent density

$$\mathbf{j}_{cs}^{(v)} = \frac{\hbar e \psi_0^2}{m} \frac{\tanh^2(|\mathbf{r} - \mathbf{R}|/\xi)}{|\mathbf{r} - \mathbf{R}|^2} \hat{\mathbf{z}} \times (\mathbf{r} - \mathbf{R}). \tag{A-14}$$

Note that $\mathbf{j}_{cs}^{(v)}$ is strongest when $|\mathbf{r} - \mathbf{R}| \sim \xi$ and approaches zero in the limits $|\mathbf{r} - \mathbf{R}| \to 0$ and $|\mathbf{r} - \mathbf{R}| \to \infty$.

B. Expressions for F_{eff} and $\mathbf{f}_{me}$

To derive approximate analytical expressions for F_{eff} and $\mathbf{f}_{me}$, we consider a sample that extends infinitely along the x-axis and has a width of $2L_y$ along the y-axis. A domain wall is positioned at $x = 0$. We assume that the Vortex is located near the domain wall and the origin of the sample, satisfying $|\mathbf{R}| \ll 1$. Additionally, we examine narrow domain walls where $\lambda \to 0$. By performing a series expansion of Equation (9.5) in terms of $\mathbf{R}$ and considering the scenarios where the heterostructure hosts Néel and Bloch walls, we obtain the following expressions for the effective free energies:

$$\begin{aligned} F_{\text{eff}}^{(nw)} &= (\epsilon F_0^{(1)} - F_0^{(2)}) \left[c_2 \tilde{R}_x^3 - c_1 \tilde{R}_x + c_3 \tilde{R}_x \tilde{R}_y^2\right] \\ F_{\text{eff}}^{(bw)} &= -\epsilon c_4 F_0^{(1)} \tilde{R}_y - F_0^{(2)} \left[c_2 \tilde{R}_x^3 - c_1 \tilde{R}_x + c_3 \tilde{R}_x \tilde{R}_y^2\right] \end{aligned} \tag{A-15}$$

Here, $\tilde{\mathbf{R}} = \mathbf{R}/\xi$, $F_0^{(1)} = \pi\hbar\lambda\kappa^{(1)} h_0 \psi_0^2$, $F_0^{(2)} = 2\hbar\kappa^{(2)} h_0 \psi_0^2$. $F_{\text{eff}}^{(nw)} = F_{\text{eff}}|_{\gamma=\pi/2-\epsilon\pi/2}$ and $F_{\text{eff}}^{(bw)} = F_{\text{eff}}|_{\gamma=\epsilon\pi/2}$ represent the free energies for systems containing Néel and Bloch walls, respectively, where $\epsilon \in \{1, -1\}$ depending on the helicity of the domain wall. In deriving the above expressions, we have utilized the approximations $(1/\pi\lambda)\operatorname{sech}(x/\lambda) \approx \delta(x)$ and $(1/2\lambda)\operatorname{sech}^2(x/\lambda) \approx \delta(x)$ in the limit $\lambda \to 0$ (where $\delta(x)$ is the Dirac delta function). To capture the main features of the numerically computed free energies in Figure 9.5, we have expanded $F_{\text{eff}}^{(nw)}$ to third order in $\mathbf{R}$ and $F_{\text{eff}}^{(bw)}$ to second order. Notably, the second-order contributions vanish in both cases. The expressions for the coefficients c_1, c_2, c_3, and c_4 in the series expansions in Equation (A-15) are given as

$$\begin{aligned} c_1 &= \int_{-\tilde{L}_y}^{\tilde{L}_y} g_1 \mathrm{d}\tilde{y}, \\ c_2 &= \int_{-\tilde{L}_y}^{\tilde{L}_y} g_2 \mathrm{d}\tilde{y}, \\ c_3 &= \int_{-\tilde{L}_y}^{\tilde{L}_y} g_3 \mathrm{d}\tilde{y} = \frac{2}{\tilde{L}_y^3}\left[1 - \operatorname{sech}^2\left(\tilde{L}_y\right)\left(1 + \tilde{L}_y \tanh\left(\tilde{L}_y\right)\right)\right], \\ c_4 &= \int_{-\tilde{L}_y}^{\tilde{L}_y} [g_1 - \tilde{y} g_4] \mathrm{d}\tilde{y} = \frac{2\tanh^2(\tilde{L}_y)}{\tilde{L}_y}. \end{aligned}$$

Here, $\tilde{y} = y/\xi$, $\tilde{L}_y = L_y/\xi$, and the dimensionless functions g_1, g_2, g_3, and g_4 are given by the expressions

$$\begin{aligned} g_1 &= \frac{\tanh^2(\tilde{y})}{\tilde{y}^2}, \\ g_2 &= -\frac{\tanh^2(\tilde{y})[\tilde{y}\operatorname{csch}(\tilde{y})\operatorname{sech}(\tilde{y}) - 1]}{\tilde{y}^4}, \\ g_3 &= -\frac{\operatorname{sech}^4(\tilde{y})[3\sinh^2(2\tilde{y}) - 8\tilde{y}\sinh(2\tilde{y}) - 4\tilde{y}^2(\cosh(2\tilde{y}) - 2)]}{4\tilde{y}^4}, \\ g_4 &= \frac{2\tanh(\tilde{y})[\tanh(\tilde{y}) - \tilde{y}\operatorname{sech}^2(\tilde{y})]}{\tilde{y}^3}. \end{aligned}$$

Exact analytic expressions for c_1 and c_2 are not obtainable, but the parameters approach the values $c_1 \approx 3.41$ and $c_2 \approx 1.32$ as $\tilde{L}_y$ tends to infinity. Importantly, all the coefficients are positive, depend on the system size L_y, and approach the values $c_1 \approx 3.41$, $c_2 \approx 1.32$, $c_3 = 2\xi^3/L_y^3$, and $c_4 = 2\xi/L_y$ when $L_y/\xi \gg 1$. Thus, the terms proportional to c_3 and c_4 account for finite-size effects, as these coefficients vanish in the limit $L_y/\xi \to \infty$.

C. Collective Coordinate Description

The dynamics of a Vortex is described by the following effective action $\mathcal{S}$ and dissipation functional $\mathcal{G}$

$$\begin{aligned} \mathcal{S}[\mathbf{R}, \dot{\mathbf{R}}] &= \int \mathrm{dt} \left[\frac{1}{2} m_v \dot{\mathbf{R}}^2 + \boldsymbol{\mathcal{A}}(\mathbf{R}) \cdot \mathbf{R} \right], && \text{(A-16)} \\ \mathcal{G}[\mathbf{R}, \dot{\mathbf{R}}] &= \int \mathrm{dt} \frac{1}{2} \alpha \dot{\mathbf{R}}^2, && \text{(A-17)} \end{aligned}$$

where $\boldsymbol{\nabla}_{\mathbf{R}} \times \boldsymbol{\mathcal{A}} = \mathbf{G} = 2\pi n_s q_v \hat{\mathbf{z}}$. The equation of motion of the Vortex is determined by the variational equation

$$\frac{\delta \mathcal{S}}{\delta \mathbf{R}} = \frac{\delta \mathcal{G}}{\delta \dot{\mathbf{R}}}. \tag{A-18}$$

Equation (A-18) yields the equation in Equation (9.8).

B.3. Skyrmion-Antivortex Pairs

Here we applied the same analysis as in Section 10.2 but for Skyrmion-Antivortex Pairs. Therefore we used a Néel Skyrmion with $\phi = \pi$ which binds to antivortices and plotted its energy spectrum in Figure B.2. Here we see, that the energy spectrum is mirrored compared with the one for the Skyrmion-Vortex Pairs shown in Figure 10.3. This comes from the negative vorticity of antivortices which introduces an overall minus sign in the proximity coupling term (8.4). In the energy spectrum, we do not have a gap closing and we therefore cannot observe Majorana modes. Furthermore, the lowest energy wavefunction is not well localized which additionally supports that we have not found Majorana zero modes in this system. In order to obey Majorana zero modes in this system more research has to be done and the parameters have to be optimized. One reason why the simple replacement of a Vortex through an Antivortex does not preserve the Majorana modes is that an Antivortex has a completely different symmetry in the order parameter.

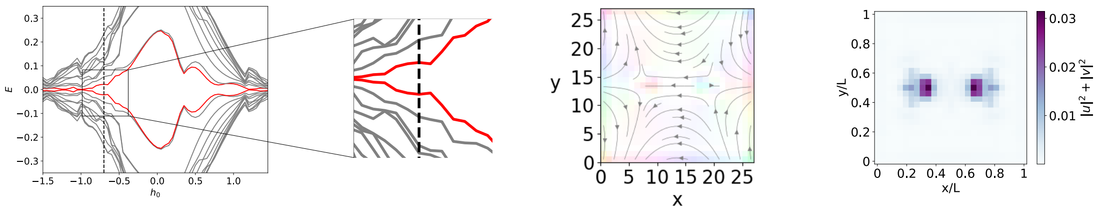

Figure B.2.: Low energy spectrum and the corresponding wavefunctions of the red marked modes of a superconductor-ferromagnet heterostructure with two Skyrmion-Antivortex Pairs as a function of magnetic field strength h_0. The lowest mode which could be Majorana mode is shown in red for better visibility. The horizontal line at $h_0 = -0.55$ is intended to clarify where we would have expected the gap closing to take place. We show a zoom as well as a Plot of the superconducting order parameter and a Plot of the Majorana modes (from left to right). The sample has dimensions L=27a. It is a similar analysis as shown in Figure 10.3.

B.4. Python Module to Model a Superconductor-Magnet Heterostructure

In Figure B.3 we show an overview of the self-written python module used to model a superconductor-magnet heterostructure. This module is based on a Matlab implementation by Kjetil Hals [113].

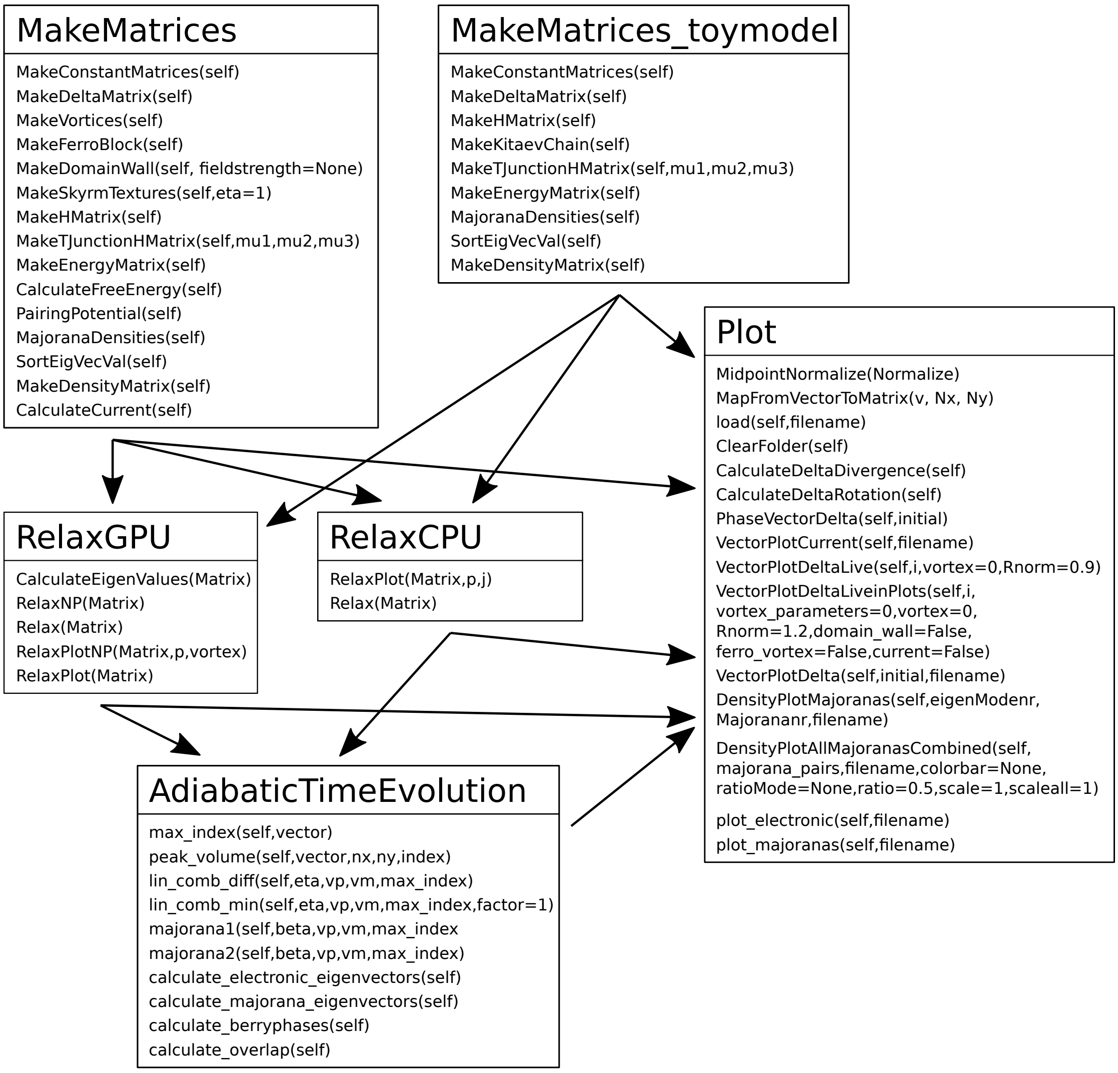
MakeMatrices
MakeConstantMatrices(self)
MakeDeltaMatrix(self)
MakeVortices(self)
MakeFerroBlock(self)
MakeDomainWall(self, fieldstrength=None)
MakeSkyrmTextures(self,eta=1)
MakeHMatrix(self)
MakeTJunctionHMatrix(self,mu1,mu2,mu3)
MakeEnergyMatrix(self)
CalculateFreeEnergy(self)
PairingPotential(self)
MajoranaDensities(self)
SortEigVecVal(self)
MakeDensityMatrix(self)
CalculateCurrent(self)
MakeMatrices_toymodel
MakeConstantMatrices(self)
MakeDeltaMatrix(self)
MakeHMatrix(self)
MakeKitaevChain(self)
MakeTJunctionHMatrix(self,mu1,mu2,mu3)
MakeEnergyMatrix(self)
MajoranaDensities(self)
SortEigVecVal(self)
MakeDensityMatrix(self)
Plot
MidpointNormalize(Normalize)
MapFromVectorToMatrix(v, Nx, Ny)
load(self,filename)
ClearFolder(self)
CalculateDeltaDivergence(self)
CalculateDeltaRotation(self)
PhaseVectorDelta(self,initial)
VectorPlotCurrent(self,filename)
VectorPlotDeltaLive(self,i,vortex=0,Rnorm=0.9)
VectorPlotDeltaLiveinPlots(self,i, vortex_parameters=0,vortex=0, Rnorm=1.2,domain_wall=False, ferro_vortex=False,current=False)
VectorPlotDelta(self,initial,filename)
DensityPlotMajoranas(self,eigenModenr, Majorananr,filename)
DensityPlotAllMajoranasCombined(self, majorana_pairs,filename,colorbar=None, ratioMode=None,ratio=0.5,scale=1,scaleall=1)
plot_electronic(self,filename)
plot_majoranas(self,filename)
RelaxGPU
CalculateEigenValues(Matrix)
RelaxNP(Matrix)
Relax(Matrix)
RelaxPlotNP(Matrix,p,vortex)
RelaxPlot(Matrix)
RelaxCPU
RelaxPlot(Matrix,p,j)
Relax(Matrix)
AdiabaticTimeEvolution
max_index(self,vector)
peak_volume(self,vector,nx,ny,index)
lin_comb_diff(self,eta,vp,vm,max_index)
lin_comb_min(self,eta,vp,vm,max_index,factor=1)
majorana1(self,beta,vp,vm,max_index
majorana2(self,beta,vp,vm,max_index)
calculate_electronic_eigenvectors(self)
calculate_majorana_eigenvectors(self)
calculate_berryphases(self)
calculate_overlap(self)

Figure B.3.: Overview of the numerical code. Here the boxes represent the different classes and within the boxes, the most important functions listed can be called from these classes.

The module is structured into different classes, that represent different steps of the numerical calculations. In the `MakeMatrices_toymodel` class we build a Hamiltonian without a magnet on top. In the `MakeMatrices` class a Hamiltonian is being put together which represents the superconductor-magnet heterostructure with different structures within the magnet and the superconductor. When we have the Hamiltonian defined we can then either call it within the `Plot` class where we can visualize different parts of the Hamiltonian, or we can call it within the `RelaxGPU` or `RelaxCPU` class to find the energy minimum. It is important to note here that only the superconductor is treated within this energy minimization and the magnet is assumed to be static. Using the class `Plot`, we can visually observe the dynamics during the relaxation process as well as examine the outcome after relaxation. But we can also treat the result further in the `AdiabaticTimeEvolution` class, where we can load different Matrices and compare them. In the following, we will show the most important functions within the classes `MakeMatrices_toymodel`, `MakeMatrices` and `RelaxGPU` to explain the implementation in more detail. For the other classes, the reader is referred to the github page of the code [144].

A. Defining Hamiltonian without Spin

Here we discuss functions from the class `MakeMatrices_toymodel`. To build the matrix we will first define the constants that we will use in the other functions. This is done by the function `MakeConstantMatrices`.

```
def MakeConstantMatrices(self):
        """ Define the constants that appear in the discretized
        version of the (scaled) Rashba Hamiltonian. It sets the following
            matrices:\n
        `self.V0`: The matrix carrying all the one particle information\n
        `self.tx`: The matrix describing hopping in x-direction\n
        `self.ty`: The matrix describing hopping in y-direction
        :param self.alpha: Dimensionless Rashba parameter
        :type self.alpha: float
        :param self.mu: Dimensionless chemical potential
        :type self.mu: float
        :param self.a: Lattice spacing
        :type self.a: float
        """

        #Write matrices:
        self.V0= (4*0 - self.mu)
        self.tx= -1
        self.ty= -1
```

The Hamiltonian consists of a part that incorporates the single particle terms ($\overleftrightarrow{h}$) and a part that consists of the interaction between two particles ($\overleftrightarrow{\Delta}$) as defined in Equation (6.11). In the following, we will define functions for both of these parts. We start with the interaction between two particles defined in the function `MakeDeltaMatrix`.

```
def MakeDeltaMatrix(self,eps=0):

        #Initiate H matrix:
        N= self.Nx*self.Ny
        DeltaMatrix= np.zeros(shape=(N, N),dtype=complex)

        deltax = self.deltax+eps
        deltay = self.deltay+eps
        deltaxd = -self.deltax-eps
        deltayd = -self.deltay-eps

        #Write Hmatrix:
        for i in range(self.Nx):
            for j in range(self.Ny):
                #Define indices:
```

```
                n= i + j*self.Nx
                ip= i+1
                im= i-1
                jp= j+1
                jm= j-1
                nip= ip + j*self.Nx
                nim= im + j*self.Nx
                njp= i + jp*self.Nx
                njm= i + jm*self.Nx
                #Write off-diagonal elements:
                if (i>0 and i<self.Nx-1 and j>0 and j<self.Ny-1): # internal lattice points
                    DeltaMatrix[n,nip]= deltax[n]
                    DeltaMatrix[n,nim]= deltaxd[n]
                    DeltaMatrix[n,njp]= deltay[n]
                    DeltaMatrix[n,njm]= deltayd[n]
                elif (i==0 and j==0): #corner points
                    if(i<self.Nx-1):
                        DeltaMatrix[n,nip]= deltax[n]
                    if(j<self.Ny-1):
                        DeltaMatrix[n,njp]= deltay[n]
                elif (i==0 and j==self.Ny-1):
                    if(i<self.Nx-1):
                        DeltaMatrix[n,nip]= deltax[n]
                    DeltaMatrix[n,njm]= deltayd[n]
                elif (j==0 and i==self.Nx-1):
                    DeltaMatrix[n,nim]= deltaxd[n]
                    if(j<self.Ny-1):
                        DeltaMatrix[n,njp]= deltay[n]
                elif (i==self.Nx-1 and j==self.Ny-1):
                    DeltaMatrix[n,nim]= deltaxd[n]
                    DeltaMatrix[n,njm]= deltayd[n]
                elif (i==0 and j>0 and j<self.Ny-1): #boundary points
                    if(i<self.Nx-1):
                        DeltaMatrix[n,nip]= deltax[n]
                    DeltaMatrix[n,njp]= deltay[n]
                    DeltaMatrix[n,njm]= deltayd[n]
                elif (i==self.Nx-1 and j>0 and j<self.Ny-1):
                    DeltaMatrix[n,nim]= deltaxd[n]
                    DeltaMatrix[n,njp]= deltay[n]
                    DeltaMatrix[n,njm]= deltayd[n]
                elif (j==0 and i>0 and i<self.Nx-1):
                    DeltaMatrix[n,nip]= deltax[n]
                    DeltaMatrix[n,nim]= deltaxd[n]
                    if(j<self.Ny-1):
                        DeltaMatrix[n,njp]= deltay[n]
                elif (j==self.Ny-1 and i>0 and i<self.Nx-1):
                    DeltaMatrix[n,nip]= deltax[n]
                    DeltaMatrix[n,nim]= deltaxd[n]
                    DeltaMatrix[n,njm]= deltayd[n]

        self.DeltaMatrix= DeltaMatrix
```

Next, we look at the single particle part in the function `MakeHMatrix` consisting of the onsite potential $V0$, which also contains the chemical potential mu, and the hopping parameters tx and ty.

```
def MakeHMatrix(self):
        """
        Purpose:
        The function returns the discretized version of the Rashba Hamiltonian with a position-dependent exchange field.

        Definitions:
        V0: on-site potential proportional to the 2 times 2 identity matrix (contains chemical potential).
        tx,ty: 2 times 2 hopping matrices (contains Rashba SOC).
        Nx, Ny: Number of lattice points along the x and y directions.
        bx=0 if open boundary condition along x-boundary, bx=1 if periodic boundary condition.
        """

        #Initiate H matrix:
        N= self.Nx*self.Ny
        Hmatrix= np.zeros(shape=(N, N),dtype=complex)

        #Define daggered matrices:
        txd= np.conj(np.transpose(self.tx))
        tyd= np.conj(np.transpose(self.ty))

        #Write Hmatrix:
        for i in range(self.Nx):
            for j in range(self.Ny):
                #Define indices:
                n= i + j*self.Nx
                ip= i+1
                im= i-1
                jp= j+1
                jm= j-1
                nip= ip + j*self.Nx
                nim= im + j*self.Nx
                njp= i + jp*self.Nx
                njm= i + jm*self.Nx
                #Write diagonal elements:
                V= self.V0
                Hmatrix[n,n]= V
                #Write off-diagonal elements:
                if (i>0 and i<self.Nx-1 and j>0 and j<self.Ny-1): # internal lattice points
                    Hmatrix[n,nip]= self.tx
                    Hmatrix[n,nim]= txd
                    Hmatrix[n,njp]= self.ty
                    Hmatrix[n,njm]= tyd
                elif (i==0 and j==0): #corner points
                    if(i<self.Nx-1):
                        Hmatrix[n,nip]= self.tx
                    if(j<self.Ny-1):
                        Hmatrix[n,njp]= self.ty
                elif (i==0 and j==self.Ny-1):
                    if(i<self.Nx-1):
                        Hmatrix[n,nip]= self.tx
                    Hmatrix[n,njm]= tyd
                elif (j==0 and i==self.Nx-1):
                    Hmatrix[n,nim]= txd
                    if(j<self.Ny-1):
                        Hmatrix[n,njp]= self.ty
                elif (i==self.Nx-1 and j==self.Ny-1):
                    Hmatrix[n,nim]= txd
                    Hmatrix[n,njm]= tyd
```

```
               elif (i==0 and j>0 and j<self.Ny-1): #boundary points
                   if(i<self.Nx-1):
                       Hmatrix[n,nip]= self.tx
                   Hmatrix[n,njp]= self.ty
                   Hmatrix[n,njm]= tyd
               elif (i==self.Nx-1 and j>0 and j<self.Ny-1):
                   Hmatrix[n,nim]= txd
                   Hmatrix[n,njp]= self.ty
                   Hmatrix[n,njm]= tyd
               elif (j==0 and i>0 and i<self.Nx-1):
                   Hmatrix[n,nip]= self.tx
                   Hmatrix[n,nim]= txd
                   if(j<self.Ny-1):
                       Hmatrix[n,njp]= self.ty
               elif (j==self.Ny-1 and i>0 and i<self.Nx-1):
                   Hmatrix[n,nip]= self.tx
                   Hmatrix[n,nim]= txd
                   Hmatrix[n,njm]= tyd

       self.Hmatrix= Hmatrix
```

By manipulating the chemical potential μ at different sites, we have the ability to determine whether the corresponding sample is topological or trivial locally. Consequently, we can incorporate a T-Junction structure into our sample using the function `MakeTJunctionHMatrix`. This class plays a crucial role in the results showcased in Section 7.5.

```
def MakeTJunctionHMatrix(self,mu1,mu2,mu3):
       #Initiate H matrix:
       N= self.Nx*self.Ny
       Hmatrix= np.zeros(shape=(N, N),dtype=complex)

       #Define daggered matrices:
       txd= np.conj(np.transpose(self.tx))
       tyd= np.conj(np.transpose(self.ty))

       #Write Hmatrix:
       for i in range(self.Nx):
           for j in range(self.Ny):
               #Define indices:
               n= i + j*self.Nx
               ip= i+1
               im= i-1
               jp= j+1
               jm= j-1
               nip= ip + j*self.Nx
               nim= im + j*self.Nx
               njp= i + jp*self.Nx
               njm= i + jm*self.Nx
               #Write diagonal elements:
               if (i<int(self.Nx/2) and i>0 and j==1):
                   Hmatrix[n,n]= -mu1[i-1]
               elif (i>int(self.Nx/2) and i<self.Nx-1 and j==1):
                   Hmatrix[n,n]= -mu2[i-int(self.Nx/2)-1]
               elif (i==int(self.Nx/2) and j>0 and j<self.Ny-1):
                   Hmatrix[n,n]= -mu3[j-1]
               else:
                   Hmatrix[n,n]=-self.mu
               #Write off-diagonal elements:
               if (i>0 and i<self.Nx-1 and j>0 and j<self.Ny-1): # internal lattice points
                   Hmatrix[n,nip]= self.tx
                   Hmatrix[n,nim]= txd
                   Hmatrix[n,njp]= self.ty
                   Hmatrix[n,njm]= tyd
               elif (i==0 and j==0): #corner points
                   if(i<self.Nx-1):
                       Hmatrix[n,nip]= self.tx
                   if(j<self.Ny-1):
                       Hmatrix[n,njp]= self.ty
               elif (i==0 and j==self.Ny-1):
                   if(i<self.Nx-1):
                       Hmatrix[n,nip]= self.tx
                   Hmatrix[n,njm]= tyd
               elif (j==0 and i==self.Nx-1):
                   Hmatrix[n,nim]= txd
                   if(j<self.Ny-1):
                       Hmatrix[n,njp]= self.ty
               elif (i==self.Nx-1 and j==self.Ny-1):
                   Hmatrix[n,nim]= txd
                   Hmatrix[n,njm]= tyd
               elif (i==0 and j>0 and j<self.Ny-1): #boundary points
                   if(i<self.Nx-1):
                       Hmatrix[n,nip]= self.tx
                   Hmatrix[n,njp]= self.ty
                   Hmatrix[n,njm]= tyd
               elif (i==self.Nx-1 and j>0 and j<self.Ny-1):
                   Hmatrix[n,nim]= txd
                   Hmatrix[n,njp]= self.ty
                   Hmatrix[n,njm]= tyd
               elif (j==0 and i>0 and i<self.Nx-1):
                   Hmatrix[n,nip]= self.tx
                   Hmatrix[n,nim]= txd
                   if(j<self.Ny-1):
                       Hmatrix[n,njp]= self.ty
               elif (j==self.Ny-1 and i>0 and i<self.Nx-1):
                   Hmatrix[n,nip]= self.tx
                   Hmatrix[n,nim]= txd
                   Hmatrix[n,njm]= tyd

       self.Hmatrix= Hmatrix
```

We implement the Kitaev chain toy model with the function `MakeKitaevChain`. This class is an essential part of the results we show in Section 7.2 and in Section 7.4.

```
def MakeKitaevChain(self,mu1):
        #Initiate H matrix:
        N= self.Nx*self.Ny
        Hmatrix= np.zeros(shape=(N, N),dtype=complex)

        #Define daggered matrices:
        txd= np.conj(np.transpose(self.tx))

        #Write Hmatrix:
        for i in range(self.Nx):
            #Define indices:
            ip= i+1
            im= i-1
            #Write diagonal elements:
```

```
            if (i<self.Nx and i>=0):
                Hmatrix[i,i]= -mu1[i-1]
            else:
                Hmatrix[i,i]=self.mu
            #Write off-diagonal elements:
            if (i>0 and i<self.Nx-1): #internal lattice points
                Hmatrix[i,ip]= self.tx
                Hmatrix[i,im]= txd
            elif (i==0): #corner points
                if(i<self.Nx-1):
                    Hmatrix[i,ip]= self.tx
            elif (i==self.Nx-1):
                Hmatrix[i,im]= txd

        self.Hmatrix= Hmatrix
```

We merge the `Hmatrix` and `DeltaMatrix` which we do in the function `MakeEnergyMatrix`.

```
def MakeEnergyMatrix(self,eps=0):
    """
    Combine the whole matrix
    """
    N = len(self.Hmatrix[0,:])
    EnergyMatrix = np.zeros(shape=(2*N,2*N),dtype= complex)
    EnergyMatrix[0:N,0:N] = self.Hmatrix
    EnergyMatrix[0:N,N:2*N] = self.DeltaMatrix
    EnergyMatrix[N:2*N,0:N] = -np.conjugate(self.DeltaMatrix)
    EnergyMatrix[N:2*N,N:2*N] = -np.conjugate(self.Hmatrix)

    self.EnergyMatrix= EnergyMatrix
```

B. Defining Hamiltonian with Spin

Here we discuss functions from the function `MakeMatrices`. We start with the parts of the Hamiltonian that stay mostly constant and incorporate the hopping as well as the spin-orbit coupling and the chemical potential.

```
def MakeConstantMatrices(self):
        """ Make the :math:`2 \\times 2` constant (dimensionless) matrices that appear in the discretized
        version of the (scaled) Rashba Hamiltonian. It sets the following matrices:\n
        `self.V0`: The matrix carrying all the one particle information\n
        `self.tx`: The matrix describing hopping in x-direction\n
        `self.ty`: The matrix describing hopping in y-direction
        :param self.alpha: Dimensionless Rashba parameter
        :type self.alpha: float
        :param self.mu: Dimensionless chemical potential
        :type self.mu: float
        :param self.a: Lattice spacing
        :type self.a: float
        """

        #Write matrices:
        self.V0= (4*0 - self.mu)*self.s0
        self.tx= -1*self.s0 + 1j*self.alphaR*self.s2
        self.ty= -1*self.s0 - 1j*self.alphaR*self.s1
```

In this part of the Hamiltonian one is also able to introduce structures in the superconducting order parameter. At first, we show a class that defines a superconductor with a uniform phase in the whole sample with the function `MakeDeltaMatrix`.

```
def MakeDeltaMatrix(self):
      """ The function makes the discretized version of the pairing potential
      :param self.delta: Vector containing the on-site pairing potential (:math:`Nx \cdot Ny` elements)
      :type self.delta: numpy array
      :param self.Nx: Number of lattice points along the x direction
      :type self.Nx: integer
      :param self.Ny: Number of lattice points along y direction
      :type self.Ny: integer
      """

      N= 2*self.Nx*self.Ny
      DeltaMatrix= np.zeros(shape=(N, N),dtype=complex)

      for i in range(self.Nx):
          for j in range(self.Ny):
              n= 2*i + 2*j*self.Nx
              nd= i + j*self.Nx
              DeltaMatrix[n:n+2,n:n+2]= self.delta[nd]*self.p

      self.DeltaMatrix=DeltaMatrix
```

We show how a single Vortex and multiple vortices are defined. Here the two parameters `self.Vrx` and `alternate` of the function `MakeVortices` are crucial. By examining the list of x-values in `self.Vrx`, we can determine the number of vortices to be inserted into the superconductor. Additionally, the parameter `alternate` allows us to regulate whether neighboring vortices possess opposite chirality. The function `f(x,y,x0,y0,sig` is a Gaussian function that we convolved with the Vortex profile to make it localized so that we can add different Vortex profiles. This class is used extensively in Section 8.3.2, Chapter 9 and Chapter 10.

```
def MakeVortices(self,alternate=True):
        def f(x,y,x0,y0,sig):
            return np.exp(-((x-x0)**2/sig**2+(y-y0)**2/sig**2))

        for i in range(self.Nx):
            for j in range(self.Ny):
                phase=0
                for k in range(len(self.Vrx)):
                    n= i + j*self.Nx
                    vx= np.double(i-self.Vrx[k])
                    vy= np.double(j-self.Vry[k])
                    v= [vx, vy]
                    if alternate:
                        if(k%2==0):
                            phase= phase+(f(i,j,self.Vrx[k],self.Vry[k],5)*np.exp(1j*self.Vq*(np.angle(complex(v[0],v[1]))+self.thetaVortex)))
                        else:
                            phase= phase+(f(i,j,self.Vrx[k],self.Vry[k],5)*np.exp(1j*self.Vq*(np.angle(complex(v[0],v[1]))+self.thetaVortex+np.pi)))
                    else:
                        phase=phase+(f(i,j,self.Vrx[k],self.Vry[k],5)*np.exp(1j*self.Vq*(np.angle(complex(v[0],v[1]))+self.thetaVortex)))
                self.delta[n]= phase
```

In the single-particle sector of the Hamiltonian, the control over whether our sample resides in the topological or trivial regime remains the same as before. However, instead of manipulating the chemical potential, we introduce a proximity coupling between the superconductor and a magnet through the parameter h, representing the magnet's exchange field. As discussed in Section 8.2, we can regulate the topological nature of the sample by adjusting the parameter h. This characteristic is utilized in the subsequent classes, starting with the presentation of a class that defines a ferromagnet in the `MakeHMatrix` function.

```
def MakeHMatrix(self):
      """
      Purpose:
      The function returns the discretized version of the Rashba Hamiltonian with a position-dependent exchange field.
      Definitions:
      V0: on-site potential proportional to the 2 times 2 identity matrix (contains chemical potential).
      tx,ty: 2 times 2 hopping matrices (contains Rashba SOC).
      h: exchange field (3*Nx*Ny elements).
      Nx, Ny: Number of lattice points along the x and y directions.
      bx=0 if open boundary condition along x-boundary, bx=1 if periodic boundary condition.
      """

      #Initiate H matrix:
      N= 2*self.Nx*self.Ny
      Hmatrix= np.zeros(shape=(N, N),dtype=complex)

      #Define daggered matrices:
      txd= np.conj(np.transpose(self.tx))
      tyd= np.conj(np.transpose(self.ty))

      #Write Hmatrix:
      for i in range(self.Nx):
          for j in range(self.Ny):
              #Define indices:
              n= 2*i + 2*j*self.Nx
              ip= i+1
              im= i-1
              jp= j+1
              jm= j-1
              nip= 2*ip + 2*j*self.Nx
              nim= 2*im + 2*j*self.Nx
              njp= 2*i + 2*jp*self.Nx
              njm= 2*i + 2*jm*self.Nx
              nh= 3*i + 3*j*self.Nx
              #Write diagonal elements:
              htemp= self.h[nh:nh+3]
              V= self.V0 + htemp[0]*self.s1 + htemp[1]*self.s2 + htemp[2]*self.s3
              Hmatrix[n:n+2,n:n+2]= V
              #Write off-diagonal elements:
              if (i>0 and i<self.Nx-1 and j>0 and j<self.Ny-1): # internal lattice points
                  Hmatrix[n:n+2,nip:nip+2]= self.tx
                  Hmatrix[n:n+2,nim:nim+2]= txd
                  Hmatrix[n:n+2,njp:njp+2]= self.ty
                  Hmatrix[n:n+2,njm:njm+2]= tyd
              elif (i==0 and j==0): #corner points
                  if(i<self.Nx-1):
                      Hmatrix[n:n+2,nip:nip+2]= self.tx
                  if(j<self.Ny-1):
                      Hmatrix[n:n+2,njp:njp+2]= self.ty
              elif (i==0 and j==self.Ny-1):
                  if(i<self.Nx-1):
                      Hmatrix[n:n+2,nip:nip+2]= self.tx
                  Hmatrix[n:n+2,njm:njm+2]= tyd
              elif (j==0 and i==self.Nx-1):
                  Hmatrix[n:n+2,nim:nim+2]= txd
                  if(j<self.Ny-1):
                      Hmatrix[n:n+2,njp:njp+2]= self.ty
              elif (i==self.Nx-1 and j==self.Ny-1):
                  Hmatrix[n:n+2,nim:nim+2]= txd
                  Hmatrix[n:n+2,njm:njm+2]= tyd
              elif (i==0 and j>0 and j<self.Ny-1): #boundary points
                  if(i<self.Nx-1):
                      Hmatrix[n:n+2,nip:nip+2]= self.tx
                  Hmatrix[n:n+2,njp:njp+2]= self.ty
                  Hmatrix[n:n+2,njm:njm+2]= tyd
              elif (i==self.Nx-1 and j>0 and j<self.Ny-1):
                  Hmatrix[n:n+2,nim:nim+2]= txd
                  Hmatrix[n:n+2,njp:njp+2]= self.ty
                  Hmatrix[n:n+2,njm:njm+2]= tyd
              elif (j==0 and i>0 and i<self.Nx-1):
                  Hmatrix[n:n+2,nip:nip+2]= self.tx
                  Hmatrix[n:n+2,nim:nim+2]= txd
                  if(j<self.Ny-1):
                      Hmatrix[n:n+2,njp:njp+2]= self.ty
              elif (j==self.Ny-1 and i>0 and i<self.Nx-1):
                  Hmatrix[n:n+2,nip:nip+2]= self.tx
                  Hmatrix[n:n+2,nim:nim+2]= txd
                  Hmatrix[n:n+2,njm:njm+2]= tyd

      self.Hmatrix= Hmatrix
```

In the function MakeDomainWall we introduce one and two domain walls in the magnet. More than two domain walls are not implemented yet. This class is mostly used in chapter 9.

```
def MakeDomainWall(self, fieldstrength=None):
    """
    Purpose: Makes a domain wall (aligned along the y-axis) in the ferromagnet. Which is one part of the H-Matrix
    Psi: one of the angles of rotating magnetisation (Psi=+-pi/2 is a NÃ©el wall; Psi=0 or Psi=pi is a Bloch wall)
    Delta0_wall: "thickness" of wall
    x_wall_offset: offset to a domain wall positioned exactly in the middle
    r: direction in which wall "rolls"
    author: Lucas Goerzen, Jonas Nothhelfer
    """
    h= np.zeros(3*self.Nx*self.Ny,dtype=complex)
    x1= self.randPointsX
    x2= self.Nx - self.randPointsX
    y1= self.randPointsY
    y2= self.Ny - self.randPointsY

    VX= np.zeros(shape=(self.Nx,self.Ny),dtype=complex)
    VY= np.zeros(shape=(self.Nx,self.Ny),dtype=complex)
    VZ= np.zeros(shape=(self.Nx,self.Ny),dtype=complex)

    x_wall_current_location = 0.5*self.Nx + np.array(self.x_wall_location)

    if fieldstrength==None:
        fieldstrength=self.h0

    if len(self.x_wall_location)==1:
        def Theta_func(x):
            return 2*self.r[0]*np.arctan(np.exp(x[0]/self.Delta0_wall[0]))
    elif len(self.x_wall_location)==2:
        def Theta_func(x):
            return abs(2*(self.r[0]*np.arctan(np.exp(x[0]/self.Delta0_wall[0]))+self.r[1]*np.arctan(np.exp(x[1]/self.Delta0_wall[1]))))
    else:
        print("Not implemented!")
        exit()

    for i in range(x1, x2):
        for j in range(y1, y2):
            nh = 3*i + 3*j*self.Nx
            x_distance = i - x_wall_current_location
            h[nh] = fieldstrength*np.sin(Theta_func(x_distance))*np.cos(self.Psi)
            h[nh+1] = fieldstrength*np.sin(Theta_func(x_distance))*np.sin(self.Psi)
            h[nh+2] = fieldstrength*np.cos(Theta_func(x_distance))
            VX[i,j]= h[nh]
            VY[i,j]= h[nh+1]
            VZ[i,j]= h[nh+2]

    self.h= h
    self.VX=VX
    self.VY=VY
    self.VZ=VZ
    self.x_wall_current_location=x_wall_current_location
```

In the function `MakeSkyrmTextures` we introduce single and multiple Skyrmions to the magnet by adding their profiles. As for the vortices we extract from the spatial positions of the Skyrmions labelled by x in the parameter `self.Srx` the number of Skyrmions.

```
def MakeSkyrmTextures(self,eta=1):
        """
        Purpose: Make a Skyrmion texture. Which is a part of the H matrix
        If thetaSkyrm=0, the function returns Neel Skyrmion, if thetaSkyrm=pi/2 the function return a Bloch Skyrmion.
        """

        x1= self.randPointsX
        x2= self.Nx - self.randPointsX
        y1= self.randPointsY
        y2= self.Ny - self.randPointsY

        #size1= x2 - x1
        #size2= y2 - y1

        #l1= A #A*np.double(min([size1, size2]))

        h= np.zeros(3*self.Nx*self.Ny)

        VX= np.zeros(shape=(self.Nx,self.Ny),dtype=complex)
        VY= np.zeros(shape=(self.Nx,self.Ny),dtype=complex)
        VZ= np.zeros(shape=(self.Nx,self.Ny),dtype=complex)


        for i in range(x1,x2):
            for j in range(y1,y2):
                nh= 3*i + 3*j*self.Nx
                xlist=[]
                ylist=[]
                distancelist=[]
                for k in range(len(self.Srx)):
                    xlist.append(np.double(i-self.Srx[k]))
                    ylist.append(np.double(j-self.Sry[k]))
                    distancelist.append(np.sqrt((i-self.Srx[k])**2+(j-self.Sry[k])**2))
                x = xlist[np.argmin(distancelist)]
                y = ylist[np.argmin(distancelist)]
                r= np.sqrt(x**2 + y**2)
                mxy= np.sin(self.Sp*4*np.arctan(np.exp(r/eta)))
                mz= -self.Sq*np.cos(self.Sp*4*np.arctan(np.exp(r/eta)))
                h[nh]= h[nh]+ np.sin(self.Sw*np.arctan2(x,y)+self.thetaSkyrm+np.pi)*mxy
                h[nh+1]= h[nh+1]+ np.cos(self.Sw*np.arctan2(x,y)+self.thetaSkyrm+np.pi)*mxy
                h[nh+2]= h[nh+2]+ mz
                VX[i,j]= h[nh]
                VY[i,j]= h[nh+1]
                VZ[i,j]= h[nh+2]

        self.h= self.h0*h#/len(self.Srx)
        self.VX=VX#/len(self.Srx)
        self.VY=VY#/len(self.Srx)
        self.VZ=VZ#/len(self.Srx)
```

The function MakeEnergyMatrix merges the three Hamiltonian parts.

```
    def MakeEnergyMatrix(self):
        """
        Combine the whole matrix
        """

        N = len(self.Hmatrix[0,:])
        EnergyMatrix = np.zeros(shape=(2*N,2*N),dtype= complex)
        EnergyMatrix[0:N,0:N] = self.Hmatrix
        EnergyMatrix[0:N,N:2*N] = self.DeltaMatrix
        EnergyMatrix[N:2*N,0:N] = -np.conj(self.DeltaMatrix)
        EnergyMatrix[N:2*N,N:2*N] = -np.conj(self.Hmatrix)

        self.EnergyMatrix= EnergyMatrix
```

C. Calculating the Pairing Potential

The self-consistency condition defined in Equation (8.14) is implemented in class `PairingPotential`.

```
def PairingPotential(self):
    """
    Purpose:
    The function calculates the pairing potential.
    Definitions:
    EigenVectors/values: contains all the eigenvectors and (positive) eigenvalues of the BdG equation.
    g: attractive on-site e-e coupling strength (g > 0).
    thermE: thermal energy.
    epsilonD: cut-off energy ~ Debye energy.
    """

    delta= np.zeros(self.Nx*self.Ny,dtype=complex)
    Ne= len(self.EigenValuesP)

    for k in range(Ne):
        etemp= self.EigenValuesP[k]
        if (etemp > self.epsilonD):
            break
        if (np.real(etemp) < 1e-13):
            etemp = 1.j*np.imag(etemp)
        if (np.imag(etemp) < 1e-13):
            etemp = np.real(etemp)
        etemp= etemp/self.thermE

        fermi= 1/(np.exp(np.real(etemp)) + 1)
        fermi= 2*fermi - 1
        for i in range(self.Nx):
            for j in range(self.Ny):
                n= 2*i + 2*j*self.Nx
                nd= i + j*self.Nx
                prod= self.EigenVectorsU[n+1,k]*np.conj(self.EigenVectorsV[n,k]) - self.EigenVectorsU[n,k]*np.conj(self.EigenVectorsV[n+1,k])
                delta[nd]= delta[nd] + 0.5*self.g*prod*fermi

    self.deltan= delta
```

D. Calculate Eigenvalues

The function `CalculateEigenValues` calculates the eigensystem of a Matrix on a graphic card using the python module cupy. This function speeded up the calculations by a factor of 4 compared to calculating it on a cpu because of its high parallelization.

```
def CalculateEigenValues(Matrix):

    Matrix.MakeEnergyMatrix()
    a_gpu = cp.asarray(Matrix.EnergyMatrix)
    [w_gpu, v_gpu] = cp.linalg.eigh(a_gpu)
    Matrix.EigenValues = cp.asnumpy(w_gpu)
    Matrix.EigenVectors = cp.asnumpy(v_gpu)
    Matrix.SortEigVecVal()
```

B.5. Sample Scripts using the Modules

In this section, we show sample scripts that were used to obtain the results of this thesis.

A. The Kitaev Chain

This sample script is mostly used in Section 7.2 and in Section 7.4. Here we define the Matrix which incorporates a Kitaev chain and then we recalculate the eigensystem while rotating the superconducting phase by 180 degrees.

```
# -*- coding: utf-8 -*-
"""
@author: Jonas Nothhelfer
"""
import numpy as np

from SCF_BDG import MakeMatrices_toymodel, Plot, RelaxGPU

Matrix=MakeMatrices_toymodel.MakeMatrix()
p = Plot.Plot()
p.Matrix = Matrix
p.ClearFolder()

# ======================================================================
#Initialize Matrix

#Set max iterations and error (in pairing potential):
Matrix.Nmax=30
Matrix.Maxerror= 0.00001

#System size parameters:
Matrix.Nx=20
Matrix.Ny=1

#System parameters (dimensionless):
Matrix.mu=-10 #setting chemical potential outside of Kitaev chain
mu = 1.5*np.ones(int(Matrix.Nx),dtype=complex) #setting chemical porential of Kitaev chain
Matrix.thermE= 0.001 #thermal energy, kB*T
Matrix.epsilonD= 2 #Debye frequency

#Initialize pairing potential:
Matrix.g= 5.0 #attractive coupling strength
Matrix.deltax = -np.ones(Matrix.Nx*Matrix.Ny,dtype=complex) #Homogenous order-parameter field
Matrix.deltay = 0*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex)

#Make constant matrices:
Matrix.MakeConstantMatrices()
Matrix.MakeKitaevChain(mu)
Matrix.MakeDeltaMatrix()
Matrix.MakeEnergyMatrix()

# ========================================================================
# Relax and save result

RelaxGPU.CalculateEigenValues(Matrix)

#Print the lowest energy states
N_2=int(len(Matrix.EigenValues)/2)
print(’\n’)
print(’lowest energy values:\n’)
print(np.real(Matrix.EigenValues[N_2-5:N_2+5]))
print(’\n’)

vmOld= Matrix.EigenVectors[:,N_2-1]#this is f^dagger (the hole wavefunction).
vpOld= Matrix.EigenVectors[:,N_2]#this is f (the particle wavefunction).

#Save data:
p.DensityPlotMajoranasCombined(0,0.5,’MajoranaCombined_0.png’)
Matrix.Save("Matrix0")

#turning the superconducting phase by 180 degree
for i in range(361):
    print(i)
    Matrix.deltax=np.exp(1.j*i/360*np.pi)*np.ones(Matrix.Nx*Matrix.Ny, dtype=complex)
    Matrix.MakeDeltaMatrix()
    RelaxGPU.CalculateEigenValues(Matrix)
    p.DensityPlotMajoranasCombined(0,0.5,’MajoranaCombined_%d.png’%(i))
    p.VectorPlotDeltaPWave(’Phase_%d.png’%(i))
    Matrix.Save("Matrix%d"%(i))
```

B. T-Junction

This sample script is mostly used in section 7.5. Here we define the matrix which incorporates a T-junction and then we move the Majorana modes inside this T-junction by modifying the chemical potential at different sites of the T-junction.

```
# -*- coding: utf-8 -*-
"""
@author: Jonas Nothhelfer
"""
import numpy as np

from SCF_BDG import MakeMatrices_toymodel, Plot, RelaxGPU

Matrix=MakeMatrices_toymodel.MakeMatrix()
p = Plot.Plot()
p.Matrix = Matrix
p.ClearFolder()

# ========================================================================
#Initialize Matrix

#Set max iterations and error (in pairing potential):
```

```
Matrix.Nmax=30
Matrix.Maxerror= 0.00001

#System size parameters:
Matrix.Nx=19
Matrix.Ny=11

#System parameters (dimensionless):
Matrix.mu=-10 #setting chemical potential outside of T-junction
mu1 = -0.7*np.ones(int(Matrix.Nx/2)-1,dtype=complex)
mu2 = -0.7*np.ones(int(Matrix.Nx/2)-1,dtype=complex)
mu3 = -10*np.ones(int(Matrix.Ny)-2,dtype=complex)
mu3[0]= -0.7 #crosspoint of T-junction
Matrix.thermE= 0.001 #thermal energy, kB*T
Matrix.epsilonD= 2 #Debye frequency

#Initialize pairing potential:
Matrix.g= 5.0 #attractive coupling strength
Matrix.deltax = 1*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex) #Homogenous order-parameter field
Matrix.deltay = 1j*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex)

#Make constant matrices:
Matrix.MakeConstantMatrices()

Matrix.MakeTJunctionHMatrix(mu1,mu2,mu3)
Matrix.MakeDeltaMatrix()
Matrix.MakeEnergyMatrix(eps=1)

# ========================================================================
# Relax and save result

RelaxGPU.CalculateEigenValues(Matrix)

#Print the lowest energy states
N_2=int(len(Matrix.EigenValues)/2)
print(’\n’)
print(’lowest energy values:\n’)
print(np.real(Matrix.EigenValues[N_2-5:N_2+5]))
print(’\n’)

#Save data:
p.VectorPlotDeltaPWave(’VectorDelta.pdf’)
p.DensityPlotMajoranasCombined(0,0.5,’MajoranaCombined_0.png’)
Matrix.Save("Matrix0")
p.VectorPlotDeltaPWave(’InitialPWave.png’)

#Movement of the Majorana modes
for i in range(77):
    print(i)
    if (i>0-1 and i<8):
        mu1[i]=-10
        Matrix.MakeTJunctionHMatrix(mu1,mu2,mu3)
        RelaxGPU.CalculateEigenValues(Matrix)
        p.DensityPlotMajoranasCombined(0,0.5,’MajoranaCombined_%d.png’%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
    if (i>8-1 and i<15):
        mu3[i-7]=-0.7 # begin with i-7=1 because the connection point is already topological
        Matrix.MakeTJunctionHMatrix(mu1,mu2,mu3)
        RelaxGPU.CalculateEigenValues(Matrix)
        p.DensityPlotMajoranasCombined(0,0.5,'MajoranaCombined_%d.png'%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
    if (i>15-1 and i<23):
        mu2[22-i]=-10
        Matrix.MakeTJunctionHMatrix(mu1,mu2,mu3)
        RelaxGPU.CalculateEigenValues(Matrix)
        p.DensityPlotMajoranasCombined(0,0.5,'MajoranaCombined_%d.png'%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
    if (i>23-1 and i<39):
        Matrix.deltay=1j*np.exp(1.j*(i-23)/15*np.pi)*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex)
        Matrix.MakeDeltaMatrix()
        RelaxGPU.CalculateEigenValues(Matrix)
        p.VectorPlotDeltaPWave('MajoranaCombined_%d.png'%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
    if (i>39-1 and i<47):
        mu1[46-i]=-0.7
        Matrix.MakeTJunctionHMatrix(mu1,mu2,mu3)
        RelaxGPU.CalculateEigenValues(Matrix)
        p.DensityPlotMajoranasCombined(0,0.5,'MajoranaCombined_%d.png'%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
    if (i>47-1 and i<54):
        mu3[54-i]=-10 # last entry 54-i=1 since the connection point should stay topological
        Matrix.MakeTJunctionHMatrix(mu1,mu2,mu3)
        RelaxGPU.CalculateEigenValues(Matrix)
        p.DensityPlotMajoranasCombined(0,0.5,'MajoranaCombined_%d.png'%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
    if (i>54-1 and i<62):
        mu2[i-54]=-0.7
        Matrix.MakeTJunctionHMatrix(mu1,mu2,mu3)
        RelaxGPU.CalculateEigenValues(Matrix)
        p.DensityPlotMajoranasCombined(0,0.5,'MajoranaCombined_%d.png'%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
    if (i>62-1 and i<76):
        Matrix.deltay=-1j*np.exp(1.j*(62-i)/15*np.pi)*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex)
        Matrix.MakeDeltaMatrix()
        RelaxGPU.CalculateEigenValues(Matrix)
        p.VectorPlotDeltaPWave('MajoranaCombined_%d.png'%(i+1))
        Matrix.Save("Matrix%d"%(i+1))
```

C. Interaction of Domain Walls and Superconducting Vortices

This sample script is used mostly in Chapter 9. Here we define the matrix such that it incorporates a domain wall in the magnet with a separation from a Vortex in the superconductor. We move the domain wall towards the Vortex and relax the system.

```
# -*- coding: utf-8 -*-
"""
```

```
@author: Jonas Nothhelfer
"""
import numpy as np
import cupy as cp

from SCF_BDG import MakeMatrices, Plot, RelaxGPU

cp.cuda.Device(0).use()

Matrix=MakeMatrices.MakeMatrix()
p = Plot.Plot()
p.ClearFolder()
p.Matrix=Matrix

# =========================================================================
#Initialize Matrix

#Set max iterations and error (in pairing potential):
Matrix.Nmax=200
Matrix.Maxerror= 0.0001

#System size parameters:
Matrix.Nx=53 #should be an odd (prime) number
Matrix.Ny=33 #should be an odd (prime) number

#System parameters (dimensionless):
Matrix.alphaR= 0.75 # Rashba SOC
Matrix.mu= -4 # Chemical potential
Matrix.h0= 0.6 # strength of exchange field
Matrix.thermE= 0.001 #thermal energy, kB*T
Matrix.epsilonD= 100 #Debye frequency

#Make magnetic Skyrmion texture:
Matrix.Sq=1
Matrix.Sw=1
Matrix.Sp=1
NxMid= int((Matrix.Nx-1)/2)
NyMid= int((Matrix.Ny-1)/2)
Matrix.Srx= [NxMid+5,NxMid-5]#[NxMid-2,NxMid+2]# center of Skyrmion
Matrix.Sry= [NyMid,NyMid]#[NyMid-2,NyMid+2]
Matrix.randPointsX= 0#distance from boundary of sample to ferromagnetic block
Matrix.randPointsY= 0
Matrix.thetaSkyrm= 0*np.pi/2 # thetaSkyrm= 0 (Neel Skyrmion); thetaSkyrm= pi/2 (Bloch Skyrmion)
Matrix.x_wall_location=[-19]
Matrix.Psi=0
Matrix.r=[1]
Matrix.Delta0_wall=[3]
Matrix.MakeDomainWall(fieldstrength=0.7*Matrix.h0)

p.ExportMagnetisationProfileVTK()

#Initialize pairing potential:
Matrix.g= 5.0 #attractive coupling strength
Matrix.delta= 1*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex) #Homogenous order-parameter field
Matrix.Vq=1.0 #vorticity
Matrix.thetaVortex= 0*np.pi/2  # thetaVortex= 0 ("Neel" Vortex); thetaVortex= pi/2 ("Bloch" Vortex)
Matrix.Vrx= [NxMid] #Position of Vortex
Matrix.Vry= [NyMid]
Matrix.MakeVortices() #Insert a Vortex in the pair potential
Matrix.MakeDeltaMatrix()

#Make constant matrices:
Matrix.MakeConstantMatrices()
Matrix.MakeHMatrix()

# ========================================================================
# Relax and save result

maxit = RelaxGPU.RelaxPlot(Matrix,p,0,vortex=1,Rnorm=0.75)
Matrix.Save("Matrix"+str(0))
p.VectorPlotDeltaLiveinPlots(i=0, vortex=1, domain_wall=True)

#Move domain wall towards Vortex
for i in range(1,38):
  Matrix.x_wall_location=[-19+i]
  Matrix.MakeDomainWall(fieldstrength=0.7*Matrix.h0)
  Matrix.MakeHMatrix()
  maxit=RelaxGPU.RelaxPlot(Matrix,p,maxit,vortex=1,Rnorm=0.75)
  Matrix.Save("Matrix"+str(i))
  p.VectorPlotDeltaLiveinPlots(i=i, vortex=1, domain_wall=True, current=True)
```

D. Skyrmion Vortex-Pairs

This sample script is mostly used in Chapter 10. Here we define the Hamiltonian such that it incorporates Néel Skyrmions which are on the same position as the superconducting vortices. We move the Skyrmions in a circle until the positions of the two Skyrmions have been exchanged. This path we divide in 360 steps and after moving the Skyrmion each step we let the system relax so that the Vortex can follow this motion.

```
# -*- coding: utf-8 -*-
"""
@author: Jonas Nothhelfer
"""
import numpy as np
import cupy as cp

from SCF_BDG import MakeMatrices, Plot, RelaxGPU

cp.cuda.Device(0).use()

Matrix=MakeMatrices.MakeMatrix()
p = Plot.Plot()
p.ClearFolder()
p.Matrix=Matrix

# ========================================================================
#Initialize Matrix

#Set max iterations and error (in pairing potential):
Matrix.Nmax=200
Matrix.Maxerror= 0.0001

```

```python
#System size parameters:
Matrix.Nx=27 #should be an odd (prime) number
Matrix.Ny=27 #should be an odd (prime) number

#System parameters (dimensionless):
Matrix.alphaR= 0.75 # Rashba SOC
Matrix.mu= -4# Chemical potential
Matrix.h0= 0.7 # strength of exchange field
Matrix.thermE= 0.001 #thermal energy, kB*T
Matrix.epsilonD= 100 #Debye frequency

#Make magnetic Skyrmion texture:
Matrix.Sq=1
Matrix.Sw=1
Matrix.Sp=1
NxMid= int((Matrix.Nx-1)/2)
NyMid= int((Matrix.Ny-1)/2)
Matrix.Srx= [NxMid+5,NxMid-5]# center of Skyrmion
Matrix.Sry= [NyMid,NyMid]
Matrix.randPointsX= 0 #distance from boundary of sample to ferromagnetic block
Matrix.randPointsY= 0
Matrix.thetaSkyrm= 0*np.pi/2 # thetaSkyrm= 0 (Neel Skyrmion); thetaSkyrm= pi/2 (Bloch Skyrmion)
Matrix.MakeSkyrmTextures() # for homogeneous magnetization Matrix.MakeFerroBlock()

#Initialize pairing potential:
Matrix.g= 5.0 #attractive coupling strength
Matrix.delta= 1*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex) #Homogenous order-parameter field
Matrix.Vq=1.0 #vorticity
Matrix.thetaVortex= 0*np.pi/2  # thetaVortex= 0 ("Neel" Vortex); thetaVortex= pi/2 ("Bloch" Vortex)
Matrix.Vrx= [NxMid+5,NxMid-5]#,NxMid+3] #Position of Vortex
Matrix.Vry= [NyMid,NyMid]#,NyMid+3]
Matrix.MakeVortices() #Insert a Vortex in the pair potential

#Make constant matrices:
Matrix.MakeConstantMatrices()
Matrix.MakeHMatrix()

# ========================================================================
# Relax and save result

maxit = RelaxGPU.RelaxPlot(Matrix,p,0,vortex=1,Rnorm=0.9)
Matrix.Save("Matrix"+str(0))
p.DensityPlotMajoranasCombined(0,'MajoranaCombined_%d.svg'%(0+1))

Matrix.Nmax=30

#Braiding of Skyrmions
for i in range(1,361):
    Matrix.Srx[0] = NxMid+5*np.cos(i/360*np.pi)
    Matrix.Sry[0] = NyMid+5*np.sin(i/360*np.pi)
    Matrix.Srx[1] = NxMid+5*np.cos(i/360*np.pi+np.pi)
    Matrix.Sry[1] = NyMid+5*np.sin(i/360*np.pi+np.pi)
    Matrix.MakeSkyrmTextures()
    Matrix.MakeHMatrix()
    maxit = RelaxGPU.RelaxPlotFixSwave(Matrix,p,maxit,vortex=1,Rnorm=0.9,n
        =[1,1,1,1])
    Matrix.Save("Matrix"+str(i))
    p.DensityPlotMajoranasCombined(0,'MajoranaCombined_%d.svg'%(i+1))
```

Bibliography

[1] P. Shiu, The Mathematical Gazette **91**, 453 (2007).

[2] L. Euler, Novi Commentarii academiae scientiarum Petropolitanae **4**, 109 (1758).

[3] P. S. Aleksandrov, Russian Mathematical Surveys **27**, 157 (1972).

[4] E. Gibney and D. Castelvecchi, Nature **538**, 18 (2016).

[5] E. Maxwell, Physics Today **5**, 14 (1952).

[6] A. D. Cobb, Philosophy of Science **76**, 624 (2009).

[7] A. Kosevich, B. Ivanov, and A. Kovalev, Physics Reports **194**, 117 (1990).

[8] T. H. R. Skyrme, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **260**, 127 (1961).

[9] T. Skyrme, Nuclear Physics **31**, 556 (1962).

[10] S. Mühlbauer, B. Binz, F. Jonietz, C. Pfleiderer, A. Rosch, A. Neubauer, R. Georgii, and P. Böni, Science **323**, 915 (2009).

[11] X. Z. Yu, Y. Onose, N. Kanazawa, J. H. Park, J. H. Han, Y. Matsui, N. Nagaosa, and Y. Tokura, Nature **465**, 901 (2010).

[12] N. Romming, C. Hanneken, M. Menzel, J. E. Bickel, B. Wolter, K. von Bergmann, A. Kubetzka, and R. Wiesendanger, Science **341**, 636 (2013).

[13] K. Everschor-Sitte, J. Masell, R. M. Reeve, and M. Kläui, Journal of Applied Physics **124**, 240901 (2018).

[14] S. Heinze, K. von Bergmann, M. Menzel, J. Brede, A. Kubetzka, R. Wiesendanger, G. Bihlmayer, and S. Blügel, Nature Physics **7**, 713 (2011).

[15] J. Masell and K. Everschor-Sitte, in *Topics in Applied Physics* (Springer International Publishing, 2021) pp. 147–181.

[16] H. K. Onnes, in *Through Measurement to Knowledge* (Springer Netherlands, 1991) pp. 261–263.

[17] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, Physical Review **108**, 1175 (1957).

[18] A. Y. Kitaev, Physics-Uspekhi **44**, 131 (2001).

[19] A. Kitaev, Annals of Physics **303**, 2 (2003).

[20] J. Alicea, Reports on Progress in Physics **75**, 076501 (2012).

[21] L. Fu and C. L. Kane, Physical Review Letters **100**, 096407 (2008).

[22] J. Alicea, Y. Oreg, G. Refael, F. von Oppen, and M. P. A. Fisher, Nature Physics **7**, 412 (2011).

[23] J. D. Sau, R. M. Lutchyn, S. Tewari, and S. D. Sarma, Physical Review Letters **104**, 040502 (2010).

[24] J. Alicea, Y. Oreg, G. Refael, F. von Oppen, and M. P. A. Fisher, Nature Physics **7**, 412 (2011).

[25] T.-P. Choy, J. M. Edge, A. R. Akhmerov, and C. W. J. Beenakker, Physical Review B **84**, 195442 (2011).

[26] J. N. Rjabinin and L. W. Shubnikow, Nature **135**, 581 (1935).

[27] A. I. Buzdin, Reviews of Modern Physics **77**, 935 (2005).

[28] D. Rodrigues, J. Nothhelfer, M. Mohseni, R. Knapman, P. Pirro, and K. Everschor-Sitte, Physical Review Applied **16**, 014020 (2021).

[29] D. A. Ivanov, Physical Review Letters **86**, 268 (2001).

[30] J. Nothhelfer, K. M. D. Hals, K. Everschor-Sitte, and M. Rizzi, European Patent Application EP3751472A1 / International Patent Application WO2020EP66120 / Further Country Applications US020220261678A1 JP002022536714A CN000114008639A (2019).

[31] J. Nothhelfer, Master thesis, Johannes Gutenb. Univ. Mainz (2019).

[32] J. M. D. Coey and S. S. Parkin, *Handbook of Magnetism and Magnetic Materials* (Springer International Publishing, 2021).

[33] R. S. Alex Hubert, *Magnetic Domains* (Springer Berlin Heidelberg, 1998).

[34] K. M. Hals and K. Everschor-Sitte, Physical Review Letters **119**, 127203 (2017).

[35] V. P. Kravchuk, D. D. Sheka, U. K. Rößler, J. van den Brink, and Y. Gaididei, Physical Review B **97**, 064403 (2018).

[36] I. Dzyaloshinsky, Journal of Physics and Chemistry of Solids **4**, 241 (1958).

[37] T. Moriya, Physical Review **120**, 91 (1960).

[38] L. Landau and E. Lifshitz, in *Perspectives in Theoretical Physics* (Elsevier, 1992) pp. 51–65.

[39] T. Gilbert, IEEE Transactions on Magnetics **40**, 3443 (2004).

[40] Micromagnum, `https://github.com/MicroMagnum/MicroMagnum` (2022), accessed 18-04-2022.

[41] M. J. Donahue and D. G. Porter, Interagency Report **NISTIR 6376** (1999).

[42] A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, and B. V. Waeyenberge, AIP Advances **4**, 107133 (2014).

[43] W. Jiang, G. Chen, K. Liu, J. Zang, S. G. te Velthuis, and A. Hoffmann, Physics Reports **704**, 1 (2017).

[44] N. Nagaosa and Y. Tokura, Nature Nanotechnology **8**, 899 (2013).

[45] M. Ezawa, Physical Review B **83**, 100408 (2011).

[46] G. Chen, S. P. Kang, C. Ophus, A. T. N'Diaye, H. Y. Kwon, R. T. Qiu, C. Won, K. Liu, Y. Wu, and A. K. Schmid, Nature Communications **8**, 15302 (2017).

[47] P. Milde, D. Köhler, J. Seidel, L. M. Eng, A. Bauer, A. Chacon, J. Kindervater, S. Mühlbauer, C. Pfleiderer, S. Buhrandt, C. Schütte, and A. Rosch, Science **340**, 1076 (2013).

[48] R. Knapman, D. R. Rodrigues, J. Masell, and K. Everschor-Sitte, Journal of Physics D: Applied Physics **54**, 404003 (2021).

[49] J. Müller, J. Rajeswari, P. Huang, Y. Murooka, H. M. Rønnow, F. Carbone, and A. Rosch, Physical Review Letters **119**, 137201 (2017).

[50] R. Zarzuela, V. K. Bharadwaj, K.-W. Kim, J. Sinova, and K. Everschor-Sitte, Physical Review B **101**, 10.1103/physrevb.101.054405 (2020).

[51] S. A. Meynell, M. N. Wilson, K. L. Krycka, B. J. Kirby, H. Fritzsche, and T. L. Monchesky, Physical Review B **96**, 054402 (2017).

[52] A. Fert, V. Cros, and J. Sampaio, Nature Nanotechnology **8**, 152 (2013).

[53] O. Boulle, J. Vogel, H. Yang, S. Pizzini, D. de Souza Chaves, A. Locatelli, T. O. Menteş, A. Sala, L. D. Buda-Prejbeanu, O. Klein, M. Belmeguenai, Y. Roussigné, A. Stashkevich, S. M. Chérif, L. Aballe, M. Foerster, M. Chshiev, S. Auffret, I. M. Miron, and G. Gaudin, Nature Nanotechnology **11**, 449 (2016).

[54] C. Moreau-Luchaire, C. Moutafis, N. Reyren, J. Sampaio, C. A. F. Vaz, N. V. Horne, K. Bouzehouane, K. Garcia, C. Deranlot, P. Warnicke, P. Wohlhüter, J.-M. George, M. Weigand, J. Raabe, V. Cros, and A. Fert, Nature Nanotechnology **11**, 444 (2016).

[55] R. Tomasello, E. Martinez, R. Zivieri, L. Torres, M. Carpentieri, and G. Finocchio, Scientific Reports **4**, 06784 (2014).

[56] M. Mochizuki, Physical Review Letters **108**, 017601 (2012).

[57] V. P. Kravchuk, D. D. Sheka, U. K. Rößler, J. van den Brink, and Y. Gaididei, Physical Review B **97**, 064403 (2018).

[58] M. Garst, J. Waizner, and D. Grundler, Journal of Physics D: Applied Physics **50**, 293002 (2017).

[59] C. Schütte and M. Garst, Physical Review B **90**, 094423 (2014).

[60] D. R. Rodrigues, A. Abanov, J. Sinova, and K. Everschor-Sitte, Physical Review B **97**, 134414 (2018).

[61] D. Quievy and F. Desjouis, US4967160A - Frequency multiplier with programmable order of multiplication - Google Patents — patents.google.com, `https://patents.google.com/patent/US4967160A/de` (1990), accessed 29-06-2023.

[62] H. B. Sanderford Jr, USRE35829E - Binary phase shift keying modulation system and/or frequency multiplier - Google Patents — patents.google.com, `https://patents.google.com/patent/USRE35829E/en` (1998), [Accessed 29-Jun-2023].

[63] L. Kool, W. H. DeGuelle, and J. Y. Tichelaar, US5903609A - Transmission system using transmitter with phase modulator and frequency multiplier - Google Patents — patents.google.com, `https://patents.google.com/patent/US5903609A/en` (1999), accessed 29-06-2023.

[64] R. Salvi, R. Ponce, and A. J. Gonzalez, CA2231636A1 - Self centering frequency multiplier - Google Patents — patents.google.com, `https://patents.google.com/patent/CA2231636A1/en` (1999), accessed 29-06-2023.

[65] G. Chien and P. Gray, in *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No.00CH37056)* (IEEE, 2020) p. 839749.

[66] G. Chattopadhyay, E. Schlecht, J. Ward, J. Gill, H. Javadi, F. Maiwald, and I. Mehdi, IEEE Transactions on Microwave Theory and Techniques **52**, 1538 (2004).

[67] S. Rohart and A. Thiaville, Physical Review B **88**, 184422 (2013).

[68] B. F. McKeever, D. R. Rodrigues, D. Pinna, A. Abanov, J. Sinova, and K. Everschor-Sitte, Physical Review B **99**, 054430 (2019).

[69] D. Kumar and A. O. Adeyeye, Journal of Physics D: Applied Physics **50**, 343001 (2017).

[70] D. R. Rodrigues, A. Abanov, J. Sinova, and K. Everschor-Sitte, Physical Review B **97**, 134414 (2018).

[71] M. Beg, M. Albert, M.-A. Bisotti, D. Cortés-Ortuño, W. Wang, R. Carey, M. Vousden, O. Hovorka, C. Ciccarelli, C. S. Spencer, C. H. Marrows, and H. Fangohr, Physical Review B **95**, 014433 (2017).

[72] scikit-image: Image processing in Python; scikit-image — scikit-image.org, `https://scikit-image.org/` (2023), accessed 09-03-2023.

[73] A. Dussaux, B. Georges, J. Grollier, V. Cros, A. Khvalkovskiy, A. Fukushima, M. Konoto, H. Kubota, K. Yakushiji, S. Yuasa, K. Zvezdin, K. Ando, and A. Fert, Nature Communications **1**, 1006 (2010).

[74] G. Finocchio, M. Ricci, R. Tomasello, A. Giordano, M. Lanuzza, V. Puliafito, P. Burrascano, B. Azzerboni, and M. Carpentieri, Applied Physics Letters **107**, 262401 (2015).

[75] K. Wagner, A. Kákay, K. Schultheiss, A. Henschke, T. Sebastian, and H. Schultheiss, Nature Nanotechnology **11**, 432 (2016).

[76] S. J. Hämäläinen, M. Madami, H. Qin, G. Gubbiotti, and S. van Dijken, Nature Communications **9**, 41467 (2018).

[77] R. B. Holländer, C. Müller, J. Schmalz, M. Gerken, and J. McCord, Scientific Reports **8**, 41598 (2018).

[78] G. Csaba and W. Porod, Applied Physics Reviews **7**, 011302 (2020).

[79] R. P. Feynman, International Journal of Theoretical Physics **21**, 467 (1982).

[80] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2012).

[81] D. P. DiVincenzo, Science **270**, 255 (1995).

[82] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2012).

[83] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya, Software: Practice and Experience **52**, 66 (2021).

[84] IonQ | Our Trapped Ion Technology — ionq.com, `https://ionq.com/technology` (2023), accessed 23-Jun-2023.

[85] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, B. Höfer, C. Wächter, K. Lakhmanskiy, R. Blatt, P. Schindler, and T. Monz, PRX Quantum **2**, 020343 (2021).

[86] IBM Quantum — quantum-computing.ibm.com, `https://quantum-computing.ibm.com/services/resources` (2023), accessed 23-06-2023.

[87] F. Arute, K. Arya, R. Babbush, and D. B. E. al., Nature **574**, 505 (2019).

[88] S. G. J. Philips, M. T. Mądzik, S. V. Amitonov, S. L. de Snoo, M. Russ, N. Kalhor, C. Volk, W. I. L. Lawrie, D. Brousse, L. Tryputen, B. P. Wuetz, A. Sammak, M. Veldhorst, G. Scappucci, and L. M. K. Vandersypen, Nature **609**, 919 (2022).

[89] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim, Nature Communications **10**, 41467 (2019).

[90] T. Ruster, C. Warschburger, H. Kaufmann, C. T. Schmiegelow, A. Walther, M. Hettrich, A. Pfister, V. Kaushal, F. Schmidt-Kaler, and U. G. Poschinger, Physical Review A **90**, 033410 (2014).

[91] J. K. Pachos, *Introduction to Topological Quantum Computation* (Cambridge University Press, 2012).

[92] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. D. Sarma, Reviews of Modern Physics **80**, 1083 (2008).

[93] V. Lahtinen and J. Pachos, SciPost Physics **3**, 021 (2017).

[94] E. Witten, Communications in Mathematical Physics **121**, 351 (1989).

[95] A. Kitaev, Annals of Physics **321**, 2 (2006).

[96] J. K. Pachos, *Introduction to Topological Quantum Computation* (Cambridge University Press, 2012).

[97] S. B. Bravyi and A. Y. Kitaev, Annals of Physics **298**, 210 (2002).

[98] A. Ahlbrecht, L. S. Georgiev, and R. F. Werner, Physical Review A **79**, 032311 (2009).

[99] S. Kessler, Numerical simulation of quantum operations with majorana bound states (2022).

[100] D. van Delft and P. Kes, Physics Today **63**, 38 (2010).

[101] W. Meissner and R. Ochsenfeld, Die Naturwissenschaften **21**, 787 (1933).

[102] F. London and H. London, Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences **149**, 71 (1935).

[103] V. L. Ginzburg and L. D. Landau, Zh. Eksp. Teor. Fiz. **20**, 1064 (1950).

[104] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, Physical Review **108**, 1175 (1957).

[105] J.-X. Zhu, *Bogoliubov-de Gennes Method and Its Applications* (Springer International Publishing, 2016).

[106] E. Majorana and L. Maiani, Nuovo Cim. (engl. Transl. Soryushiron Kenkyu) **63**, 149 (1981).

[107] K. Wölms, A. Stern, and K. Flensberg, Physical Review Letters **113**, 246401 (2014).

[108] B. Béri and N. R. Cooper, Physical Review Letters **109**, 156803 (2012).

[109] A. Altland and R. Egger, Physical Review Letters **110**, 196401 (2013).

[110] S. Plugge, A. Rasmussen, R. Egger, and K. Flensberg, New Journal of Physics **19**, 012001 (2017).

[111] M. Ezawa, Physical Review B **102**, 075424 (2020).

[112] P. DE GENNES, *Superconductivity of Metals and Alloys*, Advanced Book Classics (Addison-Wesley Publishing Company, 1966).

[113] K. M. Hals, M. Schecter, and M. S. Rudner, Physical Review Letters **117**, 017001 (2016).

[114] M. Ezawa, Physical Review B **102**, 075424 (2020).

[115] S. Rex, I. V. Gornyi, and A. D. Mirlin, Physical Review B **100**, 064504 (2019).

[116] A. Manchon, H. C. Koo, J. Nitta, S. M. Frolov, and R. A. Duine, Nature Materials **14**, 871 (2015).

[117] J. Nothhelfer, S. A. Díaz, S. Kessler, T. Meng, M. Rizzi, K. M. D. Hals, and K. Everschor-Sitte, Physical Review B **105**, 224509 (2022).

[118] T. I. Lee and R. A. Young, Annual Review of Genetics **34**, 77 (2000).

[119] Y. Suzuki, M. Shin, A. Yoshida, S. H. Yoshimura, and K. Takeyasu, FEBS Letters **586**, 3187 (2012).

[120] I. F. Lyuksyutov and V. L. Pokrovsky, Advances in Physics **54**, 67 (2005).

[121] A. Y. Aladyshkin, A. V. Silhanek, W. Gillijns, and V. V. Moshchalkov, Superconductor Science and Technology **22**, 053001 (2009).

[122] L. E. Helseth, P. E. Goa, H. Hauglin, M. Baziljevich, and T. H. Johansen, Physical Review B **65**, 132514 (2002).

[123] J. I. Vestgarden, D. V. Shantsev, A. A. F. Olsen, Y. M. Galperin, V. V. Yurchenko, P. E. Goa, and T. H. Johansen, Physical Review Letters **98**, 117002 (2007).

[124] D. A. Dikin, M. Mehta, C. W. Bark, C. M. Folkman, C. B. Eom, and V. Chandrasekhar, Physical Review Letters **107**, 056802 (2011).

[125] K. Michaeli, A. C. Potter, and P. A. Lee, Physical Review Letters **108**, 117003 (2012).

[126] F. S. Bergeret and I. V. Tokatly, Physical Review Letters **110**, 117003 (2013).

[127] N. Banerjee, J. A. Ouassou, Y. Zhu, N. A. Stelmashenko, J. Linder, and M. G. Blamire, Physical Review B **97**, 184521 (2018).

[128] S. Field, J. Witt, F. Nori, and X. Ling, Physical Review Letters **74**, 1206 (1995).

[129] E. Altshuler and T. H. Johansen, Reviews of Modern Physics **76**, 471 (2004).

[130] F. Laviano, in *Vortices and Nanostructured Superconductors* (Springer International Publishing, 2017) pp. 133–157.

[131] T. Matsuda, O. Kamimura, H. Kasai, K. Harada, T. Yoshida, T. Akashi, A. Tonomura, Y. Nakayama, J. Shimoyama, K. Kishio, T. Hanaguri, and K. Kitazawa, Science **294**, 2136 (2001).

[132] A. Grigorenko, S. Bending, T. Tamegai, S. Ooi, and M. Henini, Nature **414**, 728 (2001).

[133] S. J. Bending and M. J. W. Dodgson, Journal of Physics: Condensed Matter **17**, R955 (2005).

[134] O. V. Dobrovolskiy, Physica C: Superconductivity and its Applications **533**, 80 (2017).

[135] V. Mourik, K. Zuo, S. M. Frolov, S. R. Plissard, E. P. A. M. Bakkers, and L. P. Kouwenhoven, Science **336**, 1003 (2012).

[136] H. Kim, A. Palacio-Morales, T. Posske, L. Rózsa, K. Palotás, L. Szunyogh, M. Thorwart, and R. Wiesendanger, Science Advances **4**, 5251 (2018).

[137] J. Nakamura, S. Liang, G. C. Gardner, and M. J. Manfra, Nature Physics **16**, 931 (2020).

[138] K. Björnson and A. M. Black-Schaffer, Physical Review B **88**, 024501 (2013).

[139] A. O. Leonov, T. L. Monchesky, N. Romming, A. Kubetzka, A. N. Bogdanov, and R. Wiesendanger, New Journal of Physics **18**, 065003 (2016).

[140] C. Beenakker, Search for non-abelian majorana braiding statistics in superconductors (2020).

[141] L. S. Georgiev, Physical Review B **74**, 235112 (2006).

[142] X. Ma, C. J. O. Reichhardt, and C. Reichhardt, Physical Review B **101**, 024514 (2020).

[143] Signal processing (scipy.signal); scipy v1.10.1 manual — docs.scipy.org, https://docs.scipy.org/doc/scipy/reference/signal.html (2023), accessed 23-06-2023.

[144] SCF_BDG, https://github.com/jnothelf/SCF_BDG (2023), accessed 07-04-2023.

Acknowledgements

I would like to take the opportunity to express my deep gratitude and appreciation to all those who have been a part of this remarkable journey of completing my thesis.

First and foremost, I would like to extend my heartfelt gratitude to my parents for being my constant support system throughout my life. Their unwavering love, encouragement, and guidance have been instrumental in shaping me into the person I am today.

I would also like to thank my amazing girlfriend, who has been by my side during this journey, listening to my problems, and providing me with the encouragement and support that I needed. She played a significant role in keeping me motivated during some of the toughest times.

I would also like to extend my heartfelt gratitude to Kai Litzius, who played a significant role in keeping me motivated during moments when I felt my thesis was going nowhere. Our coffee breaks and insightful discussions provided me with a fresh perspective and renewed enthusiasm for my work. Additionally, I am immensely grateful for Kai's encouragement to pursue a patent application for my project, even when I doubted its potential. His belief in the significance of my research empowered me to take bold steps and explore new possibilities. Thank you, Kai, for your invaluable support and unwavering faith in my abilities.

I would like to express my heartfelt gratitude to Alessandro Pignedoli, Philipp Schmoll, Pascal Praß, and Nico Kerber for their invaluable assistance in proofreading my thesis. Their meticulous attention to detail, keen editorial eye, and constructive feedback have significantly enhanced the clarity and coherence of my work.

I am indebted to my classmates, who I had the privilege of going through years of suffering and joy during our studies. I would like to give a special shoutout to Nico Kerber, Pascal Praß and Christian Matthé, who became such good friends of mine, and I hope our friendship lasts a lifetime.

To my friends, who understood me when I went missing for months after they tried to contact me, I thank you. You all have been incredibly patient and understanding, and I appreciate you all for that.

I am grateful to the Max Planck Graduate Center for providing me with the necessary infrastructure and resources that allowed me to establish and maintain meaningful connections with my supervisors and mentors

I would like to express my sincere appreciation to Kjetil Hals, Matteo Rizzi, and Tobias Meng for their invaluable expertise and guidance throughout my thesis journey. Whenever I found myself stuck or in need of new insights, they were always there to lend a helping hand.

Lastly, I would like to express my deep gratitude to my supervisor, Karin Everschor-Sitte, for her unwavering support throughout all life situations. Her guidance and mentorship have been invaluable in shaping both my academic and personal growth. Thanks to her exceptional guidance, I was fortunate enough to secure a permanent job in close proximity to academia even before completing my thesis. Beyond being a supervisor, Karin has also become one of my closest friends. I am truly grateful for her friendship and support, and I hope that our friendship continues in the years to come.

To all of you, I extend my heartfelt appreciation for your love, support, and dedication. Your unwavering support and belief in me to complete this thesis inspired me to keep pushing forward. Thank you all from the bottom of my heart.

Declaration of Authorship

I hereby declare that I wrote the dissertation submitted without any unauthorized external assistance and used only sources acknowledged in the work. All textual passages which are appropriated verbatim or paraphrased from published and unpublished texts as well as all information obtained from oral sources are duly indicated and listed in accordance with bibliographical rules. In carrying out this research, I complied with the rules of standard scientific practice as formulated in the statutes of Johannes Gutenberg-University Mainz to insure standard scientific practice.

Mainz the July 27, 2023

Jonas Nothhelfer