

Master Thesis

**Localized Majorana modes in
superconductor-ferromagnet
heterostructures - A path towards
topological quantum computation**

Jonas Nothhelfer

March 18, 2019



JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

First referee: Dr. Karin Everschor-Sitte and Juniorprof. Dr. Matteo Rizzi
Second referee: Prof. Dr. Mathias Kläui

Roses are red, violets are blue,
particles can obey statistics
ranging continuously between Bose and Fermi
if the spatial dimension is two.

Abelian anyon

$$|\psi_1, \psi_2\rangle = e^{i\pi\theta} |\psi_2, \psi_1\rangle, \theta \neq 0, \pi$$

non-Abelian anyon

$$|\psi_1, \psi_2\rangle = \mathbf{M} |\psi_2, \psi_1\rangle, \mathbf{M} \text{ is a matrix}$$

Abstract

In this thesis we investigate the existence of Majorana zero modes in ferromagnet-superconductor heterostructures. A promising feature of these modes is the property of non-abelian statistics. Such statistics of exchanging two quasiparticles is the key ingredient to building a topological quantum computer which implements quantum error correction on a hardware level. At the moment there is no experimental technique which shows unambiguously the existence of Majorana modes. Hence, the only way to prove if the observed modes are the desired ones is to braid them and verify their non-abelian statistics. Therefore, it is vital to find a system where the latter can be realized. In order to analyse the considered systems, we diagonalize the corresponding Bogoliubov-de Gennes Hamiltonian and self-consistently calculate the superconducting gap. Within this method the energy of the system is minimized. Thereby, we obtain a stable solution, which makes it more probable to be found in an experimental setup.

As a main result, we find the existence of Majorana zero modes in skyrmion-vortex pairs. Due to the drivability of the skyrmion and the attraction between a skyrmion and a vortex, skyrmion-vortex pairs could be a way to actually braid the Majorana zero modes. As a starting point, we prove that the used method fulfils the concepts derived for the Kitaev Chain. In the course of this thesis, we expand the model to skyrmion-vortex pairs. Finally, the generation of effective spin-orbit coupling by a skyrmion with winding number two is an unstable solution in our method.

Contents

| | |
|--|-----------|
| 1. Introduction | 6 |
| 2. Non-Abelian Statistics and the Kitaev Chain | 8 |
| 2.1. Non-Abelian Statistics | 8 |
| 2.2. Kitaev Chain toy model | 10 |
| 2.3. Engineering an experimental platform | 17 |
| 3. Methods | 24 |
| 3.1. Derivation of the Bogoliubov de Gennes Hamiltonian | 24 |
| 3.2. Self-consistency condition | 26 |
| 3.3. Discretization of the model | 27 |
| 3.4. Gap Closing Condition | 29 |
| 4. From Kitaev Chain to Skyrmion-Vortex Pairs | 30 |
| 4.1. Diagonalisation of the Kitaev Chain in Real Space | 30 |
| 4.2. Self consistent calculation of Kitaev Chain | 35 |
| 4.3. Two dimensional ferromagnetic Block on a two dimensional Superconductor | 41 |
| 4.4. Majorana Modes in Vortices | 44 |
| 4.5. Majorana Modes in Skyrmions | 46 |
| 4.6. Majorana Modes in Skyrmion-Vortex Pairs | 50 |
| 5. Conclusion and Outlook | 54 |
| A. Appendix | 57 |
| A.1. Pairing | 57 |
| A.2. Programs to produce Plots in Chapter 2 | 60 |
| A.3. Program for the method in Chapter 3 | 73 |
| A.4. Relaxation | 92 |
| A.5. Procedure for the phase diagrams | 93 |

1

Introduction

In today's society computers are no longer expendable. We use them on a regular basis in our daily lives. But even more importantly, they push scientific and economic progress by simplifying tasks that were impossible before. This is achieved by creating large networks of computers that are linked via the internet. Being such an essential resource, the aim is to speed up and increase the efficiency of these computers leading to an enhancement of the whole network. For many years this acceleration, corresponding to the number of transistors per unit area in an integrated circuit, increased exponentially, which Gordon Moore pointed out in 1965 [1]. But this progress will inevitably saturate if the transistors reach an atomic level.

One way to circumvent this problem is to use a computer which does not use classical physics but quantum physics for information processing. Richard Feynman formulated this idea first in 1981 in the context of simulating a quantum mechanical problem with quantum mechanics instead of classical physics [2]. This approach was not only interesting from a scientific point of view but also from an economical one. The first breakthrough was the quantum algorithm by Peter Shor in 1994 [3]. He showed that a quantum computer is able to find the prime factor of a number in polynomial time. For this reason, the common key encryption methods based on the RSA (Rivest-Shamir-Adleman) algorithm is not safe anymore. Nevertheless, this requires an effective quantum computer we do not have by now. Nowadays quantum computing is expected to have a major impact in other fields i.e. neuronal networks, medicine, and even in logistics.

Because of the variety of possible progresses using quantum computers and other related techniques, the EU has launched a European Flagship program [4] as a large-scale, long-term research initiative, which focusses on four application areas: Quantum communication, quantum simulation, quantum computing, and quantum metrology and sensing. In addition, there is currently a competition driven race between academic and industrial research centers, working on several different physical platforms such as superconducting qubits (IBM [5] and Google [6]), ion traps [7], nitrogen-vacancy centers [8], and quantum dots [9], just to name a few.

A very promising field within the area of quantum computation is the one of topological quantum computation, which circumvents instabilities by exploiting topological properties of condensed matter systems [10, 11]. In comparison to the approaches above, quantum error-correction is implemented at the hardware level; thus, it is resilient

to control errors and erroneous perturbations. This is reached by using quasiparticles, called Anyons, which generalise the statistics observed for fermions and bosons. Certain kinds of these anyons give rise to a decoherence-free subspace, in which the quantum state can be evolved by moving anyons adiabatically around each other.

One possible kind of anyon are Majorana modes, which originally were observed as solutions to the Dirac equation, in which a Majorana particle is equal to its own antiparticle [12]. In 2001, Kitaev proposed the realization of Majorana modes in a p-wave superconductor [13]. Consequently, the search for Majoranas in experimentally observable systems began. In the pioneering work by Kane and Fu, topological insulators were considered to achieve $p_x \pm ip_y$ -wave-like pairing in a s-wave superconductor [14]. Later Sau et al. showed that it is sufficient to use a semiconductor instead [15]. In this case, effective p-wave pairing is obtained either for strong spin orbit coupling combined with a homogeneous magnetic field [16] or varying magnetic moments on top of the superconductor [17]. For these two approaches there are already signatures of Majorana modes shown in [18, 19] respectively. Following this trend, Yang et al. proposed that skyrmions (topologically protected whirls in the magnetisation pattern of specific materials) could induce an effective p-wave pairing in a nearby s-wave superconductor [20]. Therefore one should, in principle, be able to observe Majorana modes localized at the skyrmion. Since it has experimentally already been shown that skyrmions can be displaced by means of electric currents [21, 22], it may therefore be possible to design devices enabling braiding of majorana modes.

In this work, we first propose a system that is feasible for topological quantum computation. Accordingly, in chapter 2 we will follow the history of this subject going from a general toy model to an experimental platform. In chapter 3 we will describe the self-consistent field method to go beyond the pioneering work and study ferromagnet-superconducting heterostructures by considering the superconducting pairing self-consistently. Going on, in chapter 4 we show that the method is consistent with state of the art results, such as localized Majorana modes in the Kitaev chain and in superconducting vortices. Furthermore we find localized Majorana modes in skyrmion-vortex pairs as a new result. Finally, we present the future steps planned for this project in chapter 5.

2

Non-Abelian Statistics and the Kitaev Chain

In this chapter, we first introduce anyons and highlight which properties make them desirable objects for quantum computation. This is followed by the introduction of the Kitaev chain toy model, explaining the fundamental concepts of Majorana zero modes being one realization of anyons that can be used for topological quantum computation. Afterwards, we will show how we can engineer a system where the predictions of the toy model can be observed. This chapter reproduces results from three references [23, 24, 25], from where some of the plots are reproduced using the code shown in A.2.

2.1. Non-Abelian Statistics

In mathematics, topology is the study of the global properties of manifolds that are insensitive to local, smooth deformations (homeomorphisms). An example for this is the topological equivalence between a doughnut and a coffee cup: Here the topological invariant is the hole, which is present in both objects. Since we can smoothly deform the coffee cup into the doughnut without losing the hole throughout the process, it is the same topological object. We can only change the topology of it by cutting the doughnut in half or breaking the handle of the cup, which would not represent a smooth deformation anymore.

The problem is that in current quantum computers small deformations can have strong influences which makes them susceptible for local perturbations. Due to this, one must take exceptional care to ensure that outside noise does not interfere and that the evolution is precisely the desired one, to store and evolve a pure quantum state coherently. Therefore, it would be desirable to store quantum information in topological properties of matter and have quantum gates that are only dependent on the topology of the evolutions. Such an approach would protect the evolution of the quantum state from local perturbations.

This idea was first brought up by Kitaev in connection with surface codes for quantum error correction [26]. He realized that certain codes could be viewed as spin lattice models, where the elementary excitations are anyons – quasiparticles with statistics

interpolating between those of bosons and fermions. For such excitations, the non-contractible paths, which are topologically equivalent, will refer to the same quantum gate. This motivated many people to study condensed matter systems, where such excitations are possible.

The most important requirement which the system should fulfil is the necessary condition which states that the exchange of two identical particles should not change the local physics. In a three dimensional system there is only the possibility of fermions and bosons. This means that the wavefunction describing the system of either type gets -1 or $+1$ as a phase respectively by exchanging two particles. However, in a two dimensional system there is a much bigger variety of statistics possible. In addition to bosonic and fermionic exchange statistics, arbitrary phase factors, or even non-trivial unitary evolutions, can be obtained when two particles are exchanged. 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 2.1.

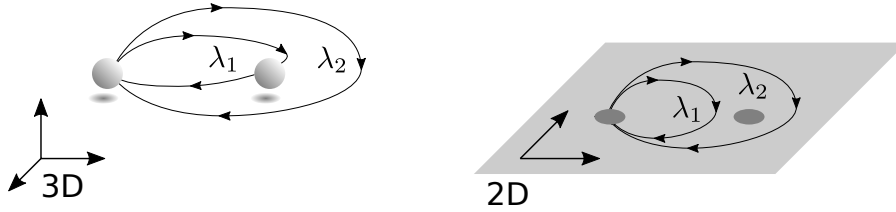


Figure 2.1.: Exchange statistics in 2D vs. 3D. In 3D the path λ_2 describing two particle exchanges is continuously deformable to λ_1 by taking it behind or front of the right-most particle, and in turn λ_1 is contractible to a point. Hence, all the paths have the same topology and thus correspond to 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 topology and different statistical quantum evolutions can be assigned to each.

This results in the condition

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

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 corresponds 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

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

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 first case, the anyons are called Abelian anyons due to their exchange operators commuting, 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. In conclusion, the statistics of two dimensional materials allow anyons to exist.

There are many works that show that the non-Abelian statistics of such anyons can be recovered in condensed matter systems. Here we will focus on two works where the anyons that are considered in these systems are called Majorana zero modes which we will discuss in more detail throughout this thesis. One work considers the Kitaev chain, explained in the following chapters, and shows that a T-shaped configuration of this chain can obey non-Abelian statistics [16]. The other work shows that Majorana zero modes in vortices of a $p_x \pm ip_y$ superconductor also recover non-Abelian statistics [27].

2.2. Kitaev Chain toy model

2.2.1. Spinless p-wave superconductor and Majorana fermions

We will start with the toy model of Kitaev. This model describes a spinless p-wave superconductor given by the Hamiltonian and first introduced in Ref. [13]

$$H_{\text{chain}} = -\mu \sum_{i=1}^N n_i - \sum_{i=1}^{N-1} (t c_i^\dagger c_{i+1} - \Delta c_i c_{i+1} + h.c.) \quad (2.1)$$

where $h.c.$ stands for hermitian conjugate, c_i is the electron annihilation operator for site i , and $n_i = c_i^\dagger c_i$ is the associated number operator. The first term in this Hamiltonian describes the internal energy of the system in terms of the particle number controlled by the chemical potential μ . The second term describes how much energy is needed to move an electron from one lattice site to the neighbouring lattice site. The third term describes the p-wave pairing of two electrons on two neighbouring lattice sites with the pairing strength Δ . A more detailed description of the different pairings can be found in Appendix A.1. The pairing Δ and the hopping t are assumed to be the same for all sites. In the following, we will assume that the superconducting phase ϕ is zero, meaning that we can write $\Delta = |\Delta| e^{i\phi} = |\Delta|$.

In Kitaev's work, he noticed that if one writes the electron creation and annihilation operators as a superposition of two half electrons then different phases, namely a trivial and a topological phase, can be seen in the toy model. By doing that, we will get

$$\begin{aligned} c_i &= \frac{1}{2}(\gamma_{i,1} + i\gamma_{i,2}) \\ c_i^\dagger &= \frac{1}{2}(\gamma_{i,1} - i\gamma_{i,2}), \end{aligned}$$

where $\gamma_{i,\alpha}$ are Majorana operators [12] living on site i . They were found as a solution to Dirac's equation with the special property, that they are their own antiparticles. By inverting this equation, we will receive

$$\begin{aligned} \gamma_{i,1} &= c_i^\dagger + c_i \\ \gamma_{i,2} &= i(c_i^\dagger - c_i) \end{aligned}$$

where we can see that $\gamma_{i,j} = \gamma_{i,j}^\dagger$. Using the fermionic anti-commutation relations for the c_i -fermions, one observes that the Majorana operators satisfy the anti-commutation relation

$$\{\gamma_{i,\alpha}, \gamma_{i,\beta}\} = 2\delta_{ij}\delta_{\alpha\beta}.$$

From this, we see that $\gamma_{i,\alpha}^2 = 1$ which means that there is no Pauli Principle for Majorana fermions. In fact, we cannot even speak of occupancy of a Majorana mode, because if we try to define a Majorana number operator we will always end with $n_i^{\text{MF}} = \gamma_{i,\alpha}^\dagger \gamma_{i,\alpha} = 1$ since they are their own antiparticles. Thus Majorana modes are in a sense always empty as well as always filled and counting does not make any sense.

Now we can use these new operators in order to rewrite the Hamiltonian in equation (2.1) resulting in

$$H_{\text{chain}} = -\frac{1}{2}\mu \sum_{i=1}^N i\gamma_{i,1}\gamma_{i,2} - \frac{1}{2} \sum_{i=1}^{N-1} i[(t - \Delta) \gamma_{i,1}\gamma_{i+1,2} - (t + \Delta) \gamma_{i,2}\gamma_{i+1,1}] - 2. \quad (2.2)$$

In the following, we will ignore the constant at the end since it will only shift our energy spectrum without contributing to the physical understanding. This reformulation of the Hamiltonian can be seen graphically in figure 2.2 (a). Now we will consider two special cases:

First case

Here we will set $\mu \neq 0$ and $\Delta = t = 0$, meaning that we only consider onsite contributions. Then the Hamiltonian in equation (2.2) can be written as

$$H_{\text{chain}} = -\frac{i}{2}\mu \sum_{i=1}^N \gamma_{i,1}\gamma_{i,2} = -\mu \sum_{i=1}^N c_i^\dagger c_i. \quad (2.3)$$

Here the chain has a gapped bulk and no zero energy edge states. We also see that if we pair two Majorana modes on the same site to a fermion, we will get the first term of our Hamiltonian. The pairing can then be represented as seen in figure 2.2 (b).

Second case

In the other special case we set $\mu = 0$ and $\Delta = t \neq 0$, meaning that we only consider contributions connecting different sites, resulting in the following Hamiltonian

$$H_{\text{chain}} = -it \sum_{i=1}^{N-1} \gamma_{i,2} \gamma_{i+1,1} \tag{2.4}$$

Here the Majorana operators $\gamma_{N,2}$ and $\gamma_{1,1}$ which are localized at the ends of the chain are completely missing from (2.4). Again, we want to pair Majorana modes to build

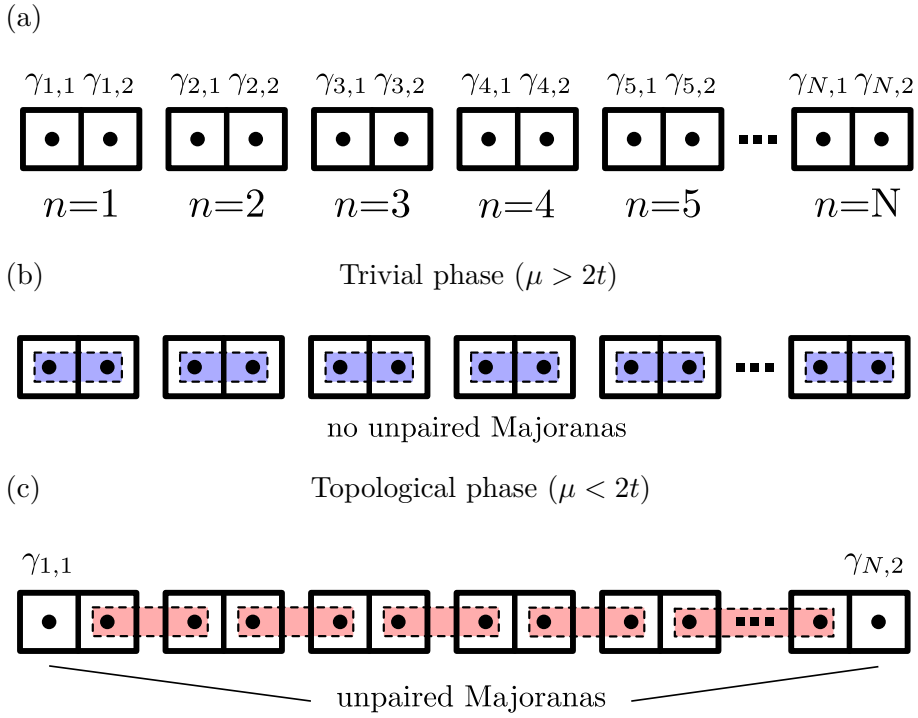


Figure 2.2.: Sketch of the Kitaev Chain. The big rectangles represent an electron and the small blocks with a black circle in the middle represent the Majoranas so that one Majorana is shown as one half of an electron. In (a) the notation for the chain is shown, where γ is identified as a Majorana. In (b) the trivial pairing between the Majoranas is shown according to (2.3). In (c) the nontrivial pairing is shown according to (2.4). Here two unpaired Majoranas appear at the edges of the chain.

fermions. In comparison to (2.3), we will now combine two Majorana operators on neighbouring sites

$$\tilde{c}_i = \frac{1}{2}(\gamma_{i+1,1} + i\gamma_{i,2}).$$

This pairing is demonstrated in figure 2.2 (c). In terms of these new fermions, equation (2.4) will now be

$$H_{\text{chain}} = 2t \sum_{i=1}^{N-1} \tilde{c}_i^\dagger \tilde{c}_i.$$

Thus, \tilde{c}_i are the annihilation operators corresponding to the eigenstates, and the energy cost of creating a \tilde{c}_i fermion is $2t$. These two Majorana operators can equivalently be described by a single fermionic state with operator

$$\tilde{c}_M = \frac{1}{2}(\gamma_{N,2} + i\gamma_{1,1}).$$

This is a highly non-local state since $\gamma_{N,2}$ and $\gamma_{1,1}$ are localized on opposite ends of the chain. Furthermore, since this fermion operator is absent from the Hamiltonian it means that it has zero energy and that the (zero-energy) ground state is two-fold degenerate. It is either occupied or unoccupied.

These two examples show that there are at least two phases in the system. In the trivial phase, two Majorana modes on the same site pair with each other, and in the topological phase, two Majorana modes on different sites pair with each other, where we get localized zero energy modes at the edges of the chain. From the energy of the corresponding modes we see that whenever $\mu > 2t$ the trivial phase will win and when $\mu < 2t$ the topological phase will win. In the next part, we will discuss how this will manifest in the diagonalization of (2.1)

2.2.2. Bogoliubov-de Gennes Hamiltonian

To diagonalize equation (2.1), it is very useful to write the Hamiltonian according to the Bogoliubov de Gennes formalism [28], meaning that our Hamiltonian will be given by $H = \frac{1}{2}C^\dagger H_{\text{BdG}}C$ with C a column vector containing all creation and annihilation operators, $C = (c_1, \dots, c_N, c_1^\dagger, \dots, c_N^\dagger)^T$, 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 $(0, \dots, 1, \dots, 0)^T$ corresponding to the n -th site of the chain. With these definitions the Bogoliubov de Gennes Hamiltonian is then given by

$$H_{\text{BdG}} = -\mu \sum_{n=1}^N \tau_z \otimes |n\rangle \langle n| - \sum_{n=1}^{N-1} [(t\tau_z + i\Delta\tau_y) \otimes |n\rangle \langle n+1| + h.c.]. \quad (2.5)$$

Note that this Hamiltonian has particle-hole symmetry since $\mathcal{P}H_{\text{BdG}}\mathcal{P}^{-1} = -H_{\text{BdG}}$ with $\mathcal{P} = \tau_x\mathcal{K}$, where \mathcal{K} is the complex conjugation. Solving this equation for the eigenvalues then gives us the spectrum in figure 2.3, which is mirrored at $E/t = 0$ due to the particle hole symmetry. Considering this, one can clearly see that there are two phases: The topological phase where we have Majorana modes and the trivial phase where we do not have Majorana modes. In order to be in the first phase, μ has to be smaller than $2t$. To be in the second phase, however, requires μ to be bigger than $2t$.

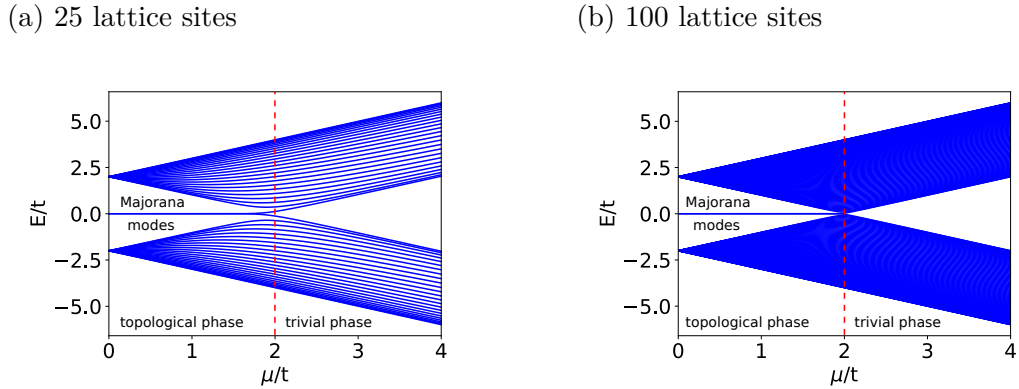


Figure 2.3.: Energy spectrum of (2.5) plotted in terms of μ . Both energies are rescaled in terms of t . Two zero energy modes appear for μ smaller than $2t$. The comparison of both plots reveals a finite size effect. (a) Here we choose 25 lattice sites and we see that the spectrum does not split at exactly $2t$ but below that. (b) Here we choose 100 lattice sites and we see that the splitting starts almost at $2t$. Thus, for an infinitely long chain the splitting will be at exactly $2t$.

To check if the zero energy states are localized at the edges, we look at the probability density in dependence of the position in the chain. The probability density is given by $|u|^2 + |v|^2$, where u are the eigenvectors corresponding to the electrons and v the eigenvectors corresponding to the holes. For the zero energy states, we indeed observe that they are localized at the edges in the topological phase and delocalized in the trivial phase (see figure 2.4). Since the number of (spatially separated) Majorana fermions is a topological invariant, this phase is called topological phase. We will explain what we mean by topological in the next section. Therefore, Majorana fermions are topologically protected in our system. It should still be mentioned that this localisation falls off exponentially the closer one gets to the transition between the phases. The localisation does not depend on the length of the chain since for longer chains the two Majorana states are more separated.

(a) topological $\mu/t = 0$

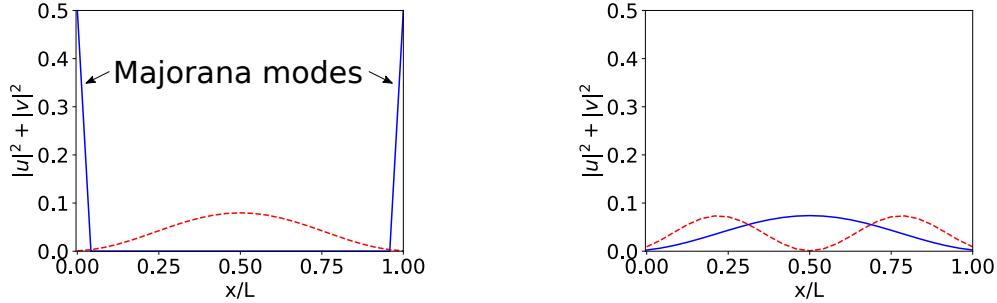
 (b) trivial $\mu/t = 4$


Figure 2.4.: Probability density of the two modes with the lowest energy with respect to the position in the chain with 25 lattice sites. Here u represents the electron eigenvector and v represents the hole eigenvector. The blue curve corresponds to the lowest energy mode and the red curve to the next higher energy mode. (a) Probability density for $\mu/t = 0$. (b) Probability density for $\mu/t = 4$.

2.2.3. Bulk-edge correspondence

Next we will examine the existence of Majorana modes by looking solely at the bulk and by exploiting the concept of bulk-edge correspondence. To do so, we have to eliminate the boundaries from the chain. We can do this by reconnecting the last site of the chain to the first one so that the chain is closed and would build a “Kitaev ring“. In the absence of boundaries, the Bogoliubov-de Gennes Hamiltonian has translation symmetry $|n\rangle \rightarrow |n+1\rangle$, since all parameters t , Δ and μ do not depend on the chain site n . In the presence of translation symmetries, it is always convenient to use Bloch’s theorem and write down the Hamiltonian in momentum space rather than in real space. In our case, a state with momentum k is given by

$$|k\rangle = \frac{1}{\sqrt{N}} \sum_{n=1}^N e^{-ikn} |n\rangle.$$

Note the boundary condition is $\langle k|n=0\rangle = \langle k|n=N\rangle$. The momentum k is then a conserved quantum number with allowed values $\frac{2\pi p}{N}$ where $p = 0, 1, 2, \dots, N-1$. Values of k , which differ by 2π , are equivalent. Using this, the Bogoliubov-de Gennes Hamiltonian in momentum space reads

$$\mathcal{H}(k) = \langle k| H_{\text{BdG}} |k\rangle = (-2t \cos k - \mu)\tau_z + 2\Delta \sin k \tau_y \quad (2.6)$$

in the basis $\psi = (\psi, \psi^\dagger) = \langle k| C$. Now we only have to diagonalize a 2x2 Matrix to obtain the band structure of the Kitaev chain model. This results in two energy bands, one with positive energy and one with negative energy:

$$E(k) = \pm \sqrt{(2t \cos k + \mu)^2 + 4\Delta^2 \sin^2 k}.$$

2. Non-Abelian Statistics and the Kitaev Chain

This shows us whether $k = 0$, resulting in a gap closing for $\mu = -2t$, or if $k = \pi$, leaving us with a gap closing for $\mu = 2t$. Since for these cases we saw a phase transition in real space, we know that a gap closing in the bulk Hamiltonian corresponds to a phase transition from trivial to topological or vice versa.

Another way to quantify this phase change is to write the Hamiltonian in the following form

$$\mathcal{H}(k) = d_x(k)\tau_x + d_y(k)\tau_y + d_z(k)\tau_z$$

where $d_{x,y,z}$ are the components of the k -dependent 3-dimensional vector $\mathbf{d}(k)$. These components can be read off from equation (2.6) and assume the form

$$d_x(k) = 0; \quad d_y(k) = 2\Delta \sin(k); \quad d_z(k) = -2t \cos(k) - \mu.$$

From this we see in figure 2.5 that as k goes from zero to 2π , the path that the endpoint of the vector $\mathbf{d}(k)$ traces out is an ellipse of radius 2 in z -direction and radius 2Δ in y -direction on the d_z, d_y plane, centered at $(\mu, 0)$. The topology of this loop can be characterized by an integer, the bulk winding number ν . This counts the number of times the loop winds around the origin of the d_z, d_y plane. The behaviour of the bands can then be directly connected to topology since objects with the same topology can be deformed continuously into each other. Additionally, it is not possible to get from one topology to another topology without making a cut.

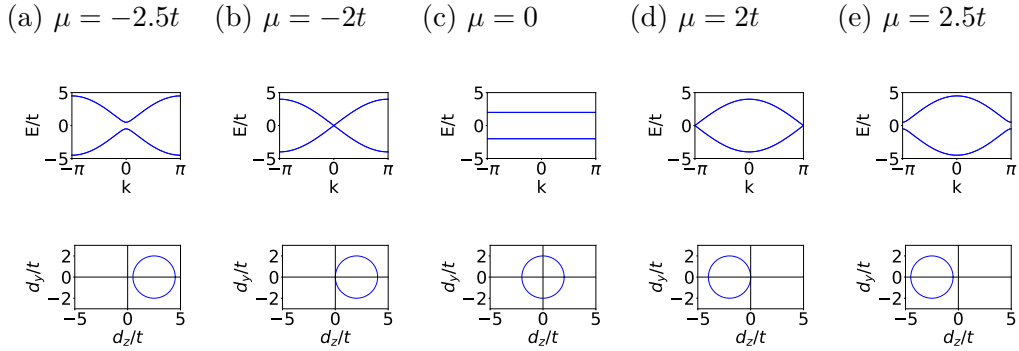


Figure 2.5.: Bandstructure of equation (2.6) for $\Delta/t=1$ upper panel and $\mathbf{d}(k)$ on lower panel. The plots show two different approaches to classify topological phases. (a) trivial phase, $\nu = 0$, $\mu = -2.5t$ (b) in between the two phases, $\nu = \text{undefined}$, $\mu = -2t$ (c) topological phase, $\nu = 1$, $\mu = 0$ (d) in between the two phases, $\nu = \text{undefined}$, $\mu = 2t$ (e) trivial phase, $\nu = 0$, $\mu = 2.5t$. Here ν is an integer which describes how many times the loop winds around the origin and μ corresponds to the chemical potential. Hence, with μ we can control if the system is topological or not.

2.3. Engineering an experimental platform

Since we understand the basic idea of the Kitaev chain and Majorana fermions, we want to engineer a system where these effects can be observed experimentally. One possibility is to build the chain out of a semiconductor. Since in semiconductors the electron density is very low, the chemical potential is near the bottom of the band. This makes it easier to define μ with respect to the bottom of the band:

$$\mu \rightarrow \mu - 2t.$$

Now the transition between trivial and non-trivial states occurs, assuming $\mu = 0$. Us-

(a) no Majorana modes

(b) Majorana modes

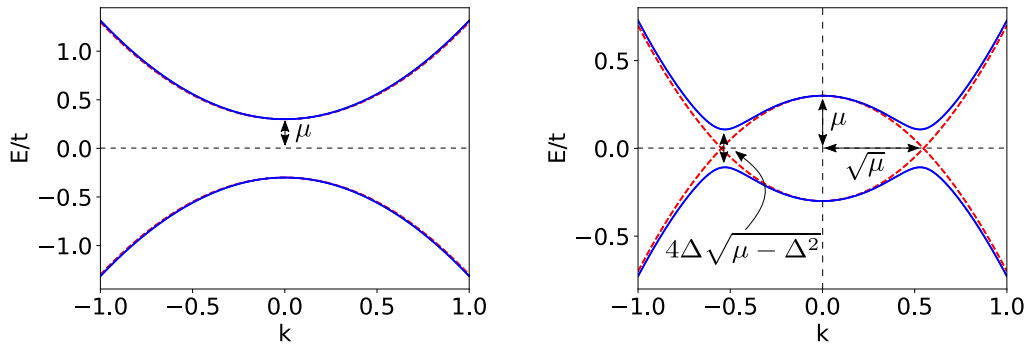


Figure 2.6.: Band structure of (2.7) with $\Delta/t = 0.1$ for the solid line and $\Delta/t = 0$ for dashed line. We plot the eigenvalues E scaled with the hopping parameter t with respect to the momentum k . There one sees that Δ opens a gap, where the two bands overlap. (a) For $\mu < \Delta$ the system is trivial. Here we chose $\mu/t = -0.3$ and we see the characteristic parabolic shape for a trivial phase. (b) For $\mu > \Delta$ the system is topological. Here we chose $\mu/t = 0.3$ and observe the characteristic double well shape for the topological phase.

ally semiconductors are not additionally superconducting. To achieve a similar effect, we build a superconductor-semiconductor heterostructure letting the superconductor induce superconductivity in the semiconductor. Next we consider that μ will always stay small compared to the bandwidth, so $\mu \ll 2t$. The same holds for superconducting pairing $\Delta \ll t$. This is the case because superconductivity is a very weak effect compared to the kinetic energy of electrons. These two inequalities combined mean that we can expand (2.6) in terms of k , at the bottom of the band leading to the continuum limit of the Kitaev model:

$$\mathcal{H} = \left(\frac{k^2}{2m} - \mu \right) \tau_z + 2\Delta\tau_y k. \quad (2.7)$$

The effective electron mass m is the coefficient of the expansion and $2t \sim \frac{1}{m}$. Let us now look at the band structure in the trivial and in the topological phase shown in figure 2.5. In the trivial phase, we see that the bands are $\sim \pm k^2$, and in the topological phase, we get a characteristic double well shape since we were looking for a gap closing and reopening. It is only a sign of topological character in the way we show it here but is not a necessary condition for that.

2.3.1. Adding spin

Until now, we only considered spinless fermions, which do not really exist. Therefore, we need to introduce spin to our Hamiltonian by introducing a new space. This new space is called spin space, and we define the Pauli matrices in it as σ_i . Since we want that only one spin component is in the topological phase and the other spin component is in the trivial phase, we have to control the chemical potential μ . For this reason we choose spin up as $\mu > 0$ and spin down as $\mu < 0$, i.e. spin down carries the topological character of the chain. Another important point is that since our Hamiltonian in (2.8) has also time reversal symmetry for $B = 0$ since $\mathcal{T}\mathcal{H}\mathcal{T}^\dagger = -\mathcal{H}$ with $\mathcal{T} = -i\sigma_y$ being the time reversal operator; the eigenstates are degenerate. This is due to Kramer's-Theorem saying that, for every eigenstate, we can construct a new eigenstate $\mathcal{T}\psi$ with the same energy. By coupling the spin to an external magnetic field this condition becomes natural and the corresponding Hamiltonian assumes the form

$$\mathcal{H} = \left(\frac{k^2}{2m} \hat{I} - \mu \hat{I} - B\sigma_z \right) \otimes \tau_z + 2\Delta k \hat{I} \otimes \tau_y \quad (2.8)$$

where we now extended our basis to $\psi = (\psi_\uparrow, \psi_\downarrow, \psi_\uparrow^\dagger, \psi_\downarrow^\dagger)$, τ are the Pauli matrices in particle-hole space and σ are the Pauli matrices in spin space. In the following, we will drop the identity matrices \hat{I} and the tensor product \otimes in our notation. By adding a magnetic field, we break the time reversal symmetry of the system. This explains why for $B \neq 0$, Kramer's degeneracy is lifted, leaving us with four instead of two bands. Whenever the magnetic field B is larger than μ , we have one Majorana fermion at the end of the chain. This is demonstrated in figure 2.7, where for the case $B < \mu$ we see that both bands have the characteristic double well shape and, thus, both bands have a topological character. For the case $B > \mu$ one of the bands has still the double well shape and is, therefore, topological but the other band has a parabolic shape, meaning it has no topological properties.

(a) no Majorana modes $B/t = 0.1$

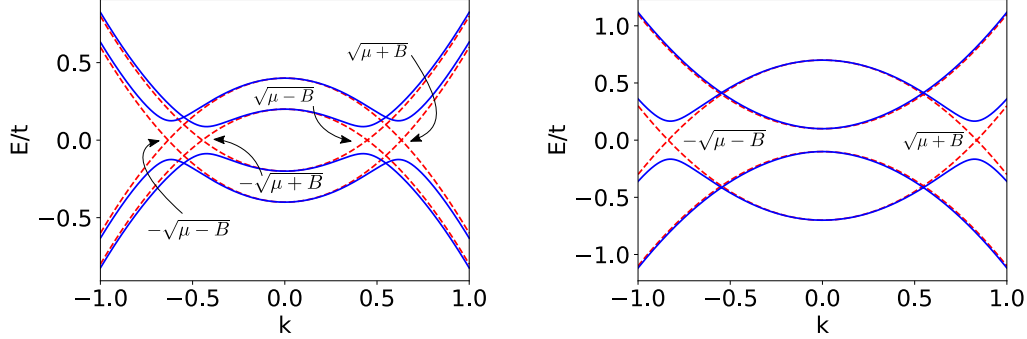
 (b) Majorana modes $B/t = 0.4$


Figure 2.7.: Band structure of (2.8) with $\Delta/t=0.1$ for solid line, $\Delta/t=0$ for dashed line and $\mu/t=0.3$. (a) For $B < \mu$ both bands are topological. This means that we cannot have Majorana modes since they would then carry a spin which contradicts the condition that they are their own antiparticles. Here we chose $B/t = 0.1$. (b) For $B > \mu$ one band is topological and the other band is trivial. Here we chose $B/t = 0.4$.

2.3.2. Superconducting s-wave pairing

In the following, we will discuss the superconducting pairing, since the most superconductors do not have p-wave pairing but rather s-wave pairing, meaning that we have a spin singlet in the annihilation and creation operators

$$\mathcal{H}_{\text{pair}} = \Delta_{\text{s-wave}}(\psi_{\uparrow}\psi_{\downarrow} - \psi_{\downarrow}\psi_{\uparrow}) + h.c. .$$

A detailed discussion on s-wave pairing can be found in appendix A.1. Before we change the superconductivity to s-wave, we make a basis transformation first. In the old basis the Hamiltonian was

$$\mathcal{H} = \begin{pmatrix} \mathcal{H}_0 & \Delta \\ \Delta^* & -\mathcal{H}_0^* \end{pmatrix}$$

where \mathcal{H}_0 describes the part of our Hamiltonian where only μ and t are present and $\Delta = 2 - i\Delta k \hat{I}$ describes the p-wave pairing as before. The basis we will use now is called the Nambu basis and is given by: $\psi = (\psi_{\uparrow}, \psi_{\downarrow}, \psi_{\downarrow}^{\dagger}, -\psi_{\uparrow}^{\dagger})$ while our Hamiltonian is now

$$\mathcal{H} = \begin{pmatrix} \mathcal{H}_0 & \Delta' \\ \Delta'^{\dagger} & -\mathcal{T}\mathcal{H}_0\mathcal{T} \end{pmatrix}$$

where $\mathcal{T} = -i\sigma_y\mathcal{K}$, $\Delta' = 2\Delta i\sigma_y k$ and \mathcal{K} is the complex conjugation. Until now we still consider p-wave pairing but in this basis the s-wave pairing is a unit matrix in spin

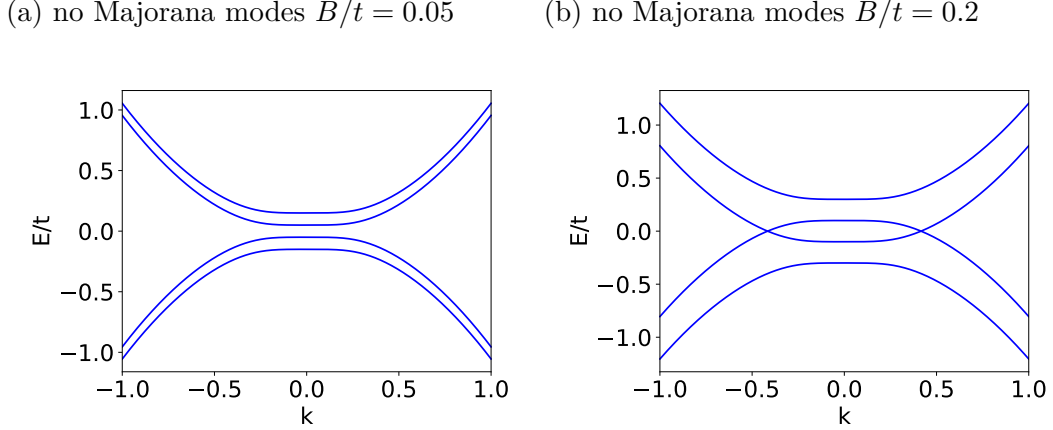


Figure 2.8.: Band structure of (2.9) with $\mu/t = 0$ and $\Delta/t = 0.1$. (a) For $B^2 < \mu^2 + \Delta^2$ the system is trivial due to spin degeneracy. Here we chose $B/t = 0.05$. (b) For $B^2 > \mu^2 + \Delta^2$ the system is topological. Here we chose $B/t = 0.2$.

space.

Now we want to write the Hamiltonian in the new basis. Since the magnetic field changes sign under time-reversal symmetry, it has the same form for electrons and holes. Because of this the Hamiltonian will be

$$\mathcal{H} = \left(\frac{k^2}{2m} - \mu \right) \tau_z + B\sigma_z + \Delta_{\text{s-wave}} \tau_x. \quad (2.9)$$

In the following, we will define $\Delta_{\text{s-wave}} \equiv \Delta$. Diagonalizing this Hamiltonian for $k = 0$ provides us the eigenvalues $E = \pm B \pm \sqrt{\mu^2 + \Delta^2}$. For $B = 0$ the system is trivial since we have spin degeneracy. That means we expect the system to be topological when $B^2 > \Delta^2 + \mu^2$. Looking at the spectrum in figure 2.8, we see that the bulk is not gapped any more which means that we cannot have Majoranas in our system since they are their own particle-hole partners and therefore cannot have any spin. The two bands that cross at zero energy in the band structure in figure 2.8 belong to opposite spin bands, and thus cannot be coupled. To open this gap, there are now two possibilities discussed in the next two subchapters.

2.3.3. Spin-orbit coupling

One way is to add spin-orbit coupling. In experiments it was shown that the spin-orbit coupling can open a gap in the s-wave superconductor [18]. By comparing (2.9) with (2.8) we see that there is a term missing which is proportional to k . Spin-orbit coupling gives such a term in its simplest form

$$\mathcal{H}_{\text{SO}} = \alpha \sigma_y k$$

which can be interpreted as a magnetic field pointing in y -direction with a strength proportional to the particle momentum. Since this term is invariant under time reversal symmetry, the Hamiltonian is

$$\mathcal{H} = \left(\frac{k^2}{2m} + \alpha\sigma_y k - \mu \right) \tau_z + B\sigma_z + \Delta\tau_x. \quad (2.10)$$

At $k = 0$ the spin-orbit coupling vanishes so it has no effect on the system being topological or trivial. Now our spectrum looks as shown in figure 2.9. We have again a

(a) no Majorana modes $B/t = 0.05$

(b) Majorana modes $B/t = 0.2$

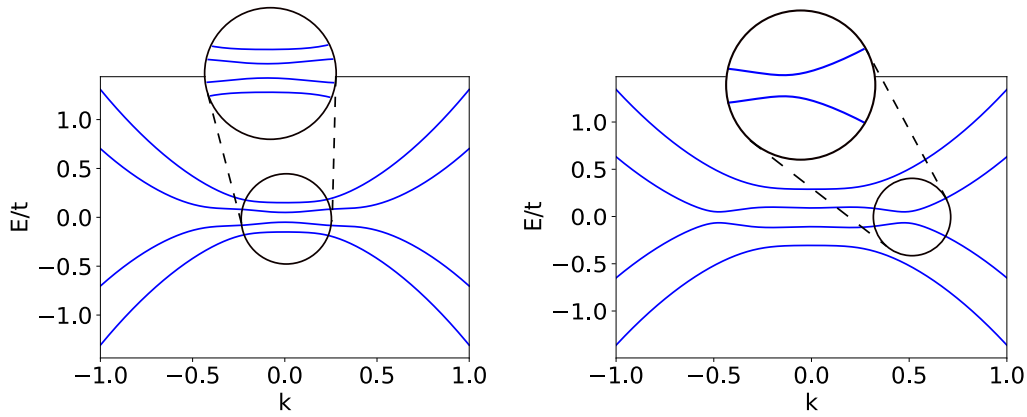


Figure 2.9.: Band structure of (2.10) with $\alpha/t = 0.3$, $\mu/t = 0$. (a) For $B^2 < \mu^2 + \Delta^2$ the system is trivial due to spin degeneracy. Here we chose $B/t = 0.05$. (b) For $B^2 > \mu^2 + \Delta^2$ the system is topological. Here we chose $B/t = 0.2$. We observe that a gap opens which allows the existence of Majorana modes in this system.

gapped bulk with two bands showing a parabolic shape. Thus, those correspond to the trivial phase. The other two bands have a double well shape, meaning that they correspond to the topological phase.

Next we need to map the Hamiltonian in equation (2.10) to the Hamiltonian in (2.7). To observe the same effects as in the toy model, there should exist a continuous transformation connecting those two Hamiltonians. Following [16], we can do this by going to the limit where the bands are nearly spin polarized in the direction of the magnetic field. We then have to only consider the lower band but the induced superconductivity is $\sim \psi_\downarrow \psi_\uparrow$. As a result, we still have to consider these terms by writing $\psi_\uparrow \sim \frac{i\alpha k}{B} \psi_\downarrow$ to project everything to one band since only then we are able to get Majorana modes. With that we can write (2.10) as

$$\psi^\dagger \mathcal{H} \psi \sim \psi_\downarrow^\dagger \left(\frac{k^2}{2m} - \mu - B \right) \psi_\downarrow + \psi_\downarrow \Delta \cdot \frac{i\alpha k}{B} \psi_\downarrow + h.c.,$$

whereas equation (2.7) can be written as

$$\psi^\dagger \mathcal{H} \psi = \psi^\dagger \left(\frac{k^2}{2m} - \mu \right) \psi + \psi 2i \Delta k \psi + h.c. .$$

Comparing these two equations, we receive an effective description for our engineered system:

$$\psi^\dagger \mathcal{H}_{\text{eff}} \psi \sim \psi^\dagger \left(\frac{k^2}{2m} - \mu_{\text{eff}} \right) \psi + \psi 2i \Delta_{\text{eff}} k \psi + h.c.$$

where $\mu_{\text{eff}} = \mu + B$ and $\Delta_{\text{eff}} = \frac{\alpha}{2B}$. With that we see that the effective superconducting pairing is reduced by the magnetic field and enhanced by the spin-orbit coupling. On top, the effective chemical potential depends on the magnetic field.

2.3.4. Inhomogeneous magnetic field

Another option to open a gap is to consider an inhomogeneous magnetic field. For that we have to go back to real space since for a general inhomogeneous magnetic field the system is not translational symmetric. So we write equation (2.9) in real space in Nambu basis, meaning $\mathbf{c} = (c_\uparrow, c_\downarrow, c_\downarrow^\dagger, -c_\uparrow^\dagger)$, leading to

$$H = \sum_i c_i^\dagger (t\tau_z) c_{i+1} + c_i^\dagger (-\mu\tau_z + \mathbf{B}_i \cdot \boldsymbol{\sigma} + \Delta\tau_x) c_i$$

where σ are the Pauli matrices in spin space and τ are the matrices in particle hole space. Now the magnetic field \mathbf{B} does depend on i . The position dependence can be written as $\mathbf{B}_i = B_0 \hat{\mathbf{b}}_i = (\sin \theta_i \cos \phi_i, \sin \theta_i \sin \phi_i, \cos \theta_i)$. In [17], Choy et al. showed that this can be mapped to the following Hamiltonian

$$H = - \sum_i (B_0 + \mu + t \frac{|\beta_i|^2 + |\beta_{i-1}|^2}{2B_0}) c_i^\dagger c_i + \sum_i [t\alpha^* c_i^\dagger c_{i+1} + t^2 \frac{\beta_{i+1}^* \beta_i}{2B_0} c_i^\dagger c_{i+2} + \left(\frac{1}{2B_0} - \frac{1}{2\mu} \right) \Delta t \beta_i c_i c_{i+1} + h.c.] \quad (2.11)$$

where α_n and β_n are parameters given by the spatial structure of the inhomogeneous magnetic field and can be written as

$$\alpha_i = \cos \frac{\theta_i}{2} \cos \frac{\theta_{i+1}}{2} + \sin \frac{\theta_i}{2} \sin \frac{\theta_{i+1}}{2} e^{-i(\phi_i - \phi_{i+1})}$$

$$\beta_i = -\sin \frac{\theta_i}{2} \cos \frac{\theta_{i+1}}{2} e^{i\phi_i} + \cos \frac{\theta_i}{2} \sin \frac{\theta_{i+1}}{2} e^{i\phi_{i+1}}.$$

This effective Hamiltonian can be identified as the Kitaev Chain with the addition of a next-nearest neighbour hopping term [17]. The pair potential $\Delta_{\text{eff}} \sim \Delta t \beta_i$ vanishes if the magnetic moments are aligned because then $\beta_i = 0$.

In the next chapter, we will introduce a method with which we take care of the self-consistency condition of the pairing Δ . This condition is part of the derivation of the theory of superconductivity and should be taken into consideration if we want to get a stable solution of the considered system.

3

Methods

In this chapter we build a framework in order to solve equation (2.10) and (2.11) not only by diagonalization but also with a self-consistent field method. With the latter we will be able to obtain solutions for the Bogoliubov de Gennes (BdG) equations which are in a local minimum of the energy landscape. This ensures that the solution is stable to local perturbations. With this method we will also be able to study the behaviour of the considered system with respect to the temperature and the Debye frequency. In the following we will do a mean field approximation which allows us to derive the self-consistency condition. The code for this method can be found in chapter A.3 and is based on a Matlab script of Kjetil Hals which he used in [29].

3.1. Derivation of the Bogoliubov de Gennes Hamiltonian

In order to include the self-consistency condition, we need to start from a mean field theory. As a consequence, we consider a Hamiltonian consisting of a contribution describing non-interacting fermions H_0 and a contribution that describes the interactions H_I . Since equation (2.10) is written in the continuum limit, we will write the Hamiltonian in a similar form. Additionally, we consider the field operators to be space dependent because this allows us to consider systems without translational invariance. Therefore the Hamiltonian can be written as

$$H = H_0 + H_I = \int d\vec{r} \psi_\alpha^\dagger(\vec{r}) h_{\alpha\beta}(\vec{r}) \psi_\beta(\vec{r}) - \frac{V}{2} \int d\vec{r} \psi_\alpha^\dagger(\vec{r}) \psi_\beta^\dagger(\vec{r}) \psi_\beta(\vec{r}) \psi_\alpha(\vec{r}) \quad (3.1)$$

where $\psi_\alpha^\dagger(\vec{r})$ is the real-space creation operator for a spin α , $h_{\alpha\beta}(\vec{r})$ the one particle Hamiltonian and $V > 0$ the interaction strength between the fermions. We added a minus sign in front of the interaction because we want to treat superconductivity and thus have an attractive interaction as formulated by Bardeen, Cooper and Schrieffer [30]; the starting point of BCS theory. Now we are able to define a mean field approximation, where we consider pairs of two electrons

$$AB \cong \langle A \rangle B + A \langle B \rangle - \langle A \rangle \langle B \rangle$$

with $A = \psi_{\uparrow}^{\dagger}(\vec{r})\psi_{\downarrow}^{\dagger}(\vec{r})$ and $B = \psi_{\downarrow}(\vec{r})\psi_{\uparrow}(\vec{r})$. Here, we have to keep in mind that this pairing is only valid in a small shell around the Fermi energy. Ignoring the $\langle A \rangle \langle B \rangle$ term being a constant which shifts our energy spectrum, our interaction Hamiltonian in the mean field approximation is given by

$$H_I = \int d\vec{r} \left\{ \Delta(\vec{r})\psi_{\uparrow}^{\dagger}(\vec{r})\psi_{\downarrow}^{\dagger}(\vec{r}) + \Delta^*(\vec{r})\psi_{\downarrow}(\vec{r})\psi_{\uparrow}(\vec{r}) \right\}$$

where $\Delta(\vec{r}) = -V\langle\psi_{\downarrow}(\vec{r})\psi_{\uparrow}(\vec{r})\rangle = V\langle\psi_{\uparrow}(\vec{r})\psi_{\downarrow}(\vec{r})\rangle = \frac{V}{2}\rho_{\alpha\beta}\langle\psi_{\alpha}(\vec{r})\psi_{\beta}(\vec{r})\rangle$ and $\rho_{\alpha\beta} = (i\sigma_y)_{\alpha\beta}$, which corresponds to s-wave pairing (see appendix A.1).

Next we perform a Bogoliubov transformation in order to diagonalize the Hamiltonian. This means we express the field operator $\psi_{\sigma}(\vec{r})$ as a linear combination of the creation γ_n^{\dagger} and annihilation γ_n operators, inducing particle-hole symmetry in which the Hamiltonian is diagonal.

$$\psi_{\sigma}(\vec{r}) = \sum_{n>0} \left\{ u_{\sigma,n}(\vec{r})\gamma_n + v_{\sigma,n}^*(\vec{r})\gamma_n^{\dagger} \right\}. \quad (3.2)$$

From now on we will discontinue writing the dependence of \vec{r} explicitly. Here γ_n corresponds to quasiparticle excitations of the system. The coefficients in front of the operators will be identified as the electron u and hole v wavefunctions of the quasiparticles.

In order to get an equation for these wavefunctions, we first calculate the commutator with the Hamiltonian in equation (3.1)

$$[H, \psi_{\sigma}] = -h_{\sigma\beta}\psi_{\beta} - \Delta\rho_{\sigma\beta}\psi_{\beta}^{\dagger}. \quad (3.3)$$

The same can be done for the diagonalized Hamiltonian $H_{\text{diag}} = \sum_n \varepsilon_n \gamma_n^{\dagger} \gamma_n$, where ε is the energy of the excitation. Since the Hamiltonian is diagonal in the γ operators they obey the commutation relation

$$\begin{aligned} [H_{\text{diag}}, \gamma_n] &= -\varepsilon_n \gamma_n \\ [H_{\text{diag}}, \gamma_n^{\dagger}] &= \varepsilon_n \gamma_n^{\dagger} \end{aligned} \quad (3.4)$$

For the first one, equation (3.3) and then (3.2) was used, while for the second one, equation (3.2) and then equation (3.4) was used. This leads to

$$\begin{aligned} [H, \psi_{\sigma}] &= \sum_{n>0} \left\{ (-h_{\sigma\beta}u_{\beta,n} - \Delta\rho_{\sigma\beta}v_{\beta,n})\gamma_n + (-h_{\sigma\beta}v_{\beta,n}^* - \Delta\rho_{\sigma\beta}u_{\beta,n}^*)\gamma_n^{\dagger} \right\} \\ [H, \psi_{\sigma}] &= \sum_{n>0} \left\{ -u_{\sigma,n}\varepsilon_n\gamma_n + v_{\sigma,n}^*\varepsilon_n\gamma_n^{\dagger} \right\}. \end{aligned}$$

Comparing the factors in front of the γ operators this obeys the following matrix equation

$$\begin{pmatrix} \vec{h} & \Delta\vec{\rho} \\ -\Delta^*\vec{\rho} & -\vec{h}^* \end{pmatrix} \begin{pmatrix} \vec{u}_n \\ \vec{v}_n \end{pmatrix} = \varepsilon_n \begin{pmatrix} \vec{u}_n \\ \vec{v}_n \end{pmatrix} \quad (3.5)$$

with $(\vec{u}_n, \vec{v}_n)^T = (u_{\uparrow,n}, u_{\downarrow,n}, v_{\uparrow,n}, v_{\downarrow,n})^T$, $\vec{\rho} = i\sigma_y$ and \vec{h} being the matrix version of $h_{\alpha\beta}$. Here we can explicitly see that u are the electron wavefunctions and v are the hole wavefunctions if we compare it with the matrix equation of the last chapter. Having this matrix equation we are able to solve a Hamiltonian consisting of a one particle contribution and an interaction of BCS type by diagonalizing for a given Δ . We can now plug in equation (2.10) and are able to obtain the results in real space. However, it is still left to calculate the pairing Δ which will lead us to the self consistency condition.

3.2. Self-consistency condition

Next we calculate the pairing Δ with the equation we got from the mean field approximation

$$\Delta = \frac{V}{2} \{ \langle \psi_{\uparrow} \psi_{\downarrow} \rangle - \langle \psi_{\downarrow} \psi_{\uparrow} \rangle \}$$

which contains the expectation values of the fermion field operators. Using the Bogoliubov transformation for the field operators (3.2) and $\langle \gamma_n^{\dagger} \gamma_n \rangle = f(\varepsilon_n)$, we receive

$$\langle \psi_{\sigma} \psi_{\bar{\sigma}} \rangle = \sum_{n>0} \left\{ (v_{\sigma,n}^* u_{\bar{\sigma},n} - v_{\bar{\sigma},n}^* u_{\sigma,n}) f(\varepsilon_n) + u_{\sigma,n} v_{\bar{\sigma},n}^* \right\}$$

where $f(\varepsilon_n) = \left(\exp\left(\frac{\varepsilon_n}{k_B T}\right) + 1 \right)^{-1}$ is the Fermi-Dirac distribution. Inserting this into the pairing Δ , we get the following equation

$$\Delta = -\frac{V}{2} \sum_{n>0} \rho_{\alpha\beta} v_{\alpha,n}^* u_{\beta,n} [1 - 2f(\varepsilon_n)]. \quad (3.6)$$

In the mean field approximation we already mentioned that the energy ε_n should stay in a small shell around the Fermi energy. Only then the superconducting state is lowest in energy and will be realized. This can be translated into the condition $0 < \varepsilon_n < \hbar\omega_D$, where ω_D is the Debye frequency. Because of this condition and the Fermi-Dirac distribution, we are now able to study the behaviour of the considered system with respect to the Debye frequency as well as the temperature. Both parameters can be used to investigate the stability of the solution. However, with the Debye temperature one is able to consider either weak superconductivity or strong superconductivity for low or high Debye frequencies respectively since it controls the region in which the superconducting state is stable.

The combination of equation (3.6) and equation (3.5) can be solved in a self-consistent way. We will use the following procedure (see figure (3.1)): First, we guess an initial pairing Δ (for example choosing the same pairing Δ for every lattice point), then we calculate the eigenvectors and eigenvalues, which can then be used to calculate the

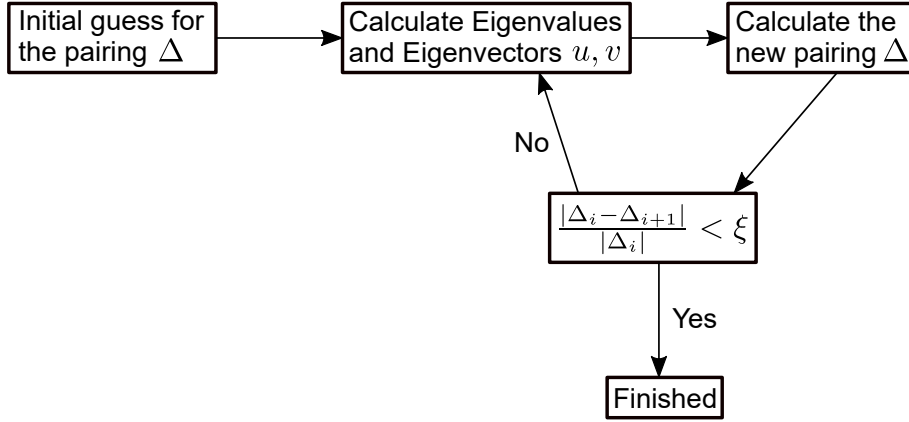


Figure 3.1.: Procedure to solve the combination of equation (3.6) and equation (3.5) self-consistently. The index i denotes the number of iterations and ξ defines the error when the iteration is fulfilled. We set the error to 10^{-4}

new pairing Δ , and finally we calculate the new eigenvectors and eigenvalues. This procedure will be repeated until the relative error of the pairing potential Δ is smaller than ξ . We measure this change with the following condition

$$\frac{|\Delta_i - \Delta_{i-1}|}{|\Delta_i|} < \xi, \quad (3.7)$$

where the index i describes the number of the iteration. We choose the value of the error ξ to be 10^{-4} because we do not see any big changes in the energy any more. In appendix A.4, one can find an example of a relaxation progress.

3.3. Discretization of the model

We derived a general model to solve a Hamiltonian with a one particle contribution and an BCS interaction contribution. However, our goal is to solve the specific Hamiltonian in equation (2.10) within this model. The BCS interaction is already included in equation (3.5), meaning that we only have to add \vec{h} as follows:

$$\vec{h} = -\frac{\hbar}{2m}(\partial_x^2 + \partial_y^2) - i\hbar\alpha_R(\hat{\sigma}_x\partial_y - \hat{\sigma}_y\partial_x) + \vec{h} \cdot \vec{\sigma}. \quad (3.8)$$

Here we generalized the previous formula so that we are able to solve 2D systems as well. For such systems, we then have to consider a more complicated spin-orbit coupling which is the Rashba spin-orbit coupling. Also we allow different directions of the magnetic field by introducing a magnetic field vector \vec{h} that couples to the spins of the electrons in the superconductor. The spins of the electrons are described by a

3. Methods

vector of Pauli matrices $\vec{\sigma}$.

In order to solve this on a computer we solve it on a lattice by choosing the central differential approximation for the derivations

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

where f is the quantity on which the derivative acts and a is the lattice constant. Performing this approximation we get

$$\begin{aligned}\vec{h}\vec{u} &= \hat{V}_{i,j}\vec{u}_{i,j} + \hat{t}_x\vec{u}_{i+a,j} + \hat{t}_x^\dagger\vec{u}_{i-a,j} + \hat{t}_y\vec{u}_{i,j+a} + \hat{t}_y^\dagger\vec{u}_{i,j-a} \\ \Delta\vec{\rho}\vec{u} &= i\Delta_{i,j}\hat{\sigma}_y\vec{u}_{i,j}\end{aligned}$$

where i and j are the indices for the lattice in x and y direction respectively, $\vec{u}_{i,j}$ are the spinors for every point on the lattice, $V_{i,j}$ are the onsite terms and t_x are hopping terms in x direction. These terms are given through

$$\begin{aligned}\hat{V}_{i,j} &= \left(\frac{2\hbar^2}{ma^2} - \mu \right) \hat{I} + \vec{h}_{i,j} \cdot \vec{\sigma} \\ \hat{t}_x &= -\frac{\hbar^2}{2ma^2} \hat{I} + i\frac{\hbar\alpha_R}{2a} \hat{\sigma}_y \\ \hat{t}_y &= -\frac{\hbar^2}{2ma^2} \hat{I} - i\frac{\hbar\alpha_R}{2a} \hat{\sigma}_x\end{aligned}$$

where \hat{I} is the identity matrix in spin space. This can then be written as a two dimensional tight binding Hamiltonians[29]

$$\begin{aligned}H &= -t \sum_{\langle ij \rangle} \left(\mathbf{c}_i^\dagger \mathbf{c}_j + h.c. \right) - \mu \sum_i \mathbf{c}_i^\dagger \mathbf{c}_i - \sum_i \mathbf{c}_i^\dagger (\mathbf{h}_i \cdot \boldsymbol{\sigma}) \mathbf{c}_i \\ &+ i\frac{\hbar\alpha_R}{2a} \sum_{\langle ij \rangle} \left(\mathbf{c}_i^\dagger \hat{\mathbf{z}} \cdot (\hat{\mathbf{d}}_{ij} \times \boldsymbol{\sigma}) \mathbf{c}_j + h.c. \right) + \sum_i \left(\Delta_i \mathbf{c}_{i\uparrow}^\dagger \mathbf{c}_{i\downarrow}^\dagger + h.c. \right).\end{aligned}\tag{3.9}$$

Here $\mathbf{c}_i^\dagger = (c_{i\uparrow}^\dagger, c_{i\downarrow}^\dagger)$, where $c_{i\alpha}^\dagger$ creates an electron with spin α at lattice site $i = (x, y)$. The symbol $\langle ij \rangle$ indicates a summation over nearest-neighbor lattice sites and \mathbf{d}_{ij} is a unit vector that points from site j to site i . Now we want to rescale these parameters by dividing everything by $t = \frac{\hbar^2}{2ma^2}$, resulting in scaling every quantity with respect to the hopping part without spin-orbit coupling. Also we ignore the constant in the onsite terms from now on. The parameters then are defined as in the following

$$\begin{aligned}\hat{V}_{i,j}^* &= -\mu^* \hat{I} + \vec{h}_{i,j}^* \cdot \vec{\sigma} \\ \hat{t}_x^* &= -\hat{I} + i\alpha_R^* \hat{\sigma}_y \\ \hat{t}_y^* &= -\hat{I} - i\alpha_R^* \hat{\sigma}_x\end{aligned}$$

where $\alpha_R^* \equiv \frac{\hbar\alpha_R}{2a} \frac{2ma^2}{\hbar^2}$.

3.4. Gap Closing Condition

In chapter 2.3 we expanded the Hamiltonian in $k = 0$ so that we could concentrate on the gap closing while engineering the experimental platform. We have to keep in mind though that there is a gap closing at $k = \pi$ as well. Since our method also treats two dimensional systems, we, furthermore, have four gap closings, namely $\mathbf{k} = (0, 0), (0, \pi), (\pi, 0), (\pi, \pi)$. In [31], Sato et al. extensively studied the Hamiltonian in equation (3.9) in momentum space with a magnetic field in z -direction. The Hamiltonian then reads

$$\mathcal{H} = \frac{1}{2} \sum_{\mathbf{k}, \sigma, \sigma'} (c_{\mathbf{k}\sigma}^\dagger, c_{-\mathbf{k}\sigma}) \mathcal{H}(\mathbf{k}) \begin{pmatrix} c_{\mathbf{k}\sigma'} \\ c_{-\mathbf{k}\sigma}^\dagger \end{pmatrix},$$

where the Bogoliubov-de Gennes Hamiltonian $\mathcal{H}(\mathbf{k})$ is given by

$$\mathcal{H}(\mathbf{k}) = \begin{pmatrix} \varepsilon(\mathbf{k}) - h_z^* \sigma_z + \alpha_R^* \mathcal{L}_0(\mathbf{k}) \cdot \boldsymbol{\sigma} & i\Delta^* \sigma_y \\ -i(\Delta^*)^* \sigma_y & -\varepsilon(\mathbf{k}) + h_z^* \sigma_z + \alpha_R^* \mathcal{L}_0(\mathbf{k}) \cdot \boldsymbol{\sigma} \end{pmatrix}$$

where the energy band dispersion is $\varepsilon(\mathbf{k}) = -2(\cos k_x + \cos k_y) - \mu^*$ and the Rashba spin-orbit coupling is $\alpha_R^* \mathcal{L}_0(\mathbf{k}) = \alpha_R^* (\sin k_y, -\sin k_x)$ with $\mathbf{k} = (k_x, k_y)$. The basis here is the same but in momentum space $\boldsymbol{\psi}_i^\dagger = (\psi_{i\uparrow}^\dagger, \psi_{i\downarrow}^\dagger)$. We will acquire the gap closing condition by diagonalising this Hamiltonian which results in

$$E(\mathbf{k}) = \sqrt{\varepsilon(\mathbf{k})^2 + (\alpha_R^*)^2 \mathcal{L}_0(\mathbf{k})^2 + (h_z^*)^2 + |\Delta^*|^2} \pm \sqrt{\varepsilon(\mathbf{k})^2 (\alpha_R^*)^2 \mathcal{L}_0(\mathbf{k})^2 + [\varepsilon(\mathbf{k})^2 + |\Delta^*|^2] (h_z^*)^2}. \quad (3.10)$$

This means that we get a gap closing if

$$\varepsilon(\mathbf{k})^2 + (\alpha_R^*)^2 \mathcal{L}_0(\mathbf{k})^2 + (h_z^*)^2 + |\Delta^*|^2 = \sqrt{\varepsilon(\mathbf{k})^2 (\alpha_R^*)^2 \mathcal{L}_0(\mathbf{k})^2 + [\varepsilon(\mathbf{k})^2 + |\Delta^*|^2] (h_z^*)^2}$$

which is equivalent to

$$\varepsilon(\mathbf{k})^2 + |\Delta^*|^2 = (\alpha_R^*)^2 \mathcal{L}_0(\mathbf{k})^2 + (h_z^*)^2, \quad |\Delta^*|^2 (\alpha_R^*)^2 \mathcal{L}_0(\mathbf{k})^2 = 0. \quad (3.11)$$

From that we find that $\mathcal{L}_0(\mathbf{k}) = 0$ since $\alpha_R^* \neq 0$ and $\Delta^* \neq 0$. Thus, we recover the four gap closing points mentioned before.

In the next chapter, we will use the derived method in order to search for Majorana modes in more complicated systems which could open the possibility to move and therefore braid these modes. The chapter is structured to continue where we opened up with the Kitaev Chain and move step by step towards the desired system.

4

From Kitaev Chain to Skyrmion-Vortex Pairs

The aim of this chapter is to find a system where we will be able to move the Majorana zero modes. This is a necessary condition which enables us to do quantum computation. We will continue to discuss the Kitaev chain, since with the introduced method, we are now able to discuss the Kitaev Chain from chapter 2 in real space representation. First, we will consider the whole system to be one dimensional. And afterwards, we study the same one dimensional system coupled to a two dimensional superconductor. These results have been compared to the ones in literature to make sure if the implemented method works correctly. Next we consider the easiest two dimensional system to identify the topological phase diagram region by looking at bulk invariants, and then we show that Majorana modes can indeed be localised, by using vortices as effective boundaries. With regard of braiding Majorana zero modes, we also investigate the system with a skyrmion in the ferromagnet. In the end, we discuss the combination of a skyrmion in the ferromagnet and a vortex in the superconductor to exploit the advantages of both systems. The code which was used throughout the whole chapter can be found in appendix A.3 and is based on a Matlab script of Kjetil Hals which he used in [29]. The code became the biggest part of the work and with it we were able to show that Majorana modes exist in skyrmion-vortex pairs.

4.1. Diagonalisation of the Kitaev Chain in Real Space

We will now continue where we left off in chapter 2 to see if the use of spin-orbit coupling or an inhomogeneous field will enable us to observe Majorana zero modes.

4.1.1. Kitaev Chain with Spin-Orbit Coupling

First, we will consider a one dimensional superconductor with an additional one dimensional magnetic semiconductor on top as shown in figure 4.1. Of course such a system does not exist, but we use it to test the proposed effective p-wave pairing. The magnetic moments are homogeneously oriented with a strength of h_0^* . We solve this

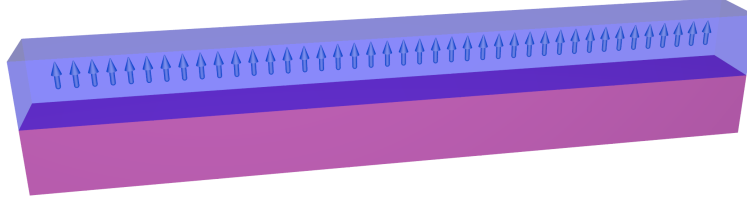


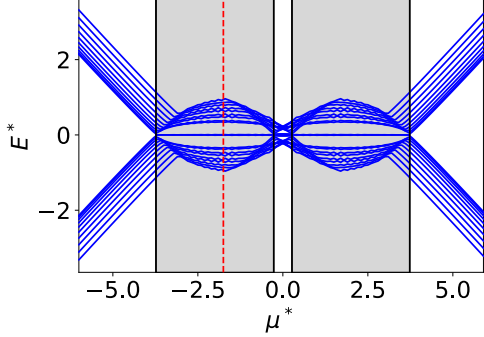
Figure 4.1.: Depiction of a 1D superconductor (purple) and a 1D magnetic semiconductor (blue). The magnetisation in the semiconductor is aligned in the z -direction.

system by diagonalizing the Bogoliubov-de Gennes Hamiltonian and plot the energy spectrum as well as the probability density of the Majorana modes in figure 4.2. Hence, we have as expected regions where the system is topological and thus we are able to observe Majorana zero modes. Further, the energy of these states is well separated from the energy of the other states. The condition in which the system is topological can be deduced from equation (3.11). Since we consider a one dimensional system, we only have two gap closing conditions being $k = 0$ and $k = \pi$. This then results in the following conditions

$$\mu^* = \pm\sqrt{(h_0^*)^2 - (\Delta^*)^2} - 2, \quad \mu^* = \pm\sqrt{(h_0^*)^2 - (\Delta^*)^2} + 2. \quad (4.1)$$

These conditions coincide with the region where the zero modes emerge. To verify those to be the zero modes we are looking for, one has to check if they are localised at the edges of the chain. To do so, we plot the probability density of these modes which can be expressed as $|u|^2 + |v|^2$ where u corresponds to the eigenvectors of the electron and v corresponds to the hole eigenvectors, see figure 4.2. One Majorana mode is localised at the right edge and the other one on the left edge with a localization length of 3 sites. We also plotted these modes for a chain length of $N_x = 25$ and $N_x = 100$ and we found out, that they get more localised by keeping the same amplitude at the edges by going to longer chains. The oscillation of the modes comes from the fact that we consider only a system which can be mapped continuously onto the Kitaev Chain toy model. We have two spin dependent bands for the electron in this system. Although we push one band with a magnetic field out of the topological region there are still effects of this non-topological band. That means that the qualitative results can be discovered but it is still not the same as the toy model. In the next part we address the idea that an inhomogeneous field can produce an effective spin-orbit coupling and therefore is sufficient to give rise to Majorana zero modes.

(a) Energy spectrum



(b) Majorana modes

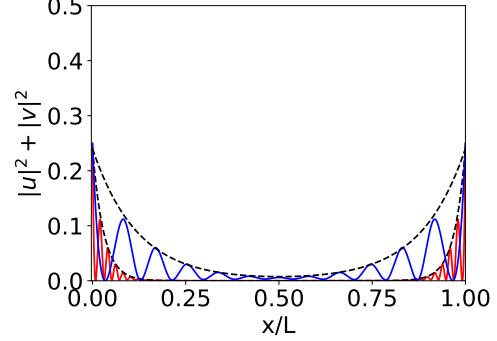


Figure 4.2.: (a) Energy spectrum of equation (3.5) where we use equation (3.8) in one dimension. We plot the 20 lowest energies with respect to the chemical potential μ . The parameters used here are: $\alpha_R^* = 0.3$, $h_0^* = 2$, $N_x = 25$ lattice points in x -direction and $N_y = 1$ lattice points in y -direction. The black vertical lines mark where the gap closes according to equation (4.1) and the shaded regions mark where the system is topological. The red dotted line marks the chemical potential for which the Majorana modes are plotted. (b) Plot of the probability density of the Majorana modes for $\mu^* = -1.75$, where u corresponds to the eigenvectors of the electrons and v corresponds to the eigenvectors corresponding to the holes. The blue graph denotes the Majorana modes of a chain with $N_x = 25$ and the red line for $N_y = 100$. One can see that the red curve gets more localised. The black dotted line is a combined exponential function for left and right edge and shows that the probability falls off exponentially to the middle of the wire with a localization length of 3 sites.

4.1.2. Kitaev Chain with inhomogeneous Magnetic Field

In order to produce the mentioned effective spin-orbit coupling, we chose a whirling magnetisation in the ferromagnet. This can be described by the following formula:

$$\mathbf{h}^* = h_0^* \cdot (\cos \Phi \sin f(r), \sin \Phi \sin f(r), \cos f(r)). \quad (4.2)$$

In this definition Φ controls whether the magnetisation whirls around the x - or y -axis. For $\Phi = \pi/2$ it whirls purely around the x -axis and for $\Phi = 0$ it whirls purely around the y -axis. Everything in between is a combination of the two cases and whirls around the axis $\sin \Phi \hat{x} - \cos \Phi \hat{y}$. The other function we define as $f(x) = n \cdot 2\pi \cdot x/N_x$ where n represents the number of windings around an axis, x is the position of the arrow and N_x is the length of the chain. The profile is defined in a way that the arrows at the edges are always pointing up. In our setup we choose $\Phi = 0$ and $n/N_x = 0.16$, which can be seen in figure 4.3. However, we don't use any spin-orbit coupling here. The energy spectrum of this configuration can again be calculated by diagonalising the corresponding Bogoliubov-de Gennes Hamiltonian. The results can be seen in figure 4.4.

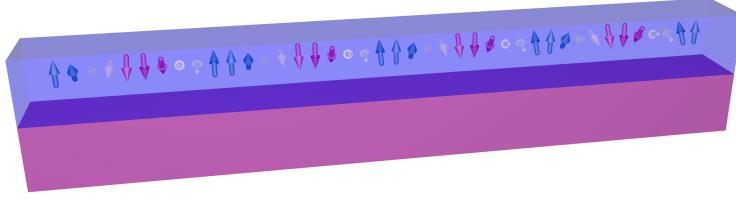
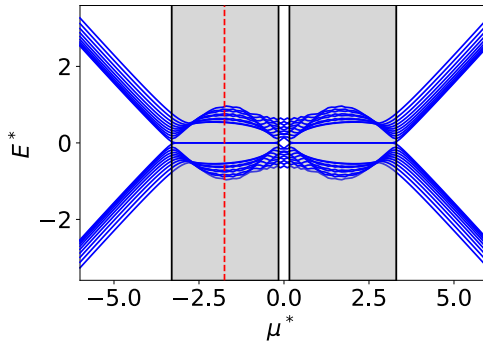


Figure 4.3.: Depiction of a 1D superconductor (purple) and a 1D magnetic semiconductor (blue). The magnetic semiconductor has an inhomogeneous magnetic structure denoted by the arrows. The colour of the arrows represent the orientation in the z -direction where blue means that the arrow is pointing up and red means that the arrow is pointing down.

This means that in the Energy spectrum we see, as expected, that there are zero modes

(a) Energy spectrum



(b) Majorana modes

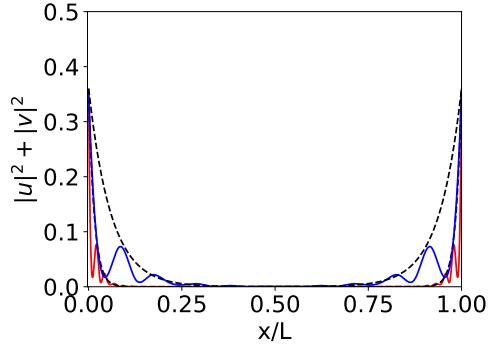


Figure 4.4.: (a) A version of figure 4.2 with spin-orbit coupling set to zero ($\alpha_R^* = 0$) and instead using an inhomogeneous magnetic field defined in equation (4.2). Furthermore, the gap between the zero modes and higher modes is bigger and the region where the zero modes emerge is shifted towards smaller μ^* . (b) Here we have a better localisation of the modes (localization length of 1.5 sites) in comparison to figure 4.2 (localization length of 3 sites).

emerging for particular μ^* . In comparison to the previous system where we had only spin-orbit coupling, the zero modes are further separated from the other states. We check if the modes are localised at the edges by plotting the probability density of the modes in the topological region. Here we see that for $N_x = 25$ the localization length is half than for the other system. Even the amplitude of the modes is much higher. If we increase the length of the chain to $N_x = 100$ the modes get more localised. Surprisingly, the system with an inhomogeneous field seems to be the more optimal one. However, then the question arises what happens if we turn on spin-orbit coupling in addition to

the inhomogeneous field. This should be studied because one will be confronted with spin-orbit coupling in an experiment.

4.1.3. Kitaev Chain with inhomogeneous Magnetic Field and Spin-Orbit coupling

In the following, we will consider a system that has spin-orbit coupling as well as an inhomogeneous magnetisation which produces effective spin-orbit coupling. We will keep the same inhomogeneous magnetisation and add spin-orbit coupling in addition. Then we diagonalise the corresponding Bogoliubov-de Gennes Hamiltonian and plot the energy spectrum and the probability density of the emerging zero modes, as seen in figure 4.5. The addition of spin-orbit coupling reduces the separation of the zero modes

(a) Energy spectrum

(b) Majorana modes

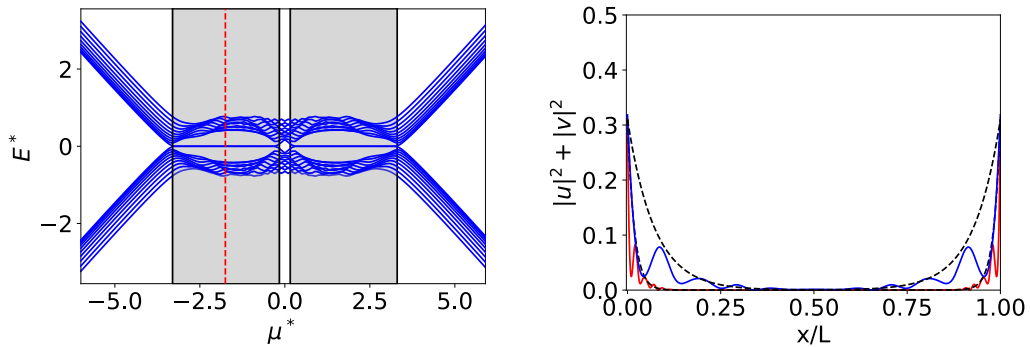


Figure 4.5.: (a) Here we used the same parameters as in figure 4.2 but with an inhomogeneous magnetisation as in figure 4.4. The gap between the zero modes and higher modes is larger than in 4.2 and smaller than in figure 4.4. The region where the zero modes emerge is identical to figure 4.4. (b) Here we get a better localisation length (1.8 sites vs 3 sites) and a higher amplitude (0.31 vs 0.24) than in figure 4.2 but a bit worse localisation (1.8 sites vs 1.5 sites) and a lower amplitude (0.31 vs 0.35) than in figure 4.4.

from the other modes. The region where it is topological is the same as in figure 4.4 since the spin-orbit coupling does not have any effect on these conditions 4.1. We also check if the modes are localised at the edges and find out that they have almost the same behaviour as for the system with only an inhomogeneous field. There is only a small increase in localization length (1.5 sites to 1.8 sites) and amplitude (0.35 to 0.31). Thus, the spin-orbit coupling does not destroy the localization of the modes, but it still lowers the threshold for which the modes are stable against external perturbations.

As a next step, we will test if these observations also occur if we solve the system self consistently. Before we only guessed a superconducting gap Δ^* but now we let the

system find its energy minimum by changing this gap spatially. This will ensure that the solution is stable against external perturbations.

4.2. Self consistent calculation of Kitaev Chain

If we try to solve the heterostructures from the previous chapter with the self consistent field method it will not converge since there is no stable one dimensional superconductor. To circumvent this problem one can consider the superconductor to be two dimensional [32]. In the following, certain parameters shown in table 4.1 are used.

| Name | Symbol | Value |
|----------------------------------|------------|-----------|
| Thermal energy | $k_B T$ | 0.001 |
| Debye frequency | ω_D | 2 |
| Effective attractive interaction | V | 5 |
| Error | ξ | 10^{-4} |

Table 4.1.: Values chosen for the self-consistent field method. All rescaled parameters with respect to t except ξ . The thermal energy was chosen to be zero because we don't study the behaviour on temperature here. However, we had to choose a small value instead of zero to avoid numerical problems. The debye frequency and the effective attractive interaction were chosen accordingly to [32] in order to see if the implemented method gives the same results as described in literature. The choice of ξ was already explained in chapter 3.

As for the thermal energy, we had to choose a value close to zero because if we choose zero the term $\frac{\varepsilon_n}{k_B T}$ in the Fermi-Dirac distribution will cause numerical problems and can be circumvented by choosing a very small value. Since we want to compare our results to [32], we chose the Debye frequency ω_D and the effective attractive interaction according to them. Since Cadez et al. [32] did only consider spin-orbit coupling and a homogeneous magnetisation, we will use this system to figure out if our method is working. After that we will extend the study in order to verify that the effective p-wave pairing also occurs in this method.

4.2.1. Spin-Orbit Coupling

At first, we consider a system consisting of a one dimensional ferromagnet placed on a two dimensional superconductor. This configuration is shown in figure 4.6.

We solve the system with the procedure shown in figure 3.1. To get an overview of when the system is topological, we plot the lowest energy mode with respect to μ^* and h_0^* . Additionally, the spin-orbit coupling is considered a constant parameter which opens the gap and therefore enables the existence of Majorana modes as discussed before. Analogously, we should see that this mode gets close to zero if we are in the topological

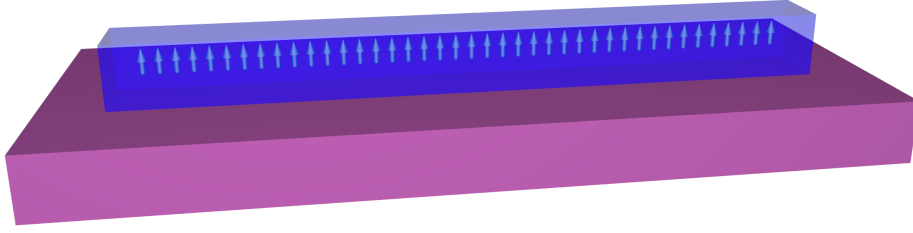


Figure 4.6.: Depiction of a 2D superconductor (purple) and a 1D magnetic semiconductor (blue). The magnetisation in the semiconductor is aligned in the z -direction. We placed the ferromagnet with 4 sites of space to the edge to improve the convergence.

region. This can be observed in figure 4.7. In principle, the blue region marks the

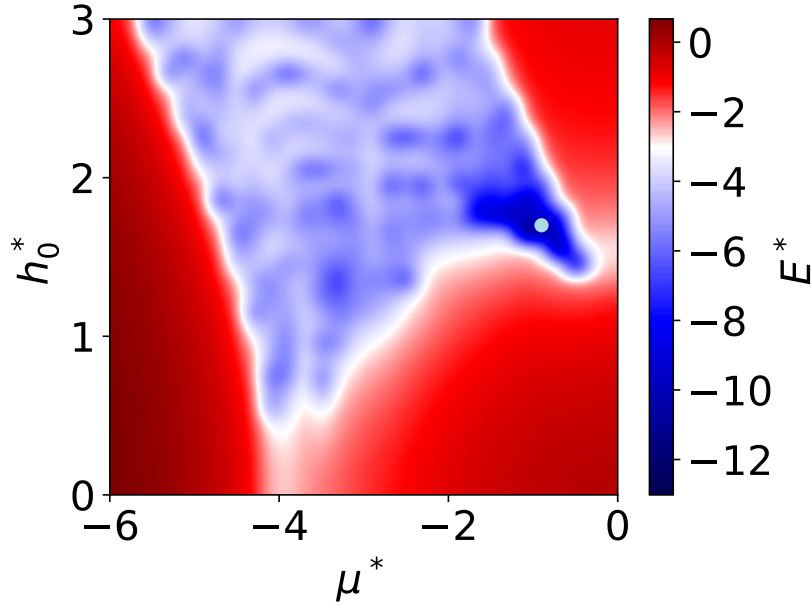


Figure 4.7.: Phase Diagram of the system with respect to h_0^* and μ^* for $\alpha_R^* = 0.3$, $N_x = 30$ and $N_y = 11$. We plotted the logarithm $\log(E^*)$ of the lowest energy mode in order to see where the transition from trivial to topological occurs. The point denotes where we will look for Majorana modes and is placed at $\mu^* = -0.9$ and $h_0^* = 1.7$.

area in which the modes localise - namely at the edges - which we also tested on a random basis. We will look into this region a bit closer and consider the parameters of the point $(-0.9, 1.7)$ as a representative for this area. Furthermore, we check the localisation of the modes for this choice of parameters. Since we have a spatially varying superconducting gap we can study how the superconductor behaves in the presence of a

magnetic field. These plots can be seen in figure 4.8. As expected, we find that the zero

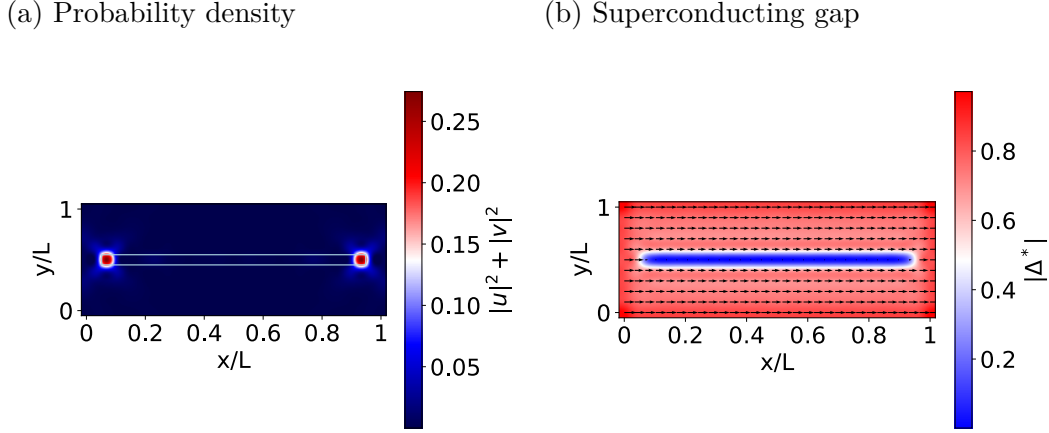


Figure 4.8.: (a) Probability density of the Majorana Modes. At the edges one can see that they are localised. The light blue block corresponds to the region, where the ferromagnet sits. (b) Plot of the superconducting gap Δ^* . The arrows denote the phase of Δ^* and the size is scaled with the absolute value. In this case the phase is always zero. The absolute value is also plotted with colours in the background. One can see that the superconducting gap goes to almost zero in the region of the ferromagnet.

modes localise at the edges of the ferromagnetic chain which is denoted as a box in the figure. We also see that in the region of the ferromagnetic box the superconducting gap goes to almost zero. That means that the magnetic field generated by the magnetisation destroys the superconductivity. This is also expected by the Ginzburg-Landau theory since one can change from the superconducting to the normal phase by applying a magnetic field. In conclusion, our superconductor is a normal metal within this region and the zero modes like to be localised here. Since there is a symmetry constraint on the configuration the modes especially like to stay at the edges of the chain.

4.2.2. Inhomogeneous Field

We will do a similar analysis for the case where the ferromagnet has a whirling magnetisation. We define this exactly as before described in equation (4.2) with the same function $f(r)$. A demonstration of this configuration can be found in figure 4.9. Analysing this with the self-consistent field method, we observe a similar diagram as for the case with only spin-orbit coupling (see figure 4.10).

We see the blue region, where we are in the topological phase, and get the localization of the modes at the edges. However, we observe that there is a region where the energy is really close to zero which coincides with the position where we found it before. We will search in the same area for Majorana zero modes as before and the results can be seen in figure 4.11. The modes are localised at the edges of the ferromagnetic region

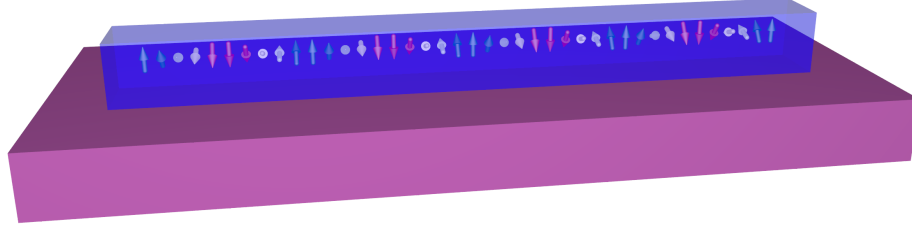


Figure 4.9.: Depiction of a 2D superconductor (purple) and a 1D magnetic semiconductor (blue). The magnetic semiconductor has a inhomogeneous magnetic structure denoted by the arrows. The colour of the arrows represents the orientation in the z -direction where blue means that the arrow is pointing up and red means that the arrow is pointing down. We placed the ferromagnet with 4 sites of space to the edge to improve the convergence.

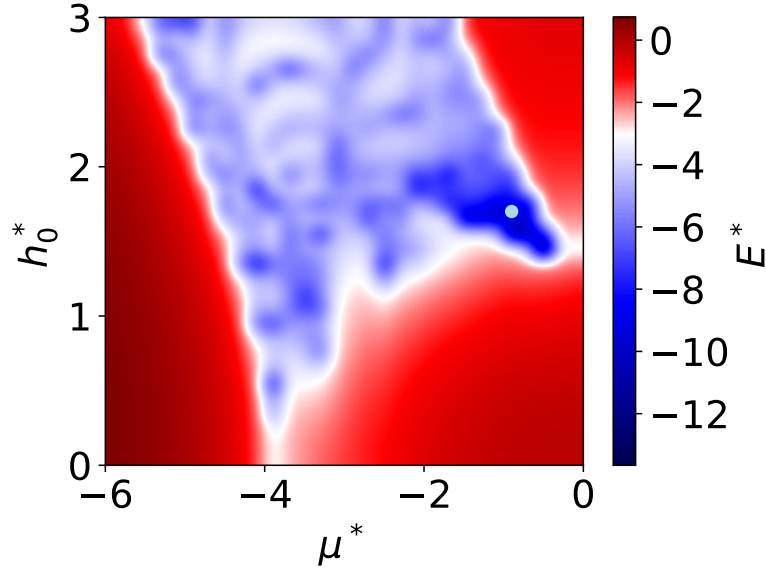


Figure 4.10.: Same parameters used as in figure 4.7 but with an inhomogeneous field and no spin-orbit coupling α_R^* . We see more or less the same plot as in 4.7. This means that the method recovers that an inhomogeneous field produces effective p-wave pairing as well as spin-orbit coupling.

and we do not see any reduction to the case before. Furthermore, the superconducting gap behaves as expected so that we have a normal metal in the vicinity of the chain and a superconductor everywhere else. This demonstrates that a winding magnetic field produces effective p-wave pairing within this method.

(a) Probability density

(b) Superconducting gap

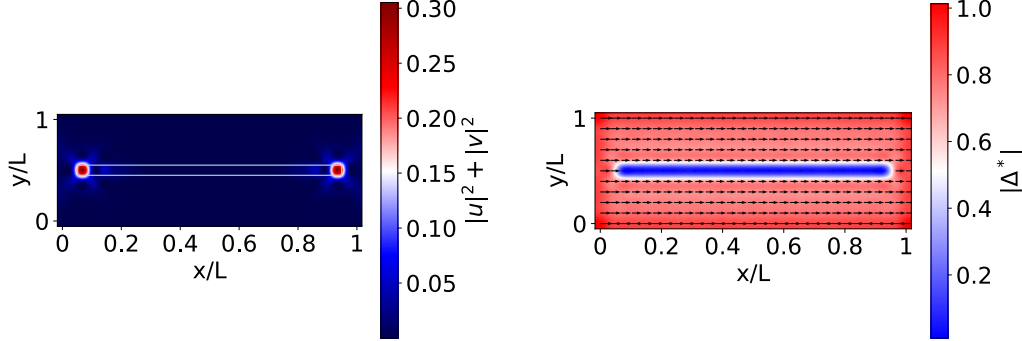


Figure 4.11.: Same as in 4.8 but with inhomogeneous field and no spin-orbit coupling. This again shows that the inhomogeneous field produces effective p-wave pairing as well as spin-orbit coupling.

4.2.3. Combination of Spin-Orbit Coupling and Inhomogeneous Field

Next we try if turning on the spin-orbit coupling destroys what we observed before. Due to the combination of spin-orbit coupling and an inhomogeneous field, one can expect that the system relaxes differently. So we plot the lowest energy mode with respect to μ^* and h_0^* , which is shown in figure 4.12. We recover the same region as before where the system is topological. We also check for the localization in figure 4.13. One can see that there is still a localization at the edges but we also see some oscillations in the middle of the ferromagnetic region. This can be improved by taking a longer chain or more windings in the magnetisation. However, the important features can still be recovered. Also, the superconducting gap behaves as before. With that we showed that the used method recovers Majorana zero modes even with the combination of spin-orbit coupling and an inhomogeneous field.

But there still remains the question of how we can move these modes in order to observe the non-Abelian statistics. This is a timely question since experiments are already able to obtain such structures [19] although the final proof of the Majorana nature is still disputed. As for the Kitaev chain, there are proposals to build a T junction and move them by tuning the local chemical potential in a time-dependent manner with the gate voltages [16]; but this is hard to do and not desirable for a device.

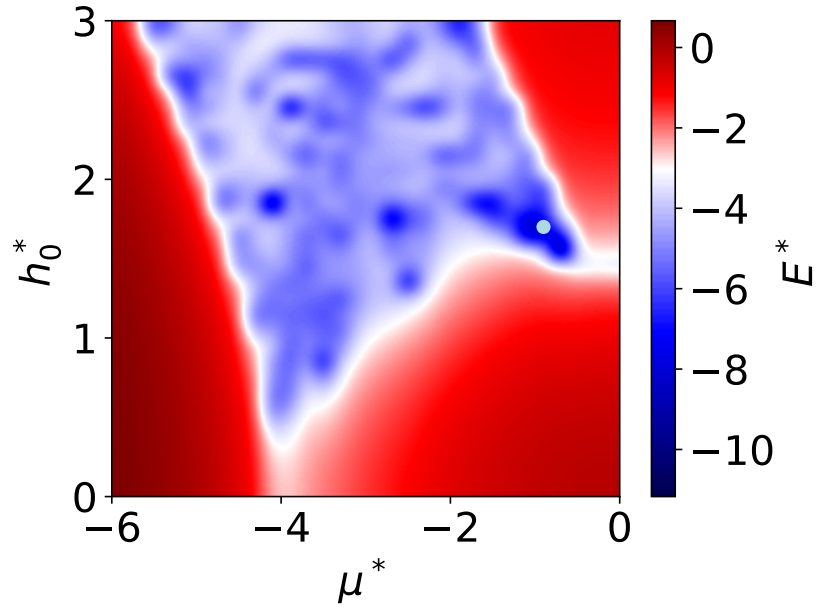


Figure 4.12.: Same parameters used as in figure 4.7 but with an inhomogeneous field. The plot looks the same as in figure 4.7 and figure 4.10. This shows that turning on the spin-orbit coupling in the presence of an inhomogeneous does not drive us into another energy minimum.

(a) Probability density

(b) Superconducting gap

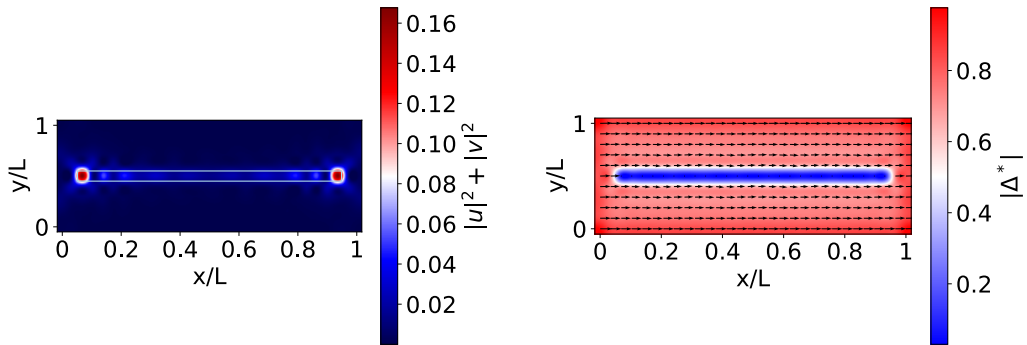


Figure 4.13.: a) Same as in 4.8 (b) Same as in 4.8. Here we see that there occur oscillations in the middle of the wire but the qualitative behaviour is the same. Furthermore, these oscillations will disappear if we consider a longer chain.

4.3. Two dimensional ferromagnetic Block on a two dimensional Superconductor

Since moving the Majorana modes in a Kitaev Chain turns out to be difficult, people began to search for Majorana modes in two dimensional systems also [33]. The easiest configuration of this is shown in figure 4.14 where a two dimensional ferromagnetic block is placed on a two dimensional superconductor. Within the ferromagnet, the magnetisation points up everywhere with the strength h_0^* . Here we want to see if we can find the signatures of Majorana modes at the edges of the ferromagnet as proposed within our method. At first we will look at the phase diagram in figure 4.15. Here the

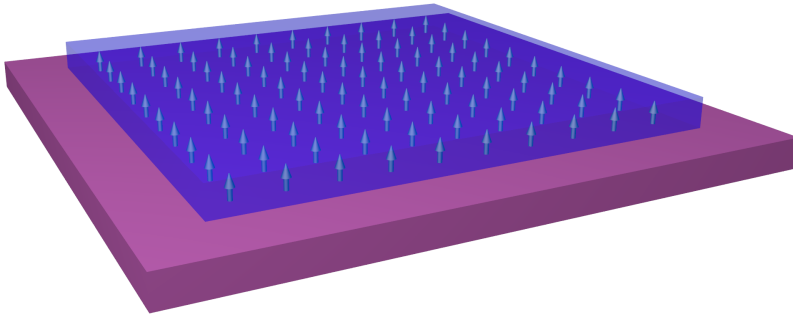


Figure 4.14.: Depiction of a homogeneous magnetized ferromagnet on top of a superconductor.

blue region shows where the system is topological. This region can also be described by the condition in equation (3.11). Since we have a two dimensional system, we have gap closings at $k = (0, 0)$, $k = (0, \pi)$, $k = (\pi, 0)$ and $k = (\pi, \pi)$. Since for a square lattice $k = (0, \pi)$ and $k = (\pi, 0)$, we will end up with the following three conditions, namely

$$(4 + \mu^*)^2 + (\Delta^*)^2 = (h_0^*)^2, \quad (\mu^*)^2 + (\Delta^*)^2 = (h_0^*)^2, \quad (4 - \mu^*)^2 + (\Delta^*)^2 = (h_0^*)^2. \quad (4.3)$$

These conditions are drawn as a black dashed line in the phase diagram. We see that this fits qualitatively with the region where the energy goes close to zero. Next we search in this phase diagram for the modes that are localised at the edge which can be seen in figure 4.16. At first we search in the same region where we searched for the Kitaev Chain. We observe that we get localised modes in the corners of the block. One mode is localised in the lower left corner and the other one is localised in the upper right corner. The other two corners are shared by both of the modes. We can exchange the position of these two modes by taking a negative chemical potential. Furthermore, we can get the configuration where one mode is in the upper left corner and the other one in the lower right corner by doing the transformation $x \rightarrow -x$ or $y \rightarrow -y$. Also, the superconducting gap goes to zero inside the ferromagnetic block as we saw it for

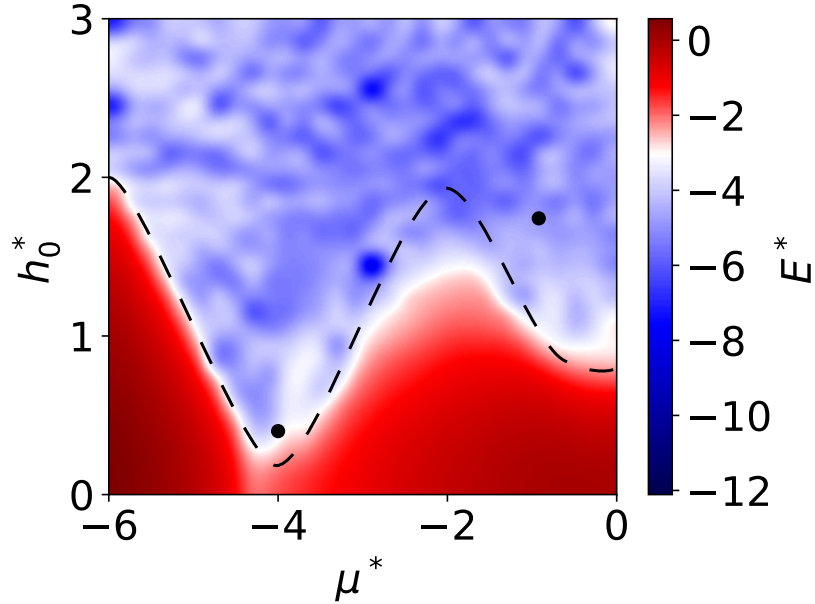


Figure 4.15.: Phase Diagram of the system with respect to h_0^* and μ^* for $\alpha_R^* = 0.5, N_x = 23$ and $N_y = 23$. We plotted the lowest energy mode logarithmic in order to see, where the transition from trivial to topological happens. Also the analytical result for the gap closing is drawn as a black dashed line. It matches only qualitatively to the shape in the picture because the superconductor is smaller than the superconductor. The points denote where we will look for the Majorana modes and is placed at $\mu^* = -0.9$ and $h_0^* = 1.7$ (point α) and $\mu^* = -4$ and $h_0^* = 0.4$ (point β). There seem to be regions, where the energy seems to be closer to zero than for the two points. However those seem to be errors from the interpolation.

the chain. A similar result can be seen in a system with a two dimensional topological insulator on top of a high temperature superconductor [34]. In comparison to our system they have two Majorana modes at each corner.

But there is another interesting region, where we get localised Majorana modes and the superconductivity is not destroyed completely. In this region one mode is localised on the upper and the right edge and the other mode is localised the left and lower edge. We can get different configurations by the same symmetry operations as before. The superconducting gap, however, behaves a bit different. Inside the block it is again reduced but it does not go as close to zero as before since the applied field is lower and the superconductor is not fully destroyed since we have a lower field.

Summing up, we see two interesting regions with localised Majorana zero modes. On one hand, they are not spatially separated and therefore are unfortunately not useful for quantum computation. On the other hand, it provides an easy check, by measuring

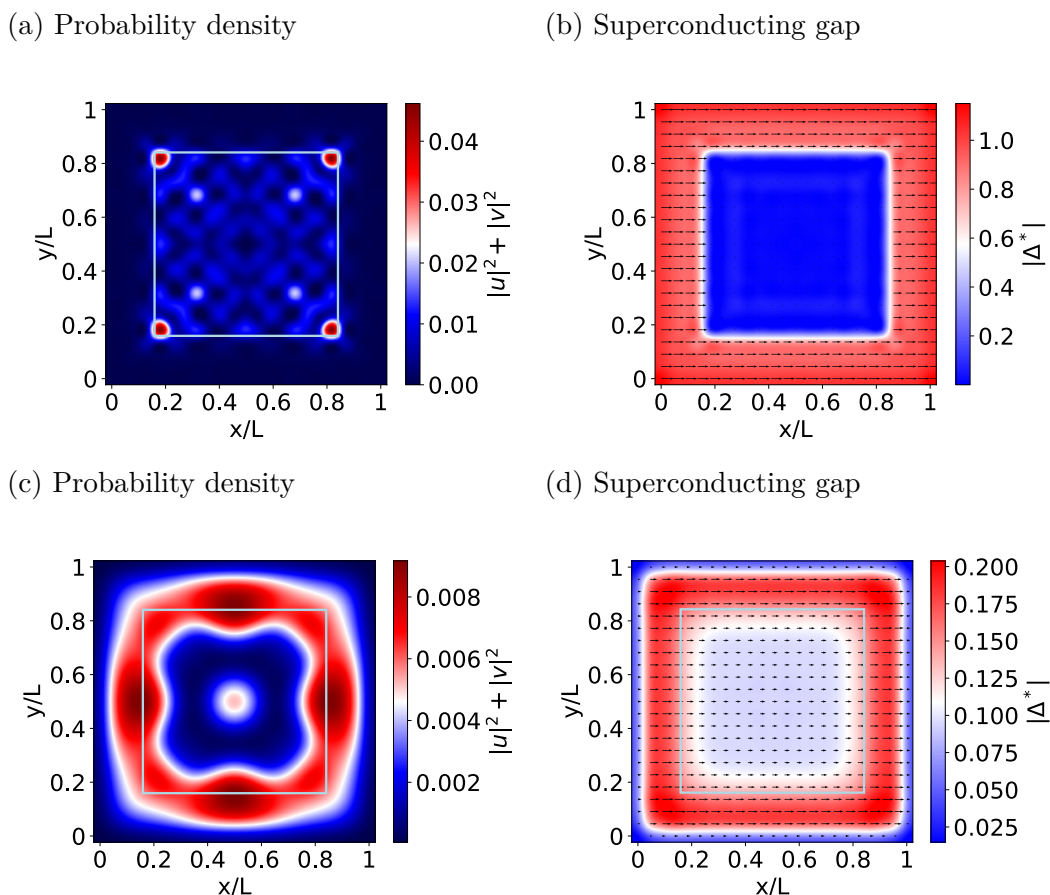


Figure 4.16.: (a,b) correspond to point α and (b,c) correspond to point β . (a) Probability density of the Majorana Modes. There are Majorana modes localised at the corners of the block and the probability is fluctuating inside the block. The light blue block corresponds to the region where the ferromagnet sits. (b) Plot of the superconducting gap Δ^* . The arrows denote the phase of Δ^* and the size is scaled with the absolute value. The absolute value is also plotted with colours in the background. One can see that the superconducting gap goes almost zero in the region of the ferromagnet. (c) Here the Majorana modes are localised at the edge of the block and there is also a small probability to find the mode in the middle. (d) The superconducting gap goes not entirely to zero but it is still reduced inside the block. Also at the edge of the superconductor it goes to zero.

the conductance, for experimentalists to see if their material has topological features. There is no need of an internal structure in the material to see if one is in the right region considering this is connected to much more effort.

4.4. Majorana Modes in Vortices

The existence of Majorana modes in superconducting vortices received great attention since they obey non-Abelian statistics [27]. For this reason, we also consider a system with a superconducting vortex within our method. We do not get an Abrikosov vortex-lattice here, which one could expect, because the system size is smaller than the separation of two vortices. With the magnetic semiconductor on top we still make sure that just one of the electron bands is topological and the spin-orbit coupling will induce effective p-wave pairing. Similar studies were also done in Refs. [35, 36]. An illustration of the configuration can be seen in figure 4.17 where the vortex is introduced as $\Delta_{\text{Vortex}} = |\Delta^*|e^{in\phi(r)}$ where $n = \pm 1$ defines the rotation of the vortex. In the following we choose $n = 1$ and $|\Delta^*| = 1$. Next we plot the lowest energy mode with respect to μ^*

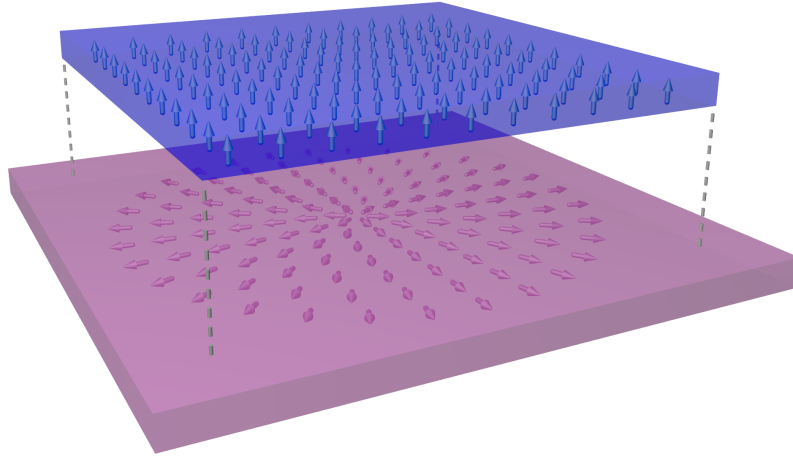


Figure 4.17.: Depiction of a homogeneous magnetized ferromagnet on top of a superconductor with a vortex in it.

and h_0^* which can be seen in figure 4.18. Here we see a completely different shape from before. It seems that we have a bigger region where our system is topological. Also, the localisation of the modes is not provided everywhere in the topological region. The parameters have to be tuned in the right way to observe the desired localization. The point in the plot marks the place where we find the best localization. In figure 4.19 we study the probability density of the zero modes and the behaviour of the superconducting gap. Here we now see that the Majorana modes are localised in the vicinity of the vortex and on the edge of the ferromagnetic region. One can also observe that the Majorana mode in the middle has the shape of a rhombus since it wants to connect to the one on the edge. The same is present for the mode on the edge which reaches for the one in the middle. This can be explained by analysing the superconducting gap. There one can observe that the superconducting gap is high outside the ferromagnetic

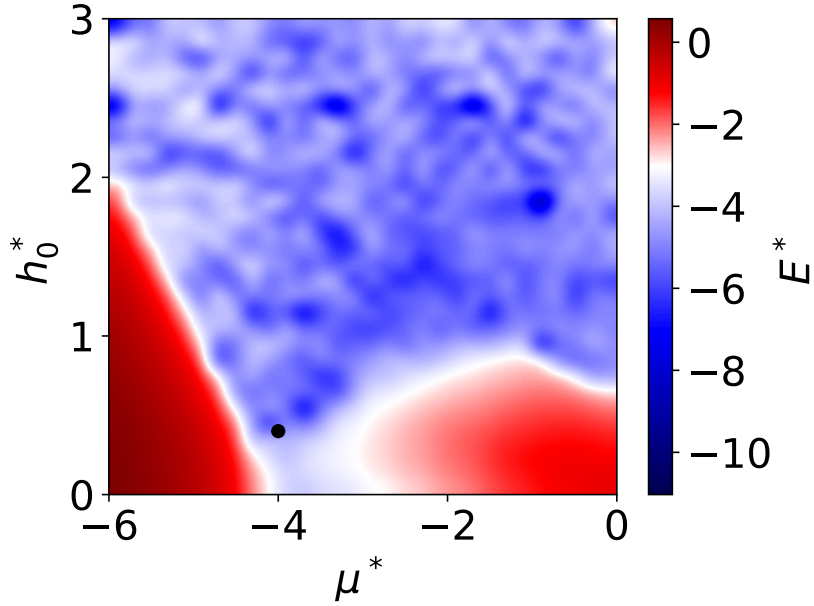


Figure 4.18.: Phase Diagram of the system with respect to h_0^* and μ^* for $\alpha_R^* = 0.5, N_x = 23$ and $N_y = 23$. We plotted the lowest energy mode logarithmic in order to see where the transition from trivial to topological happens. The point denotes where we will look for the Majorana modes and is placed at $\mu^* = -4$ and $h_0^* = 0.4$.

region and goes close to zero at the edges of the superconductor. Inside the region of the ferromagnet the superconducting gap it is lower than what we saw before. It does not go to zero though. Therefore, it is zero in the vicinity of the vortex since it is like an impurity in the superconductor. Moreover, a vortex in the superconductor is a region where the superconductor can be penetrated by a magnetic field. The Majorana modes seem to localise at regions where an abrupt change in the superconducting gap is happening. Thus, it will localise at the vortex and the edge of the ferromagnet and we have to guarantee that the superconducting gap is lower than outside the vortex. As soon as we have the same superconducting gap in the whole ferromagnet, we will lose the localisation of the Majorana zero mode at the vortex. We also tried to stabilize two vortices in this system in order to see if one Majorana mode localises at one vortex core and the other one on the second vortex core. However, because of the vortex-vortex repulsion, they push each other out of the system and the edge repulsion is not strong enough to prevent it.

We have now a two dimensional system where we observe Majorana zero modes which are spatially separated. This is convenient because vortices can be moved by low temperature magnetic force microscopy [37].

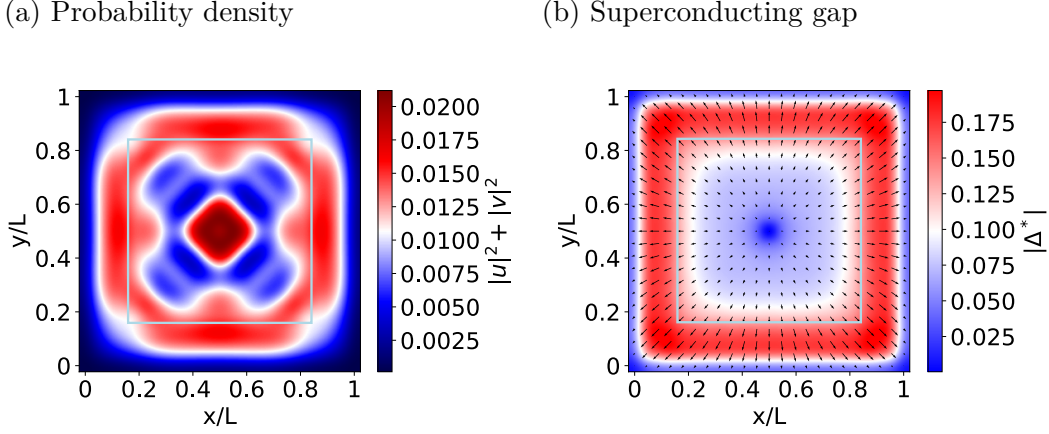


Figure 4.19.: (a) Probability density of the Majorana Modes. One can see that there is one mode localised at the vortex core and one at the edge. We scaled the one on the edge with 4 because it is hard to see since the probability is distributed over a bigger region. (b) Plot of the superconducting gap Δ^* . The arrows denote the phase of Δ^* and the size is scaled with the absolute value. The absolute value is also plotted with colours in the background. One can see that the superconducting gap goes to almost zero in the center of the vortex and is reduced in the region of the ferromagnet.

4.5. Majorana Modes in Skyrmions

Another possibility to get Majorana modes in a two dimensional system is a skyrmion in the ferromagnet on top of the superconductor. This goes back to the idea that an inhomogeneous field induces an effective spin-orbit coupling, enabling the existence of Majorana zero modes as we already have seen in chapter 4.1.3. This was studied in [20] where Yang et al. found that these modes can be localised in the vicinity of a skyrmion. Skyrmions are desired structures in the magnetisation since they can be manipulated by a variety of different techniques [38]. They are topological objects that can be characterized by the winding number

$$q = \frac{1}{4\pi} \int dx dy \mathbf{m} \cdot (\partial_x \mathbf{m} \times \partial_y \mathbf{m})$$

where \mathbf{m} describes the magnetization. This winding number is a topological invariant which induces stability to this configuration. This stability makes them suitable for future devices in information technology. Because of their drivability, one is then able to move the corresponding Majorana modes bound to the skyrmion. The most promising technique in terms of scaling arguments for quantum computers is the displacement by electrical currents [21, 22]. This could then allow for braiding and would enable using a system consisting of a skyrmion in a ferromagnet on a superconductor for quantum computation.

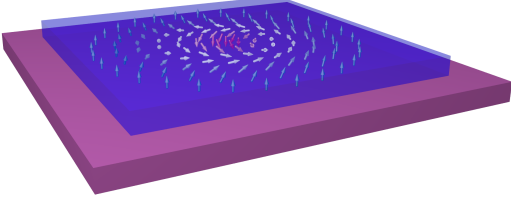
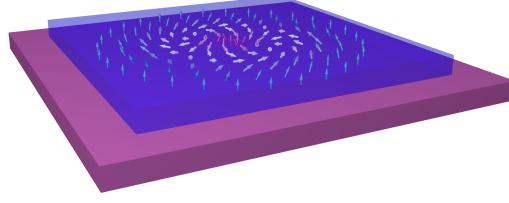
(a) Skyrmion with $q = 1$ (b) Skyrmion with $q = 2$ 

Figure 4.20.: Depiction of a ferromagnet containing a Bloch skyrmion with winding number one and two on top of a superconductor.

Two possible configurations with a skyrmion in the ferromagnet can be seen in figure 4.20, defined by

$$\mathbf{h}^* = h_0^* \cdot (\cos \Phi \sin f(r), \sin \Phi \sin f(r), \cos f(r)) \quad (4.4)$$

where h_0^* is the strength of the magnetisation, $\Phi = q\phi + \varphi$ where ϕ is the polar coordinate in real space and φ is the helicity of the skyrmion. For the function $f(r)$, we pick the following ansatz, $f(r) = \arccos(\frac{r^2 - R^2}{r^2 + R^2})$ to approximate the profile of a small skyrmion in a ferromagnetic thin film system [39]. Here r is the polar coordinate and R defines the radius of the skyrmion. In the following, we will choose $\varphi = \frac{\pi}{2}$, which corresponds to a Bloch skyrmion, $q = 2$, $R = 0.1 \cdot L$ where L corresponds to the length of the ferromagnet and we fix $f(L) = 0$.

4.5.1. Diagonalisation

First, we solve the system by diagonalizing the corresponding Bogoliubov-de Gennes Hamiltonian to see if we are able to reproduce the results in [20]. This should be the case because we used a smooth definition for the skyrmion. The results can be seen in figure 4.21. In the phase diagram, we see that the topological region starts at higher fields because for the diagonalization we considered a uniform superconducting gap Δ^* . We look at the black point for Majorana modes. Far away from this region, we will lose the localization. For a winding number $q = 1$ we do not find any localized mode at the skyrmion. That is the same result as found by Yang et al. [20]. For a winding number $q = 2$ we find one mode which forms a ring in the vicinity of the skyrmion and another mode which stays at the edge of the ferromagnet. With that we showed the existence of spatially separated Majorana modes in the presence of a skyrmion in the ferromagnet.

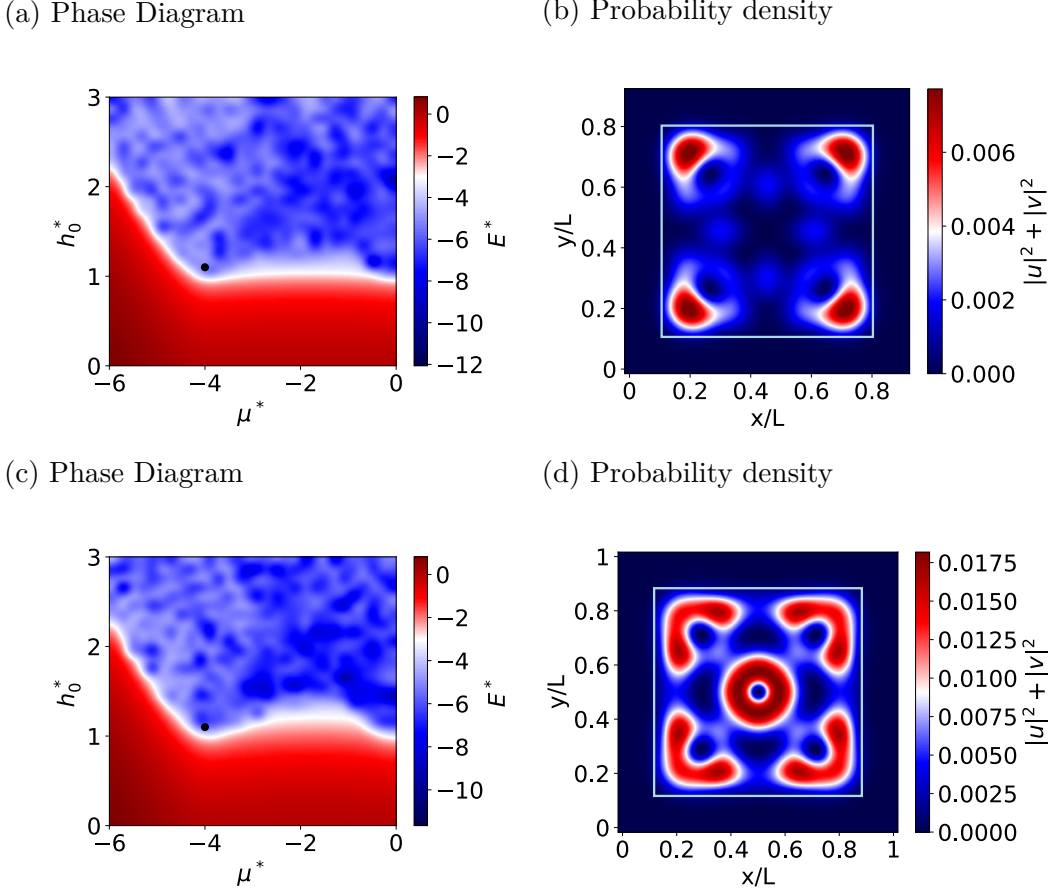


Figure 4.21.: (a,c) Here we choose $|\Delta^*| = 1$, $\alpha_R^* = 0.5$, $N_x = N_y = 31$ and a skyrmion winding number of $q = 1$ for (a) and $q = 2$ for (b) on a superconductor. The black point is at $h_0^* = 1.1$ and $\mu^* = -4$. (b,d) Probability density of the Majorana modes. For (b) there is no localisation in the skyrmion and for (d) we see a ring shaped localisation in the skyrmion.

4.5.2. Self-Consistent Field Method

Now we test if the observed Majorana mode is indeed a stable solution of this system by using the self-consistent field method. The phase diagram can be seen in figure 4.22. We see that the topological region starts for lower fields since the superconductivity is destroyed by the ferromagnet. This results in a topological region even for lower magnetisation strengths. We search for Majorana modes in a region close to the one where we found Majorana modes in the superconducting vortex. The results can be seen in figure 4.23. In this figure we observe, by using the self-consistent field method, that the spatial separation of the Majorana zero mode in the vicinity of the skyrmion

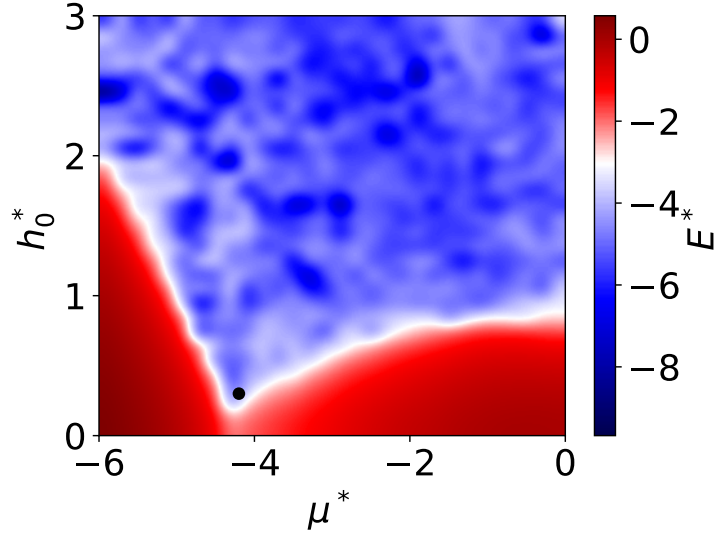


Figure 4.22.: Here we used $\alpha_R^* = 0.5$, $N_x = 23$ and $N_y = 23$ to provide convergence of the method. The point is placed at $\mu^* = -4.2$ and $h_0^* = 0.2$.

(a) Probability density

(b) Superconducting gap

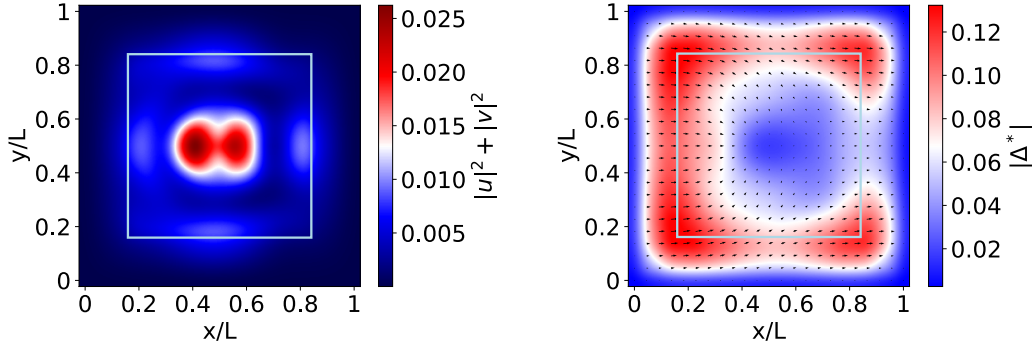


Figure 4.23.: (a) Probability density of the Majorana Modes. One can see that the modes are not spatially separated and not well localised. (b) Plot of the superconducting gap Δ^* . One can see that the superconducting gap is reduced inside the skyrmion and also on the right edge of the ferromagnet. This comes from the shape of the magnetisation in a $q = 2$ skyrmion since on the right side the magnetisation is pointing up and therefore extends the region where superconductivity is destroyed to the right.

is suppressed. We also searched for different parameters in the topological region in the phase diagram only to find that the Majorana zero modes are not localised at the

skyrmion anymore. The shape of this mode can be explained by the superconducting gap. There we see that the superconductivity is reduced in the region where the magnetisation points up and the Majorana mode wants to localise inside this region. The gap is also reduced on the right side of the ferromagnet which comes from the shape of the skyrmion. In addition, the magnetisation of the skyrmion has an influence on the phase of the superconductor since it changes its direction in the middle of the sample. Since we cannot localise Majorana modes inside the skyrmion this does not seem to be a stable solution within the self-consistent field method. Thus, in the next section we will try another configuration to exploit the features of the skyrmion for topological quantum computation.

4.6. Majorana Modes in Skyrmion-Vortex Pairs

To summarize, we saw that localization of Majorana modes is not a stable solution of the self-consistent field method. However, we observed that Majorana modes can be stabilized in vortex cores. Therefore, the idea now is to combine these two configurations and see if this gives even a better result. Another reason to consider such skyrmion-vortex pairs is that an attractive interaction between skyrmions and vortices was proposed [29]. Because of this interaction, it ought to be possible to drive the skyrmions with an electrical current dragging the vortex and the Majorana zero mode, which is bound to it, with them. This enables us to do braiding within this system

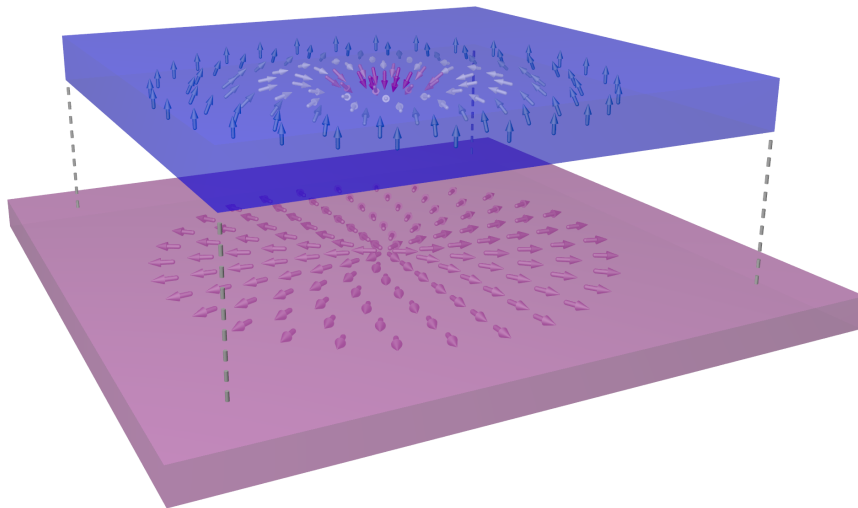


Figure 4.24.: Depiction of a ferromagnet containing a Bloch skyrmion with winding number two on top of a superconductor.

with the advantage that the zero mode is strongly bound to the vortex. A sketch of this

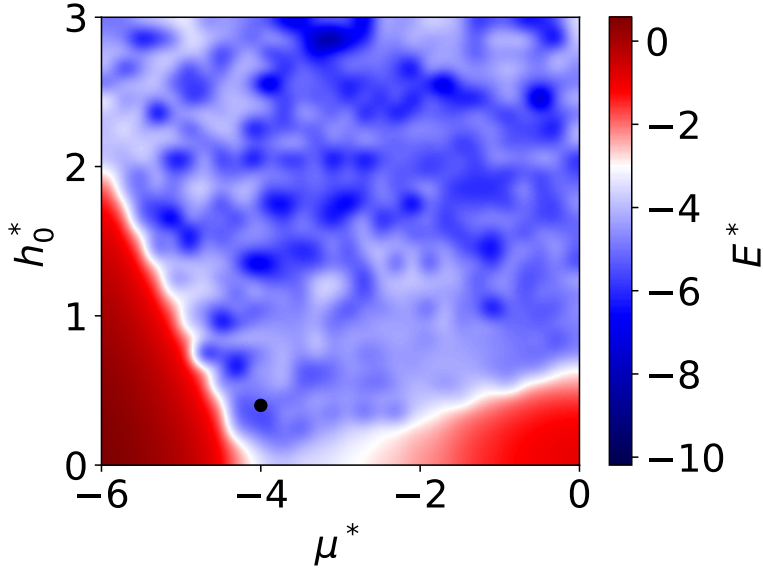


Figure 4.25.: Phase Diagram of the system with respect to h_0^* and μ^* for $\alpha_R^* = 0.5$, $N_x = 23$ and $N_y = 23$. We plotted the lowest energy mode logarithmic in order to see where the transition from trivial to topological happens. The point denotes where we will look for the Majorana modes and is placed at $\mu^* = -4$ and $h_0^* = 0.4$.

configuration can be seen in 4.24. We chose the same parametrization of the skyrmion as in equation (4.4) and the same definition of the vortex as before. However, given that a skyrmion with $q = 2$ is fairly unpractical, we choose to use $q = 1$ instead. Since we consider thin films, it is plausible to use $\varphi = 0$ which is a Néel skyrmion. The phase diagram for this system is shown in figure 4.25. We observe that the region is extended in comparison to the one of the vortex. If we look in the same region now where we looked for the vortex, we get the results shown in 4.26.

In the system where we only have one skyrmion-vortex pair, one Majorana mode localises at the pair and the other one localises at the edge. So far this is the same as for the system with only a vortex. Yet, the modes are more spatially separated than before. This means that the skyrmion in the ferromagnet enhances the localisation of the mode in the vortex. We have seen a similar result in chapter 4.1 where the inhomogeneous field also enhances the localisation of the modes. If we look at the superconducting gap, we see, in principle, the same as for the vortex.

In this system, we are also able to study the presence of two skyrmion-vortex pairs. Because of the attraction between the skyrmions and vortices, we are able to prevent the vortices from leaving the system. This also means that the attraction of the skyrmions and vortices is stronger than the vortex-vortex repulsion. We observe that in such a system each skyrmion-vortex pair carries a Majorana mode. The shape of these modes

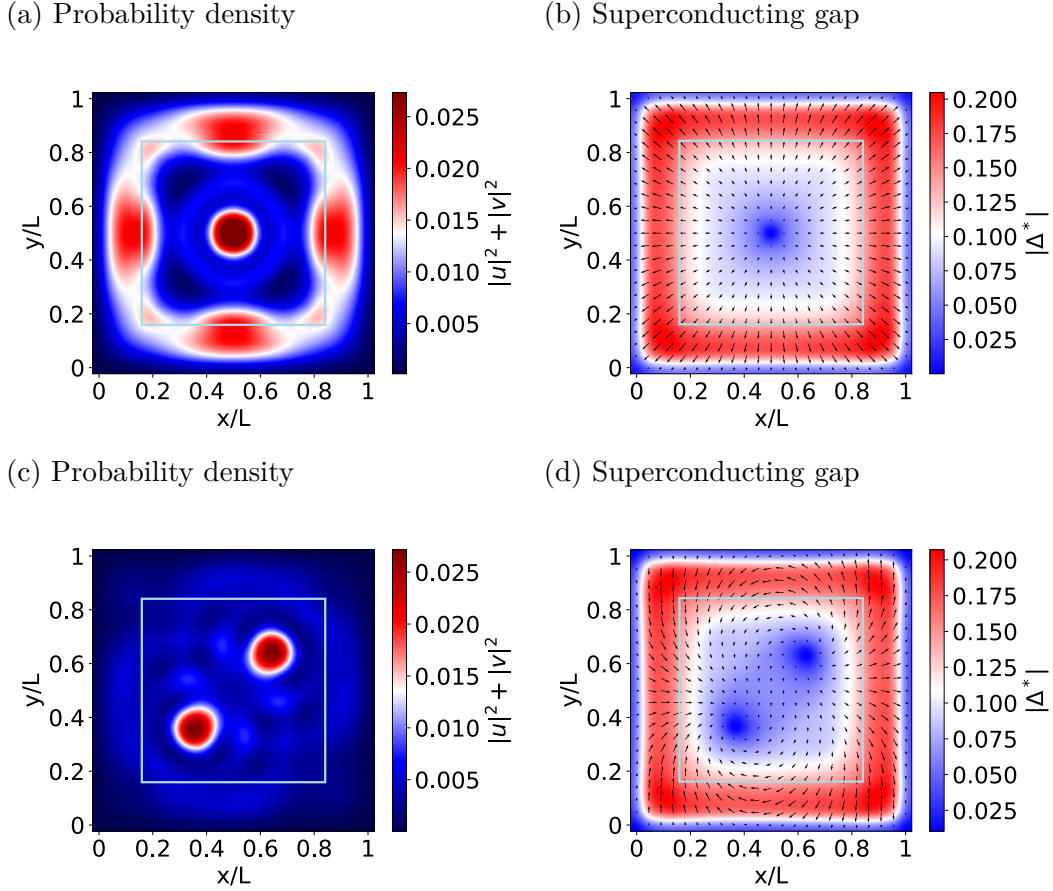


Figure 4.26.: (a) Probability density of the Majorana Modes. One can see that there is one mode localised at the vortex core and one at the edge. We scaled the one on the edge with 4 because it is hard to see since the probability is distributed over a bigger region. (b) Plot of the superconducting gap Δ^* . The arrows denote the phase of Δ^* and the size is scaled with the absolute value. The absolute value is also plotted with colours in the background. One can see that the superconducting gap goes to almost zero in the center of the vortex and is reduced in the region of the ferromagnet. (c) The same as in (a) but now with two skyrmion-vortex pairs. We see that every skyrmion-vortex pair carries one Majorana mode. (d) same as (b) but now with two skyrmion-vortex pairs. In each vortex, the superconducting gap goes almost to zero.

is not completely round due to the attractive interaction. This effect can be reduced by choosing a bigger separation of the skyrmion-vortex pairs. Looking at the superconducting gap, it can be observed that it goes close to zero for both skyrmion-vortex pairs but there is also an ellipsoid around the two pairs. Hence, it seems that they like to build an ellipsoid where the superconductivity is destroyed. This is also an effect of the proximity of the two pairs. Furthermore, the arrows now wind around the center of

the vortex; one winds in the right direction and the other one winds in the left direction. Since this corresponds to opposite vorticities, it appears that the system wants to reduce the vorticity of the system to minimize the energy. From another point of view, one can say that if we put two vortices with the same vorticity in the superconductor, as in 4.26 (b), then we would have two arrows that point in opposite directions in the middle of the sample. Since this corresponds to a non-continuous step in the phase of the superconducting gap this has a higher energy. Therefore when we let the system find its energy minimum, it wants to find a configuration where such problems will not occur. In Conclusion, we have found a system where a Majorana zero mode carried by skyrmion-vortex pairs could possibly be moved by electrical currents which is a desired property for building a scalable fault tolerant quantum computer. In the next chapter, we will summarize the main results of this thesis and explain what the next steps could be to go even further in the direction of building a topological quantum computer.

5

Conclusion and Outlook

In this thesis, we investigated the existence of Majorana zero modes in ferromagnet-superconductor heterostructures. By studying different configurations in the magnetization of the ferromagnet and in the superconducting gap, we demonstrated that there are many possibilities to realise Majorana zero modes in such systems. But, keeping in mind that the aim is to build a quantum computer, not every configuration is suitable. We analyzed these systems by either diagonalizing the corresponding Bogoliubov-de Gennes Hamiltonian, or by using the self-consistent field method presented in chapter 3.

In the beginning, we started with the simplest configuration where we considered a one-dimensional system with a ferromagnetic chain on a superconductor (chapter 4.1), which has been already realized experimentally [19]. There we saw that localized Majorana zero modes at the ends of the chain are obtained by either using spin-orbit coupling or by using an inhomogeneous field generated by the magnetization in the ferromagnet, and this opens a gap in the band structure. In this system, the self-consistent field method did not converge. To solve this problem, we considered a two dimensional superconductor (chapter 4.2), and we found the same results as for the fully one dimensional system by using the self-consistent field method. However, at the moment, the most probable way to move the edge modes is to build a T junction and move them by tuning the local chemical potential with time-dependent gate voltages [16], which is difficult to realise in an experiment.

As a next step, we moved to a fully two dimensional system (chapter 4.3) where we, at first, recovered the edge mode in a ferromagnetic block proposed in Ref. [33]. These modes are rather impractical because they are not spatially separated. Nevertheless, it can be used as a first hint that the considered system is topological.

Following this we considered the idea that there were also Majorana zero modes bound to vortices in superconductors [35, 36] and also recovered this literature result in our calculations (chapter 4.4). They are also movable by low temperature magnetic force microscopy [37]. In principle, it is therefore possible to check if these modes also have the proposed non-Abelian statistics. It needs to be noted though that this this approach will cause problems if we want to scale up the number of Majorana modes for an actual device.

A better configuration would then be to use Majorana modes bound to a skyrmion

(chapter 4.5), which was proposed in Ref. [20]. By using a more convenient definition for the skyrmion, we observed similar results by diagonalizing the Bogoliubov-de Gennes Hamiltonian of the corresponding system. Nevertheless, if we use the self-consistent field method suggests that such a situation is not stable solution of the system; instead of spatially separated modes, we are left with the modes not being spatially separated. This problem can be circumvented by exploiting the attractive interaction between skyrmions and vortices by building so-called skyrmion-vortex pairs [29]. We observed that it is possible to get localized Majorana zero modes in these Skyrmion-Vortex pairs (chapter 4.6). We expect that if we move the skyrmion it should also drag the vortex with it, thereby allowing us to braid these modes and check if they obey non-Abelian statistics. This brings us directly to the outlook of this thesis.

The next step beyond this thesis is to recover non-Abelian statistics of the Majorana modes in the skyrmion vortex pairs. In order to do so, we have to study the dynamics of the system and move them in the way proposed in Ref. [27]. We propose to use the configuration shown in figure 5.1 to realize this. This will also allow us to study

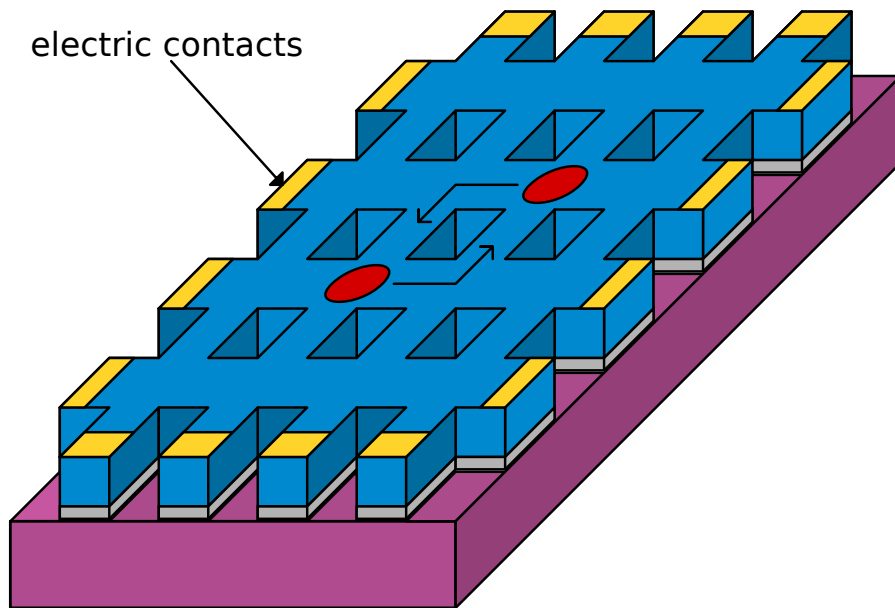


Figure 5.1.: Here we show a system where the non-Abelian statistics of skyrmion-vortex pairs could be tested. The blue material is a ferromagnet which hosts skyrmions and the purple material is a superconductor which hosts vortices. The yellow blocks are contacts which can be used to control the movement of the skyrmion-vortex pairs. The gray material is a non-magnetic insulator whose thickness controls the strength of the magnetic field in the superconductor produced by the ferromagnet. And the red ovals represent the skyrmion-vortex pairs.

the coherence of these operations. One way to extend the methodology described in this thesis is to calculate the magnetisation of two or more skyrmions by using micromagnetic simulations. Such a magnetisation configuration can then be included in the Bogoliubov-de Gennes equation as a magnetic field. Finally, there is still the problem that the ferromagnet destroys the superconductivity in the superconductor. An alternative approach to circumvent this is to instead use an antiferromagnet. We expect this to extend the region, where we obtain localized Majorana zero modes. For that, we have to study if there is any coupling between an antiferromagnet and a superconductor.

A

Appendix

A.1. Pairing

In [40] there is a really nice description of different pairings, which will help us throughout this thesis. For this reason we will show the most important arguments of it here. In BCS theory [30], the electron-phonon interaction leads to an effective attraction between electrons near the Fermi surface with opposite momenta and opposite spins, which eventually causes superconductivity. Since the interaction of electrons with phonons is isotropic in the k -space, the Cooper pairs are formed in a state with zero orbital angular momentum (s-wave pairing), which leads to a fully gapped superconducting state. On the other hand, unconventional superconductivity is mostly characterized by the anisotropic superconducting gap function with zeros (nodes) along certain directions in the momentum space. Thus, the superconducting gap structure is closely related to the pairing interaction responsible for the superconductivity. Therefore detailed knowledge of the gap structure will give a strong guide to establishing the pairing mechanism of superconductivity.

The superconducting gap function $\Delta_{s_1, s_2}^l(k)$, which is proportional to the amplitude of the wave function of a Cooper pair $\Psi_{s_1, s_2}^l(k) = \langle \psi_{k, s_1} \psi_{-k, s_2} \rangle$, serves as an order parameter system: it is non-zero only in the superconducting state. Here, k is the quasiparticle momentum, l is the orbital angular momentum, s_i is the electron spin, and ψ is the electron annihilation operator. In the simplest case where the spin-orbit coupling is negligible, the total angular momentum L and total spin $S = s_1 + s_2$ are good quantum numbers, and $\Psi_{s_1, s_2}^l(k)$ can be expressed in the form of a product of the orbital and spin parts,

$$\Psi_{s_1, s_2}^l(k) = g_l(k) \chi_s(s_1, s_2),$$

where $g_l(k)$ is the orbital wave function and $\chi_s(s_1, s_2)$ is the spin wave function. According to the Pauli's exclusion principle, the total wave function should change its sign under the exchange of two particles;

$$g_l(-k) \chi_s(s_2, s_1) = -g_l(k) \chi_s(s_1, s_2).$$

A. Appendix

The orbital part $g_l(k)$ can be expanded in terms of spherical harmonics $Y_{lm}(\hat{k})$, which are the eigenfunctions of the angular momentum operator with the momentum l and its z -projections m ,

$$g_l(k) = \sum_{m=-l}^m a_{lm}(k) Y_{lm}(\hat{k}),$$

where $\hat{k} = k/k_F$ represents the direction of momentum. $g_l(k)$ is even for even values of l and odd for odd values of l , $g_l(k) = (-1)^l g_l(k)$, and superconductors with $l = 0, 1, 2, \dots$ are labelled as s, p, d, \dots -wave, respectively. Hence, the spin component of a paired state with even (odd) orbital angular momentum l should be antisymmetric (symmetric) under the exchange of particles.

The spin wave function of the Cooper pair $\chi_s(s_1, s_2)$ is a product of the one-particle spin wave functions,

$$\psi_\uparrow = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |\uparrow\rangle \text{ and } \psi_\downarrow = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |\downarrow\rangle,$$

which are eigenstates of the operators s^2 and s_z :

$$s_z = \frac{\hbar}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad s_z \psi_\uparrow = \frac{\hbar}{2} \psi_\uparrow, \quad s_z \psi_\downarrow = -\frac{\hbar}{2} \psi_\downarrow$$

In the singlet state, $S = 0$, the spin part of the wave function is antisymmetric with respect to the particle exchange. Therefore, the eigenfunction corresponding to the spin singlet state can be given by

$$\psi_\uparrow \psi_\downarrow - \psi_\downarrow \psi_\uparrow = |\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = i\sigma_y,$$

where $\sigma_i (i = x, y, z)$ is the Pauli matrix,

$$\sigma_x = \frac{\hbar}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \frac{\hbar}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \frac{\hbar}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

As a result, the total wave function of the Cooper pair with $S = 0$ is given by

$$\Psi_{\text{singlet}}^l(k) = g_l(k) i\sigma_y,$$

where l is even.

For spin triplet pairing ($S=1$), the spin wave functions corresponding to the three

different spin projections on the quantization axis, which are symmetric under the exchange of particle, are given by

$$S_z = \begin{cases} 1, & \psi_\uparrow\psi_\uparrow = |\uparrow\uparrow\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ 0, & \psi_\uparrow\psi_\downarrow + \psi_\downarrow\psi_\uparrow = |\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ -1, & \psi_\downarrow\psi_\downarrow = |\downarrow\downarrow\rangle = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \end{cases}$$

Consequently, the total wave function can be written as

$$\Psi_{\text{triplet}}^l = g_{l,1} |\uparrow\uparrow\rangle + g_{l,0}(k)(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle) + g_{l,-1} |\downarrow\downarrow\rangle = \begin{pmatrix} g_{l,1} & g_{l,0} \\ g_{l,0} & g_{l,-1} \end{pmatrix},$$

where g_{l,S_z} is defined as the amplitudes of states,

$$g_{l,S_z} = \sum_{m=-l}^l a_{lm}^{S_z} Y_l^m(\hat{k}).$$

This equation can be rewritten as the following form by using the basis of the symmetric matrices $i\boldsymbol{\sigma}_y = (i\sigma_x\sigma_y, i\sigma_y\sigma_y, i\sigma_z\sigma_y)$,

$$\begin{aligned} \Psi_{\text{triplet}}^l &= (\mathbf{d}(k) \cdot \boldsymbol{\sigma}) i\sigma_y = (d_x(k)\sigma_x, d_y(k)\sigma_y, d_z(k)\sigma_z) i\sigma_y \\ &= \begin{pmatrix} -d_x(k) + id_y(k) & d_z(k) \\ d_z(k) & d_x(k) + id_y(k) \end{pmatrix} \end{aligned}$$

To summarize, the superconducting state can be characterized by its total spin $S = 0$ (spin-singlet) and $S = 1$ (spin-triplet). Thus, the superconducting gap functions for the singlet and triplet pairings are given by

$$\begin{aligned} \Delta_k^{\text{s-wave}} &= \Delta_0 g(k) i\sigma_y \\ \Delta_k^{\text{p-wave}} &= \Delta_0 (\mathbf{d}(k) \cdot \boldsymbol{\sigma}) i\sigma_y. \end{aligned}$$

For an s-wave superconductor we do not have any k dependence resulting in $g(k)$ being a constant which we set equal to one. For p-wave pairing there are now different choices: The most commonly used choice is $\mathbf{d}(k) = (k_x \pm ik_y, i(k_x \pm ik_y), 0)$ is called $p_x + ip_y$ pairing but we also use $\mathbf{d}(k) = (0, 0, k)$ in this thesis. These two choices come from the spherical harmonics $Y_l^m(\hat{k})$. For a p-wave superconductor we have that $l = 1$ and $m = 0, \pm 1$ which results in $Y_1^{\pm 1}(\hat{k}) \propto k_x \pm ik_y$ and $Y_1^0(\hat{k}) \propto k_z$.

Concluding the three different pairings mentioned in this thesis are

$$\Delta_k^{\text{s-wave}} \propto \Delta_0 (\psi_\uparrow\psi_\downarrow - \psi_\downarrow\psi_\uparrow) \quad (\text{A-1})$$

$$\Delta_k^{p_x + ip_y\text{-wave}} \propto \Delta_0 (k_x \pm k_y) \psi_\uparrow\psi_\uparrow \quad (\text{A-2})$$

$$\Delta_k^{\text{p-wave}} \propto \Delta_0 k (\psi_\uparrow\psi_\downarrow + \psi_\downarrow\psi_\uparrow). \quad (\text{A-3})$$

A.2. Programs to produce Plots in Chapter 2

The following code was used to produce the plots in figure 2.3 and 2.4.

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib
4
5  # =====
6  # Definitions and functions
7
8  # Sets the size of the plot
9  matplotlib.rcParams.update({'font.size':20})
10
11 # Define Pauli Matrices
12 pauli = [[[1, 0],[0, 1]],[[0, 1],[1, 0]],[[0,-1j],[1j, 0]],[[1,
13         0],[0,-1]]]
14
15 def SideDiag(n,l):
16     """
17     Purpose:
18     Makes a Matrix which has only entries at the side diagonal
19
20     Definitions:
21     l: Parameter that controls if we have entries on the right (-1) or on
22         the left (1)
23     n: Parameter defining how many sites the system has
24     """
25     a = np.zeros(shape=(n,n))
26     for j in range(n):
27         for i in range(n):
28             if j+l <= i <= j+l:
29                 a[i][j] = 1
30     return a
31
32 def MuMatrix(mu,n):
33     """
34     Purpose:
35     Makes a Matrix corresponding to the onsite part in the Hamiltonian
36
37     Definitions:
38     mu: Parameter for controlling the chemical potential of the system
39     n: Parameter defining how many sites the system has
40     """
41     return -mu*np.tensordot(pauli[3],np.identity(n),axes=0)
42
43 def HoppingMatrix(n):
44     """
45     Purpose:
46     Makes a Matrix corresponding to the hopping part in the Hamiltonian.
47
48     Comment:
```

```

47 All energies are scaled in terms of the hopping parameter t, therefore
    we choose it equals 1 here.
48
49 Definitions:
50 n: Parameter defining how many sites the system has
51 """
52 return -np.tensordot(pauli[3], SideDiag(n, -1), axes=0) - np.tensordot(pauli
    [3], SideDiag(n, 1), axes=0)
53
54 def DeltaMatrix(D, n):
55     """
56     Purpose:
57     Makes a Matrix corresponding to the pairing part in the Hamiltonian
58
59     Definitions:
60     D: Parameter for controlling the BCS type pairing of electrons
61     n: Parameter defining how many sites the system has
62     """
63     return -D*1j*np.tensordot(pauli[2], SideDiag(n, -1), axes=0) + D*1j*np.
        tensordot(pauli[2], SideDiag(n, 1), axes=0)
64
65 def H(n, mu, D):
66     """
67     Purpose:
68     Puts together the different parts to a full Hamiltonian
69
70     Definitions:
71     D: Parameter for controlling the BCS type pairing of electrons
72     mu: Parameter for controlling the chemical potential of the system
73     n: Parameter defining how many sites the system has
74     """
75     tmp = MuMatrix(mu, n) + HoppingMatrix(n) + DeltaMatrix(D, n)
76     return np.row_stack((np.column_stack((tmp[0, 0], tmp[0, 1])), np.
        column_stack((tmp[1, 0], tmp[1, 1]))))
77
78 def Eigensystem_H(n, mu, D):
79     """
80     Purpose:
81     Calculates the Eigenvectors and Eigenvalues of the Hamiltonian
82
83     Definitions:
84     D: Parameter for controlling the BCS type pairing of electrons
85     mu: Parameter for controlling the chemical potential of the system
86     n: Parameter defining how many sites the system has
87     """
88     ev, evec = np.linalg.eigh(H(n, mu, D))
89     return ev, evec
90
91 def PlotEigenvalues_H(n, mus, D):
92     """
93     Purpose:
94     Plots the Eigenvalues

```

A. Appendix

```
95
96 Definitions:
97 D: Parameter for controlling the BCS type pairing of electrons
98 mu: Parameter for controlling the chemical potential of the system
99 n: Parameter defining how many sites the system has
100 """
101 EVa=[]
102
103 for mu in mus:
104     ev, evec = Eigensystem_H(n,mu,D)
105     EVa.append(ev)
106
107 for i in range(2*n):
108     plt.plot(mus,np.transpose(EVa)[i], 'b-')
109
110 plt.xlim(0,4)
111 plt.xlabel(r"$\mu$/t")
112 plt.ylabel(r"E/t")
113 plt.tight_layout()
114
115 def PlotEigenvectors_H(n,mu,D):
116     """
117     Purpose:
118     Plots the Eigenvectors
119
120     Definitions:
121     D: Parameter for controlling the BCS type pairing of electrons
122     mu: Parameter for controlling the chemical potential of the system
123     n: Parameter defining how many sites the system has
124     """
125     ev, evec = Eigensystem_H(n,mu,D)
126
127     # Calculating the Probability density of the lowest energy modes
128     wf=evec[:,n]**2+evec[:,n-1]**2
129     # Calculating the Probability density of the next higher energy modes
130     wf_b=evec[:,n+1]**2+evec[:,n-2]**2
131
132     x = np.linspace(0,n-1,n)/(n-1)
133
134     plt.plot(x,wf[0:n], 'b-')
135     plt.plot(x,wf_b[0:n], 'r--')
136     plt.xlim(-0.003,1)
137     plt.ylim(0,0.5)
138     plt.xlabel("x/L")
139     plt.ylabel(r"$\left|u\right|^2+\left|v\right|^2$")
140     plt.tight_layout()
141
142     # =====
143     # Plot the Energie spectrum and the Probability Density of the
144     # Hamiltonian using the definitions above
145
146     n = 25
```

```
146 mus = np.linspace(0, 4, 1000)
147 D = 1
148
149 PlotEigenvalues_H(n,mus,D)
150 PlotEigenvectors_H(n,4,D)
```

A. Appendix

The following code was used to produce the plots in figure 2.5.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib
4
5 # =====
6 # Definitions and functions
7
8 # Sets the size of the plot
9 matplotlib.rcParams.update({'font.size':20})
10
11 # Define Pauli Matrices
12 pauli = [[[1, 0],[0, 1]],[[0, 1],[1, 0]],[[0,-1j],[1j, 0]],[[1,
13         0],[0,-1]]]
14
15 def MuMatrix(k,mu):
16     """
17     Purpose:
18     Makes a Matrix corresponding to the onsite part in the Hamiltonian
19
20     Definitions:
21     k: Parameter that controls the momentum of the particles
22     mu: Parameter for controlling the chemical potential of the system
23     """
24     return (-2*np.cos(k)-mu)*np.tensordot(pauli[3],np.identity(2),axes=0)
25
26 def DeltaMatrix(D,k):
27     """
28     Purpose:
29     Makes a Matrix corresponding to the pairing part in the Hamiltonian
30
31     Definitions:
32     D: Parameter for controlling the BCS type pairing of electrons
33     k: Parameter that controls the momentum of the particles
34     """
35     return 2*D*np.sin(k)*np.tensordot(pauli[2],np.identity(2),axes=0)
36
37 def H_Bulk(k,mu,D):
38     """
39     Purpose:
40     Puts together the different parts to a full Hamiltonian
41
42     Definitions:
43     k: Parameter that controls the momentum of the particles
44     mu: Parameter for controlling the chemical potential of the system
45     D: Parameter for controlling the BCS type pairing of electrons
46     B: Parameter which controls the magnetic field
47     """
48     tmp = MuMatrix(k,mu)+DeltaMatrix(D,k)
49     return np.row_stack((np.column_stack((tmp[0,0],tmp[0,1])),np.
50         column_stack((tmp[1,0],tmp[1,1]))))
```



```

50 def Eigensystem_H_Bulk(k,mu,D):
51     """
52     Purpose:
53     Calculates the Eigenvectors and Eigenvalues of the Bulk Hamiltonian in
54         momentum space
55     Definitions:
56     k: Parameter that controls the momentum of the particles
57     mu: Parameter for controlling the chemical potential of the system
58     D: Parameter for controlling the BCS type pairing of electrons
59     B: Parameter which controls the magnetic field
60     """
61     ev, evec = np.linalg.eigh(H_Bulk(k,mu,D))
62     return ev, evec
63
64 def PlotEigenvalues_H_Bulk(ks,mu,D):
65     """
66     Purpose:
67     Plots the Eigenvectors
68
69     Definitions:
70     k: Parameter that controls the momentum of the particles
71     mu: Parameter for controlling the chemical potential of the system
72     D: Parameter for controlling the BCS type pairing of electrons
73     B: Parameter which controls the magnetic field
74     """
75     fig, ax = plt.subplots()
76
77     EVa=[]
78
79     for k in ks:
80         ev, evec = Eigensystem_H_Bulk(k,mu,D)
81         EVa.append(ev)
82
83     ax.plot(ks,np.transpose(EVa)[0],'b-')
84     ax.plot(ks,np.transpose(EVa)[1],'b-')
85     ax.plot(ks,np.transpose(EVa)[2],'b-')
86     ax.plot(ks,np.transpose(EVa)[3],'b-')
87
88     plt.xlim(ks[0],ks[-1])
89     plt.ylim(-4,4)
90     plt.xlabel("k")
91     plt.ylabel("E/t")
92
93     # Making radian ticks
94     x_ticks = np.arange(-1,1.5,0.5)
95     x_label = [r"$-\pi$", r"$-\frac{\pi}{2}$", r"$0$", r"$\frac{\pi}{2}$",
96               r"$\pi$"]
97     ax.set_xticks(x_ticks*np.pi)
98     ax.set_xticklabels(x_label, fontsize=20)
99     plt.tight_layout()

```

A. Appendix

```
100
101 def PlotdVector(ks,mu,D):
102     """
103     Purpose:
104     Plots the Eigenvectors
105
106     Definitions:
107     k: Parameter that controls the momentum of the particles
108     mu: Parameter for controlling the chemical potential of the system
109     D: Parameter for controlling the BCS type pairing of electrons
110     B: Parameter which controls the magnetic field
111     """
112     dy = []
113     dz = []
114
115     for k in ks:
116         dy.append(2*D*np.sin(k))
117         dz.append(-2*np.cos(k)-mu)
118
119     plt.plot(dz,dy,'b-')
120     plt.xlabel("$d_z/t$")
121     plt.ylabel("$d_y/t$")
122     plt.axis('equal')
123     plt.xlim(-5.1,5.1)
124     plt.ylim(-5.1,5.1)
125     plt.axhline(0,color='k')
126     plt.axvline(0,color='k')
127     plt.tight_layout()
128
129     # =====
130     # Plot the Energie spectrum of the Bulk Hamiltonian using the
131         definitions above
132
133     ks = np.linspace(-np.pi,np.pi,1000)
134     mu = 0
135     D = 1
136
137     PlotEigenvalues_H_Bulk(ks,mu,D)
138     PlotdVector(ks,mu,D)
```

The following code was used to produce the plots in figure 2.6 and 2.7.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib
4
5  # =====
6  # Definitions and functions
7
8  # Sets the size of the plot
9  matplotlib.rcParams.update({'font.size':20})
10
11 # Define Pauli Matrices
12 pauli = [[[1, 0],[0, 1]],[[0, 1],[1, 0]],[[0,-1j],[1j, 0]],[[1,
13         0],[0,-1]]]
14 def MuMatrix(k,mu):
15     """
16     Purpose:
17     Makes a Matrix corresponding to the onsite part in the Hamiltonian
18
19     Definitions:
20     k: Parameter that controls the momentum of the particles
21     mu: Parameter for controlling the chemical potential of the system
22     """
23     return (k**2-mu)*np.tensordot(pauli[3],np.identity(2),axes=0)
24
25 def DeltaMatrix(D,k):
26     """
27     Purpose:
28     Makes a Matrix corresponding to the pairing part in the Hamiltonian
29
30     Definitions:
31     D: Parameter for controlling the BCS type pairing of electrons
32     k: Parameter that controls the momentum of the particles
33     """
34     return 2*D*k*np.tensordot(pauli[2],np.identity(2),axes=0)
35
36 def MagneticFieldMatrix(B):
37     """
38     Purpose:
39     Makes a Matrix corresponding to the hopping part in the Hamiltonian.
40
41     Comment:
42     All energies are scaled in terms of the hopping parameter t, therefore
43     we choose it equals 1 here.
44
45     Definitions:
46     B: Parameter which controls the magnetic field
47     """
48     return -B*np.tensordot(pauli[3],pauli[3],axes=0)
49 def H_Bulk(k,mu,D,B):

```

A. Appendix

```
50 """
51 Purpose:
52 Puts together the different parts to a full Hamiltonian
53
54 Definitions:
55 k: Parameter that controls the momentum of the particles
56 mu: Parameter for controlling the chemical potential of the system
57 D: Parameter for controlling the BCS type pairing of electrons
58 B: Parameter which controls the magnetic field
59 """
60 tmp = MuMatrix(k,mu)+DeltaMatrix(D,k)+MagneticFieldMatrix(B)
61 return np.row_stack((np.column_stack((tmp[0,0],tmp[0,1])),np.
        column_stack((tmp[1,0],tmp[1,1]))))
62
63 def Eigensystem_H_Bulk(k,mu,D,B):
64 """
65 Purpose:
66 Calculates the Eigenvectors and Eigenvalues of the Bulk Hamiltonian in
        momentum space
67
68 Definitions:
69 k: Parameter that controls the momentum of the particles
70 mu: Parameter for controlling the chemical potential of the system
71 D: Parameter for controlling the BCS type pairing of electrons
72 B: Parameter which controls the magnetic field
73 """
74 ev, evec = np.linalg.eigh(H_Bulk(k,mu,D,B))
75 return ev, evec
76
77 def PlotEigenvalues_H_Bulk(ks,mu,D,B):
78 """
79 Purpose:
80 Plots the Eigenvectors
81
82 Definitions:
83 k: Parameter that controls the momentum of the particles
84 mu: Parameter for controlling the chemical potential of the system
85 D: Parameter for controlling the BCS type pairing of electrons
86 B: Parameter which controls the magnetic field
87 """
88 EVa=[]
89
90 for k in ks:
91 ev, evec = Eigensystem_H_Bulk(k,mu,D,B)
92 EVa.append(ev)
93
94 plt.plot(ks,np.transpose(EVa)[0], 'b-')
95 plt.plot(ks,np.transpose(EVa)[1], 'b-')
96 plt.plot(ks,np.transpose(EVa)[2], 'b-')
97 plt.plot(ks,np.transpose(EVa)[3], 'b-')
98
99 plt.xlim(ks[0],ks[-1])
```

```
100 plt.xlabel("k")
101 plt.ylabel("E/t")
102 plt.xticks([-1,-0.5,0,0.5,1])
103 plt.tight_layout()
104
105 # =====
106 # Plot the Energie spectrum of the Bulk Hamiltonian using the
    definitions above
107
108 ks = np.linspace(-1,1,1000)
109 mu = 0.3
110 D = 0.1
111 B = 0.4
112
113 PlotEigenvalues_H_Bulk(ks,mu,D,B)
```

A. Appendix

The following code was used to produce the plots in figure 2.8 and 2.9.

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib
4
5  # =====
6  # Definitions and functions
7
8  # Sets the size of the plot
9  matplotlib.rcParams.update({'font.size':20})
10
11 # Define Pauli Matrices
12 pauli = [[[1, 0],[0, 1]],[[0, 1],[1, 0]],[[0,-1j],[1j, 0]],[[1,
13           0],[0,-1]]]
14
15 def MuMatrix(k,mu):
16     """
17     Purpose:
18     Makes a Matrix corresponding to the onsite part in the Hamiltonian
19
20     Definitions:
21     k: Parameter that controls the momentum of the particles
22     mu: Parameter for controlling the chemical potential of the system
23     """
24     return (k**2-mu)*np.tensordot(pauli[3],np.identity(2),axes=0)
25
26 def DeltaMatrix(D):
27     """
28     Purpose:
29     Makes a Matrix corresponding to the pairing part in the Hamiltonian
30
31     Definitions:
32     D: Parameter for controlling the BCS type pairing of electrons
33     """
34     return D*np.tensordot(pauli[1],np.identity(2),axes=0)
35
36 def MagneticFieldMatrix(B):
37     """
38     Purpose:
39     Makes a Matrix corresponding to the hopping part in the Hamiltonian.
40
41     Comment:
42     All energies are scaled in terms of the hopping parameter t, therefore
43     we choose it equals 1 here.
44
45     Definitions:
46     B: Parameter which controls the magnetic field
47     """
48     return B*np.tensordot(np.identity(2),pauli[3],axes=0)
49
50 def SpinOrbitCouplingMatrix(alpha,k):
51     """
```

```

50 Purpose:
51 Makes a Matrix corresponding to the spin-orbit coupling part in the
    Hamiltonian.
52
53 Comment:
54 All energies are scaled in terms of the hopping parameter t, therefore
    we choose it equals 1 here.
55
56 Definitions:
57 alpha: Parameter which controls the spin orbit coupling
58 k: Parameter that controls the momentum of the particles
59 """
60 return alpha*k*np.tensordot(pauli[3],pauli[2],axes=0)
61
62 def H_Bulk(k,mu,D,B,alpha):
63     """
64     Purpose:
65     Puts together the different parts to a full Hamiltonian
66
67     Definitions:
68     k: Parameter that controls the momentum of the particles
69     mu: Parameter for controlling the chemical potential of the system
70     D: Parameter for controlling the BCS type pairing of electrons
71     B: Parameter which controls the magnetic field
72     """
73     tmp = MuMatrix(k,mu)+DeltaMatrix(D)+MagneticFieldMatrix(B)+
        SpinOrbitCouplingMatrix(alpha,k)
74     return np.row_stack((np.column_stack((tmp[0,0],tmp[0,1])),np.
        column_stack((tmp[1,0],tmp[1,1]))))
75
76 def Eigensystem_H_Bulk(k,mu,D,B,alpha):
77     """
78     Purpose:
79     Calculates the Eigenvectors and Eigenvalues of the Bulk Hamiltonian in
        momentum space
80
81     Definitions:
82     k: Parameter that controls the momentum of the particles
83     mu: Parameter for controlling the chemical potential of the system
84     D: Parameter for controlling the BCS type pairing of electrons
85     B: Parameter which controls the magnetic field
86     """
87     ev, evec = np.linalg.eigh(H_Bulk(k,mu,D,B,alpha))
88     return ev, evec
89
90 def PlotEigenvalues_H_Bulk(ks,mu,D,B,alpha):
91     """
92     Purpose:
93     Plots the Eigenvectors
94
95     Definitions:
96     k: Parameter that controls the momentum of the particles

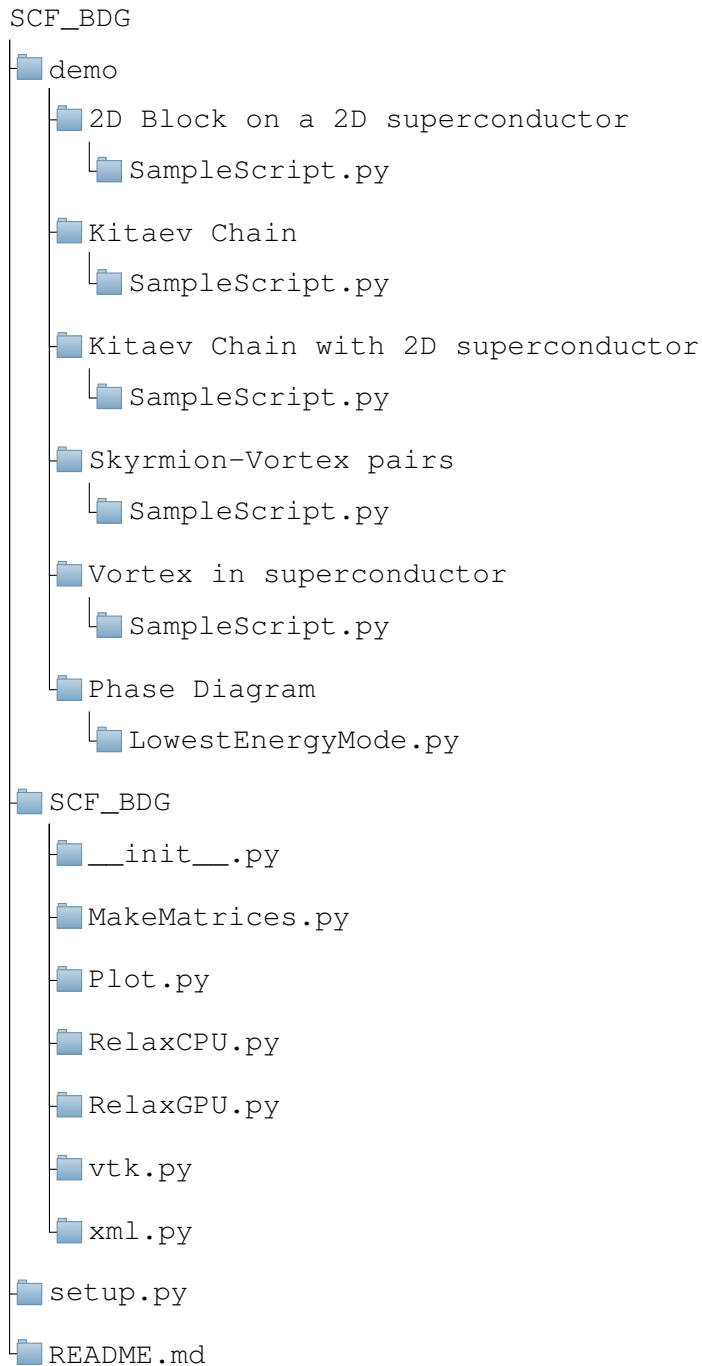
```

A. Appendix

```
97 mu: Parameter for controlling the chemical potential of the system
98 D: Parameter for controlling the BCS type pairing of electrons
99 B: Parameter which controls the magnetic field
100 """
101 EVa=[]
102
103 for k in ks:
104     ev, evec = Eigensystem_H_Bulk(k,mu,D,B,alpha)
105     EVa.append(ev)
106
107 plt.plot(ks,np.transpose(EVa)[0], 'b-')
108 plt.plot(ks,np.transpose(EVa)[1], 'b-')
109 plt.plot(ks,np.transpose(EVa)[2], 'b-')
110 plt.plot(ks,np.transpose(EVa)[3], 'b-')
111
112 plt.xlim(ks[0],ks[-1])
113 plt.xlabel("k")
114 plt.ylabel("E/t")
115 plt.xticks([-1,-0.5,0,0.5,1])
116 plt.tight_layout()
117
118 # =====
119 # Plot the Energie spectrum of the Bulk Hamiltonian using the
    definitions above
120
121 ks = np.linspace(-1,1,1000)
122 mu = 0
123 D = 0.1
124 B = 0.05
125 alpha = 0.3
126
127 PlotEigenvalues_H_Bulk(ks,mu,D,B,alpha)
```


A.3. Program for the method in Chapter 3

The package that was used to realize the method is structured as follows:



In the folder `demo` are the scripts which were used to produce the results that we showed in chapter 4. They also serve as an orientation on how to use the package. In the folder `SCF_BDG` the actual package is placed, consisting mainly out of the `MakeMatrices.py` file which we will show in the following and will handle the Bogoliubov-de Gennes Hamiltonian. The file `Plot.py` was created to simplify the procedure to plot the results and we will not show it here. One can have a look at it if interested in the github repository [41] which we will publish in the future. The heart of this package is the `RelaxCPU.py` and `RelaxGPU.py` files where the self-consistent solution will be calculated. For the GPU version we used the scikit-cuda [42] library. The two files `vtk.py` and `xml.py` are used to export the profile of the magnetization to a vtk file so that we can have a look at it with paraview [43]. We will not show them here. The `setup.py` file is used to install the whole project and the `README.md` file gives informations on how the installation could be done. The whole code was written on the basis of a Matlab code written by Kjetil Hals used in [29].

We will show a few of the files contained in the package. At first we will show `MakeMatrices.py`.

```

1 import numpy as np
2 import pickle
3
4 class MakeMatrix:
5     """
6     A class that helps to build the Bogoliubov-de Gennes Hamiltonian.
7     The Matrix is build the following way
8     (H      | Delta)
9     (Delta| H      )
10    The matrix H and Delta will be build by the functions contained in this
        class.
11    """
12    #Constants:
13    hbar= 1.05457173e-34 # hbar
14    me= 9.10938e-31 # electron mass
15    a=1e-6 # lattice constant
16    energyScale= hbar*hbar/(2*me*a*a) #Hopping energy in absence of SOC
17
18    #internal parameters
19    s0 = np.array([[1, 0],[ 0, 1]])
20    s1 = np.array([[0, 1],[ 1, 0]])
21    s2 = np.array([[0, -1j],[1j, 0]])
22    s3 = np.array([[1, 0],[0, -1]])
23    p = np.array([[0, 1],[-1, 0]])
24    g = 5 #coupling strength of cooper pairs
25    thermE = 0.001 #thermal energy, kB*T
26    epsilonD = 2.0 #Debye frequency
27    nrM = 5 #number of positive Eigenvalues that are printed
28
29    def __init__(self):
30        self.V0=0
31        self.tx=0
32        self.ty=0

```

```
33
34 #boundary conditions
35 self.bx=0
36 self.by=0
37
38 #parameters
39 self.Nx=0
40 self.Ny=0
41 self.alphaR=0
42 self.mu=0
43 self.h0=0
44
45 #difference in points between magnet and superconductor on boundary
46 self.randPointsX=0
47 self.randPointsY=0
48
49 #Positions of skyrmions
50 self.Srx=0
51 self.Sry=0
52
53 #Skyrmion Profile
54 self.VX=0
55 self.VY=0
56 self.VZ=0
57
58 #Polarity of Skyrmion
59 self.Sq=0
60
61 #Number of windings along r Skyrmion
62 self.Sp=1
63
64 #Winding number of Skyrmion
65 self.Sw=1
66
67 #thetaSkyrm= 0 ("Néel skyrmion"); thetaSkyrm= pi/2 (Bloch skyrmion)
68 self.thetaSkyrm=0
69
70 #Positions of vortices
71 self.Vrx=0
72 self.Vry=0
73
74 #Vorticity of Vortex
75 self.Vq=0
76
77 #thetaVortex= 0 ("Néel" vortex); thetaVortex= pi/2 ("Bloch" vortex)
78 self.thetaVortex=0
79
80 #initial delta
81 self.delta=0
82 self.deltainitial=0
83 self.deltan=0
84
```

A. Appendix

```
85     #Matrices
86     self.DeltaMatrix=0
87     self.h=0
88     self.Hmatrix=0
89     self.EnergyMatrix=0
90     self.DensityMatrix=0
91
92     #EigenValues/Eigenvectors
93     self.EigenValues=0
94     self.EigenValuesP=0
95     self.EigenVectors=0
96     self.EigenVectorsU=0
97     self.EigenVectorsV=0
98
99     #Majorana Densities
100    self.M1=0
101    self.M2=0
102
103    #Current
104    self.jx=0
105    self.jy=0
106    self.SOC=0
107
108    def MapFromVectorToMatrix(self,v):
109        """
110        Purpose: Converts a vector of length Nx*Ny into its corresponding Nx
111                times
112                Ny matrix
113        """
114        m= np.zeros(shape=(self.Nx,self.Ny), dtype=complex);
115
116        for i in range(self.Nx):
117            for j in range(self.Ny):
118                n= i + j*self.Nx
119                m[i,j]= v[n]
120
121        return m
122
123    def MakeConstantMatrices(self):
124        """
125        Purpose:
126        Make the 2 times 2 constant (dimensionless) matrices that appear in
127                the discretized
128                version of the (scaled) Rashba Hamiltonian.
129
130        Definitions:
131        alpha: dimensionless Rashba parameter.
132        mu: dimensionless chemical potential.
133        a: lattice spacing
134        """
```

```

135     #Write matrices:
136     self.V0= (4*0 - self.mu)*self.s0
137     self.tx= -1*self.s0 + 1j*self.alphaR*self.s2
138     self.ty= -1*self.s0 - 1j*self.alphaR*self.s1
139
140     def MakeDeltaMatrix(self):
141         """
142         Purpose:
143         The function makes the discretized version of the pairing potential.
144
145         Definitions:
146         delta: vector containing the on-site pairing potential (Nx*Ny
147             elements).
148         Nx, Ny: Number of lattice points along the x and y directions.
149         """
150         N= 2*self.Nx*self.Ny
151         DeltaMatrix= np.zeros(shape=(N, N),dtype=complex)
152
153         for i in range(self.Nx):
154             for j in range(self.Ny):
155                 n= 2*i + 2*j*self.Nx
156                 nd= i + j*self.Nx
157                 DeltaMatrix[n:n+2,n:n+2]= self.delta[nd]*self.p
158
159         self.DeltaMatrix=DeltaMatrix
160
161     def MakeVortices(self):
162         """
163         Purpose: initiate pairing potential with k vortices located at (rxV,
164             ryV) with a vorticity Vq= -1/+1 and magnitude Delta0.
165         """
166         for k in range(len(self.Vrx)):
167             for i in range(self.Nx):
168                 for j in range(self.Ny):
169                     n= i + j*self.Nx
170                     vx= np.double(i-self.Vrx[k])
171                     vy= np.double(j-self.Vry[k])
172                     v= [vx, vy]
173                     NormV= np.linalg.norm(v)
174                     if (NormV > 0.0):
175                         v= v/NormV
176                         phase= np.exp(1j*self.Vq*(np.angle(complex(v[0],v[1]))+self.
177                             thetaVortex))
178                         self.delta[n]= self.delta[n]*phase
179
180     def MakeFerroBlock(self):
181         """
182         Purpose: Make a ferromagnetic block. Which is one part of the H
183             matrix
184         """

```

A. Appendix

```
183
184     h= np.zeros(3*self.Nx*self.Ny, dtype=complex)
185     x1= self.randPointsX
186     x2= self.Nx - self.randPointsX
187     y1= self.randPointsY
188     y2= self.Ny - self.randPointsY
189
190     VX= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
191     VY= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
192     VZ= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
193
194     for i in range(x1, x2):
195         for j in range(y1, y2):
196             nh= 3*i + 3*j*self.Nx
197             h[nh]= 0
198             h[nh+1]= 0
199             h[nh+2]= self.h0
200             VX[i, j]= h[nh]
201             VY[i, j]= h[nh+1]
202             VZ[i, j]= h[nh+2]
203
204     self.h= h
205     self.VX=VX
206     self.VY=VY
207     self.VZ=VZ
208
209     def MakeInhomogeneousBlock(self):
210         """
211         Purpose: Make a ferromagnetic block with inhomogeneous magnetisation.
212                 Which is one part auf the H matrix
213         """
214         h= np.zeros(3*self.Nx*self.Ny, dtype=complex)
215         x1= self.randPointsX
216         x2= self.Nx - self.randPointsX
217         y1= self.randPointsY
218         y2= self.Ny - self.randPointsY
219
220         VX= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
221         VY= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
222         VZ= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
223
224         for i in range(x1, x2):
225             for j in range(y1, y2):
226                 nh= 3*i + 3*j*self.Nx
227                 h[nh]= 0*np.sin(4*2*np.pi*(i-self.randPointsX)/(self.Nx-2*self.
228                     randPointsX-1))
229                 h[nh+1]= np.sin(4*2*np.pi*(i-self.randPointsX)/(self.Nx-2*self.
230                     randPointsX-1))
231                 h[nh+2]= np.cos(4*2*np.pi*(i-self.randPointsX)/(self.Nx-2*self.
232                     randPointsX-1))
233                 VX[i, j]= h[nh]
```

```

231         VY[i, j]= h[nh+1]
232         VZ[i, j]= h[nh+2]
233
234     self.h= self.h0*h
235     self.VX=VX
236     self.VY=VY
237     self.VZ=VZ
238
239     def MakeSkyrmTextures(self):
240         """
241         Purpose: Make a skyrmion texture. Which is a part of the H matrix
242         If thetaSkyrm=0, the function returns Neel skyrmion, if thetaSkyrm=pi
243         /2 the function return a Bloch skyrmion.
244         """
245         x1= self.randPointsX
246         x2= self.Nx - self.randPointsX
247         y1= self.randPointsY
248         y2= self.Ny - self.randPointsY
249
250         size1= x2 - x1
251         size2= y2 - y1
252
253         l1= 0.1*np.double(min([size1, size2]))
254
255         h= np.zeros(3*self.Nx*self.Ny)
256
257         VX= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
258         VY= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
259         VZ= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
260
261         for k in range(len(self.Srx)):
262             for i in range(x1, x2):
263                 for j in range(y1, y2):
264                     nh= 3*i + 3*j*self.Nx
265                     x= np.double(i-self.Srx[k])
266                     y= np.double(j-self.Sry[k])
267                     z= x**2 + y**2 - l1**2
268                     N= x**2 + y**2 + l1**2
269                     mxy= np.sin(self.Sp*np.arccos(z/N))
270                     mz= -self.Sq*np.cos(self.Sp*np.arccos(z/N))
271                     if (i==x1 or j==y1 or i==x2-1 or j==y2-1):
272                         h[nh]= 0.0
273                         h[nh+1]= 0.0
274                         h[nh+2]= -self.Sq
275                     else:
276                         h[nh]= h[nh]+ np.sin(self.Sw*np.arctan2(x, y)+self.thetaSkyrm)
277                             *mxy
278                         h[nh+1]= h[nh+1]+ np.cos(self.Sw*np.arctan2(x, y)+self.
279                             thetaSkyrm) *mxy
280                         h[nh+2]= h[nh+2]+ mz
281                     VX[i, j]= h[nh]

```

A. Appendix

```
280         VY[i,j]= h[nh+1]
281         VZ[i,j]= h[nh+2]
282
283     self.h= self.h0*h/len(self.Srx)
284     self.VX=VX/len(self.Srx)
285     self.VY=VY/len(self.Srx)
286     self.VZ=VZ/len(self.Srx)
287
288     def MakeHMatrix(self):
289         """
290         Purpose:
291         The function returns the discretized version of the Rashba
292             Hamiltonian with a position dependent exchange field.
293
294         Definitions:
295         V0: on-site potential proportional to the 2 times 2 identity matrix (
296             contains chemical potential).
297         tx,ty: 2 times 2 hopping matrices (contains Rashba SOC).
298         h: exchange field (3*Nx*Ny elements).
299         Nx, Ny: Number of lattice points along the x and y directions.
300         bx=0 if open boundary condition along x-boundary, bx=1 if periodic
301             boundary condition.
302         """
303
304         #Initiate H matrix:
305         N= 2*self.Nx*self.Ny
306         Hmatrix= np.zeros(shape=(N, N),dtype=complex)
307
308         #Define daggered matrices:
309         txd= np.conj(np.transpose(self.tx))
310         tyd= np.conj(np.transpose(self.ty))
311
312         #Write Hmatrix:
313         for i in range(self.Nx):
314             for j in range(self.Ny):
315                 #Define indices:
316                 n= 2*i + 2*j*self.Nx
317                 ip= i+1
318                 im= i-1
319                 jp= j+1
320                 jm= j-1
321                 nip= 2*ip + 2*j*self.Nx
322                 nim= 2*im + 2*j*self.Nx
323                 njp= 2*i + 2*jp*self.Nx
324                 njm= 2*i + 2*jm*self.Nx
325                 nh= 3*i + 3*j*self.Nx
326                 #Write diagonal elements:
327                 htemp= self.h[nh:nh+3]
328                 V= self.V0 + htemp[0]*self.s1 + htemp[1]*self.s2 + htemp[2]*self.
329                     s3
330                 Hmatrix[n:n+2,n:n+2]= V
331                 #Write off-diagonal elements:
```



```

328     if (i>0 and i<self.Nx-1 and j>0 and j<self.Ny-1): #internal
329         lattice points
330         Hmatrix[n:n+2,nip:nip+2]= self.tx
331         Hmatrix[n:n+2,nim:nim+2]= txd
332         Hmatrix[n:n+2,njp:njp+2]= self.ty
333         Hmatrix[n:n+2,njm:njm+2]= tyd
334     elif (i==0 and j==0): #corner points
335         nim= 2*i + 2*j*self.Nx
336         njm= 2*i + 2*j*self.Nx
337         if(i<self.Nx-1):
338             Hmatrix[n:n+2,nip:nip+2]= self.tx
339             if(j<self.Ny-1):
340                 Hmatrix[n:n+2,njp:njp+2]= self.ty
341     elif (i==0 and j==self.Ny-1):
342         nim= 2*i + 2*j*self.Nx
343         njp= 2*i + 2*j*self.Nx
344         if(i<self.Nx-1):
345             Hmatrix[n:n+2,nip:nip+2]= self.tx
346             Hmatrix[n:n+2,njm:njm+2]= tyd
347     elif (j==0 and i==self.Nx-1):
348         nip= 2*i + 2*j*self.Nx
349         njm= 2*i + 2*j*self.Nx
350         Hmatrix[n:n+2,nim:nim+2]= txd
351         if(j<self.Ny-1):
352             Hmatrix[n:n+2,njp:njp+2]= self.ty
353     elif (i==self.Nx-1 and j==self.Ny-1):
354         nip= 2*i + 2*j*self.Nx
355         njp= 2*i + 2*j*self.Nx
356         Hmatrix[n:n+2,nim:nim+2]= txd
357         Hmatrix[n:n+2,njm:njm+2]= tyd
358     elif (i==0 and j>0 and j<self.Ny-1): #boundary points
359         if(i<self.Nx-1):
360             Hmatrix[n:n+2,nip:nip+2]= self.tx
361             Hmatrix[n:n+2,njp:njp+2]= self.ty
362             Hmatrix[n:n+2,njm:njm+2]= tyd
363     elif (i==self.Nx-1 and j>0 and j<self.Ny-1):
364         Hmatrix[n:n+2,nim:nim+2]= txd
365         Hmatrix[n:n+2,njp:njp+2]= self.ty
366         Hmatrix[n:n+2,njm:njm+2]= tyd
367     elif (j==0 and i>0 and i<self.Nx-1):
368         Hmatrix[n:n+2,nip:nip+2]= self.tx
369         Hmatrix[n:n+2,nim:nim+2]= txd
370         if(j<self.Ny-1):
371             Hmatrix[n:n+2,njp:njp+2]= self.ty
372     elif (j==self.Ny-1 and i>0 and i<self.Nx-1):
373         Hmatrix[n:n+2,nip:nip+2]= self.tx
374         Hmatrix[n:n+2,nim:nim+2]= txd
375         Hmatrix[n:n+2,njm:njm+2]= tyd
376
377     self.Hmatrix= Hmatrix
378
379 def MakeEnergyMatrix(self):

```

A. Appendix

```
379     """
380     Combine the whole matrix
381     """
382
383     N = len(self.Hmatrix[0,:])
384     EnergyMatrix = np.zeros(shape=(2*N,2*N),dtype= complex)
385     EnergyMatrix[0:N,0:N] = self.Hmatrix
386     EnergyMatrix[0:N,N:2*N] = self.DeltaMatrix
387     EnergyMatrix[N:2*N,0:N] = -np.conj(self.DeltaMatrix)
388     EnergyMatrix[N:2*N,N:2*N] = -np.conj(self.Hmatrix)
389
390     self.EnergyMatrix= EnergyMatrix
391
392 def PairingPotential(self):
393     """
394     Purpose:
395     The function calculates the pairing potential.
396
397     Definitions:
398     EigenVectors/values: contains all the eigenvectors and (positive)
399                        eigenvalues of the BdG equation.
400     g: attractive on-site e-e coupling strength (g > 0).
401     thermE: thermal energy.
402     epsilonD: cut-off energy ~ Debye energy.
403     """
404
405     delta= np.zeros(self.Nx*self.Ny,dtype=complex)
406     Ne= len(self.EigenValuesP)
407
408     for k in range(Ne):
409         etemp= self.EigenValuesP[k]
410         if (etemp > self.epsilonD):
411             break
412         if (np.real(etemp) < 1e-13):
413             etemp = 1.j*np.imag(etemp)
414         if (np.imag(etemp) < 1e-13):
415             etemp = np.real(etemp)
416         etemp= etemp/self.thermE
417         fermi= 1/(np.exp(np.real(etemp)) + 1)
418         fermi= 2*fermi - 1
419         for i in range(self.Nx):
420             for j in range(self.Ny):
421                 n= 2*i + 2*j*self.Nx
422                 nd= i + j*self.Nx
423                 prod= self.EigenVectorsU[n+1,k]*np.conj(self.EigenVectorsV[n,k
424                 ]) - self.EigenVectorsU[n,k]*np.conj(self.EigenVectorsV[n
425                 +1,k])
426                 delta[nd]= delta[nd] + 0.5*self.g*prod*fermi
427
428     self.deltan= delta
429
430 def MajoranaDensities(self):
```

```

428     """
429     Purpose: Calculate probability density of the majorana fermions of
430             the
431             lowest energy states.
432     Comment:
433     Let f be a standard fermionic operator. It can be split into two MFs,
434             m1 and m2, by
435      $f = (m1 + i*m2)/2 \Rightarrow m1 = f^\dagger + f$  and  $m2 = i(f^\dagger - f)$ .
436     """
437     N_2= int(len(self.EigenValues)/2)
438
439     M1= np.zeros(shape=(self.Nx*self.Ny, self.nrM));
440     M2= np.zeros(shape=(self.Nx*self.Ny, self.nrM));
441
442     NV= 2*self.Nx*self.Ny
443
444     for k in range(self.nrM):
445         nmi= N_2 - k-1
446         npl= N_2 + k
447         vm= self.EigenVectors[:,nmi]#this is f^\dagger (the hole
448             wavefunction).
449         vp= self.EigenVectors[:,npl]#this is f (the particle wavefunction).
450         m1= (vp + vm)/np.sqrt(2)
451         m2= (vp - vm)/np.sqrt(2)
452         Um1= m1[0:NV]
453         Vm1= m1[NV:2*Nv]
454         Um2= m2[0:NV]
455         Vm2= m2[NV:2*Nv]
456
457         for i in range(self.Nx):
458             for j in range(self.Ny):
459                 n= 2*i + 2*j*self.Nx
460                 nd= i + j*self.Nx
461                 u1= Um1[n:n+2]
462                 v1= Vm1[n:n+2]
463                 u2= Um2[n:n+2]
464                 v2= Vm2[n:n+2]
465                 u1= np.linalg.norm(u1)
466                 v1= np.linalg.norm(v1)
467                 u2= np.linalg.norm(u2)
468                 v2= np.linalg.norm(v2)
469                 M1[nd,k]= u1*u1 + v1*v1
470                 M2[nd,k]= u2*u2 + v2*v2
471
472     self.M1=M1
473     self.M2=M2
474
475     def SortEigVecVal(self):
476         """
477         Purpose:

```

A. Appendix

```
477     The function picks out the positive eigenvalues and the corresponding
478         eigenvectors.
479
480     Definitions:
481     eigval: diagonal matrix containing the eigenvalues
482     eigvec: matrix containing the eigenvectors, where eigvec(n,:)
483         corresponds
484         to the eigenvector of the BdG Hamiltonian with eigenvalue eigval(n,n)
485         .
486     The eigenvectors are normalized such that  $\sum_{ij} (u_{ij}^2 + v_{ij}^2) = 1$ .
487
488     However, note that the correct normalization condition is the
489     unscaled
490     system should be  $\sum_{ij} a^2 (u_{ij}^2 + v_{ij}^2) = 1$ .
491     """
492
493     Ne= len(self.EigenValues)
494     Ne= np.int64(Ne/2)
495     N= len(self.EigenVectors[:,1])
496     N_2= np.int64(N/2)
497
498     #initiate matrices:
499     self.EigenValuesP= np.zeros(Ne, dtype=complex)
500     self.EigenVectorsU= np.zeros(shape=(N_2, Ne), dtype=complex)
501     self.EigenVectorsV= np.zeros(shape=(N_2, Ne), dtype=complex)
502
503     #Write vectors and values
504     for i in range(Ne):
505         itemp= Ne + i; #1,...,Ne contains the negative eigenvalues.
506         self.EigenValuesP[i] = np.real(self.EigenValues[itemp])
507         self.EigenVectorsU[:,i]= self.EigenVectors[0:N_2,itemp]
508         self.EigenVectorsV[:,i]= self.EigenVectors[N_2:N,itemp]
509
510     def MakeDensityMatrix(self):
511         """
512         Purpose: Calculates density of each eigenfunction.
513         """
514         Ne= len(self.EigenVectorsU[1,:])
515         density= np.zeros(shape=(self.Nx*self.Ny, Ne))
516
517         for k in range(Ne):
518             for i in range(self.Nx):
519                 for j in range(self.Ny):
520                     n= 2*i + 2*j*self.Nx
521                     nd= i + j*self.Nx
522                     u= self.EigenVectorsU[n:n+2,k]
523                     v= self.EigenVectorsV[n:n+2,k]
524                     u= np.linalg.norm(u)
525                     v= np.linalg.norm(v)
526                     density[nd,k]= u*u + v*v
527
528         self.DensityMatrix=density
```

```

524
525 def CalculateCurrent(self):
526     """
527     Purpose: Calculates the equilibrium current density in units of 2*e*t
           /hbar= C/s (t: spin-independent hopping amplitude).
528     """
529
530     if (self.SOC==0):
531         vsox= -self.s2
532         vsoy= self.s1
533     elif (self.SOC==1):# x=[100]. NB: x=[010] is found by changing the
           alpha sign
534         vsox= -self.s1
535         vsoy= self.s2
536     elif (self.SOC==2):# x=[110]. NB: x=[-110] is found by changing the
           alpha sign
537         vsox= self.s2
538         vsoy= self.s1
539
540     jSOScale= self.alphaR;
541     j0Scale= 1;
542
543     Ne= len(self.EigenValuesP);
544     j0x= np.zeros(shape=(self.Nx, self.Ny));
545     j0y= np.zeros(shape=(self.Nx, self.Ny));
546     jSOx= np.zeros(shape=(self.Nx, self.Ny));
547     jSOy= np.zeros(shape=(self.Nx, self.Ny));
548     epsilonD= self.epsilonD/self.thermE;
549
550     for k in range(Ne):
551         ek= self.EigenValuesP[k]/self.thermE
552         if(ek > epsilonD):
553             break
554         if (np.real(ek) < 1e-13):
555             ek = 1.j*np.imag(ek)
556         if (np.imag(ek) < 1e-13):
557             ek = np.real(ek)
558         fe= 1/(np.exp(ek) + 1)
559         fh= 1- fe
560         for i in range(1, self.Nx-1):
561             for j in range(1, self.Ny-1):
562                 ip= i+1
563                 im= i-1
564                 jp= j+1
565                 jm= j-1
566                 n= 2*i + 2*j*self.Nx
567                 nxp= 2*ip + 2*j*self.Nx
568                 nxm= 2*im + 2*j*self.Nx
569                 nyp= 2*i + 2*jp*self.Nx
570                 nym= 2*i + 2*jm*self.Nx
571                 u= self.EigenVectorsU[n:n+2,k]#electron-like spinor at n.
572                 v= self.EigenVectorsV[n:n+2,k]#hole-like spinor at n.

```

A. Appendix

```
573         ux= np.dot (vsox, u)
574         vx= np.dot (vsox, np.conj (v) )
575         uy= np.dot (vsoy, u)
576         vy= np.dot (vsoy, np.conj (v) )
577         u= np.conj (u)
578         dudx= 0.5*(self.EigenVectorsU[nxp:nxp+2,k] - self.EigenVectorsU
579             [nxm:nxm+2,k])
580         dvdx= np.conj(0.5*(self.EigenVectorsV[nxp:nxp+2,k] - self.
581             EigenVectorsV[nxm:nxm+2,k]))
582         dudy= 0.5*(self.EigenVectorsU[nyp:nyp+2,k] - self.EigenVectorsU
583             [nym:nym+2,k])
584         dvdy= np.conj(0.5*(self.EigenVectorsV[nyp:nyp+2,k] - self.
585             EigenVectorsV[nym:nym+2,k]))
586         current0x= np.imag(np.dot (u, dudx)*fe + np.dot (v, dvdx)*fh)
587         current0y= np.imag(np.dot (u, dudy)*fe + np.dot (v, dvdy)*fh)
588         currentSOx= np.real(np.dot (u, ux)*fe + np.dot (v, vx)*fh)
589         currentSOy= np.real(np.dot (u, uy)*fe + np.dot (v, vy)*fh)
590         j0x[i, j]= j0x[i, j] + current0x
591         j0y[i, j]= j0y[i, j] + current0y
592         jSOx[i, j]= jSOx[i, j] + currentSOx
593         jSOy[i, j]= jSOy[i, j] + currentSOy
594
595     self.jx= j0Scale*j0x + jSOScale*jSOx
596     self.jy= j0Scale*j0y + jSOScale*jSOy
597
598     def CalculateEigenValues (self) :
599         """
600         Purpose: solve Bdg eigenvalue problem
601         """
602         self.MakeDeltaMatrix()
603         self.MakeEnergyMatrix()
604         [self.EigenValues, self.EigenVectors]= np.linalg.eigh(self.
605             EnergyMatrix)
606         self.SortEigVecVal()
607
608     def Save (self, filename) :
609         """
610         Purpose: saves the whole object into a file
611         """
612         pickle.dump (self, open (filename, "wb"), pickle.HIGHEST_PROTOCOL)
```

Next we show [RelaxCPU.py](#).

```
1 import numpy as np
2
3 def RelaxPlot (Matrix, p) :
4     """
5     Purpose: solve Bdg eigenvalue problem self consistently and show the
6         pairing potential of every step.
```

```

6  """
7
8  Matrix.deltainitial=Matrix.delta
9
10 for i in range(Matrix.Nmax):
11     Matrix.MakeDeltaMatrix()
12     Matrix.MakeEnergyMatrix()
13     [Matrix.EigenValues,Matrix.EigenVectors]= np.linalg.eigh(Matrix.
        EnergyMatrix)
14     Matrix.SortEigVecVal()
15     Matrix.PairingPotential()
16     dv= abs(Matrix.deltan) - abs(Matrix.delta);
17     error= np.linalg.norm(dv)/np.linalg.norm(abs(Matrix.deltan));#
        checking convergence of the cooper pair density
18     print('iteration number: %d, error: %f, total energy: %f\n'%(i,error,
        sum(np.real(Matrix.EigenValuesP))))
19     if(error < Matrix.Maxerror) or (np.linalg.norm(dv) == 0):
20         break
21     Matrix.delta= Matrix.deltan
22     p.VectorPlotDeltaLive(Matrix,i)
23
24 def Relax(Matrix):
25     """
26     Purpose: solve Bdg eigenvalue problem Matrix consistently
27     """
28
29     Matrix.deltainitial=Matrix.delta
30
31     for i in range(Matrix.Nmax):
32         Matrix.MakeDeltaMatrix()
33         Matrix.MakeEnergyMatrix()
34         [Matrix.EigenValues,Matrix.EigenVectors]= np.linalg.eigh(Matrix.
            EnergyMatrix)
35         Matrix.SortEigVecVal()
36         Matrix.PairingPotential()
37         dv= abs(Matrix.deltan) - abs(Matrix.delta)
38         error= np.linalg.norm(dv)/np.linalg.norm(abs(Matrix.deltan)) #
            checking convergence of the cooper pair density
39         print('iteration number: %d, error: %f, total energy: %f\n'%(i,error,
            sum(np.real(Matrix.EigenValuesP))))
40         if(error < Matrix.Maxerror) or (np.linalg.norm(dv) == 0):
41             break
42         Matrix.delta= Matrix.deltan

```

Next we show [RelaxGPU.py](#).

```

1 import numpy as np
2 import pycuda.gpuarray as gpuarray
3 import pycuda.autoinit
4 from skcuda import linalg
5

```

A. Appendix

```
6 def RelaxGPU_NP(Matrix):
7     """
8     Purpose: solve Bdg eigenvalue problem self consistently, without
           printing lines.
9     """
10    linalg.init()
11
12    Matrix.deltainitial=Matrix.delta
13
14    for i in range(Matrix.Nmax):
15        Matrix.MakeDeltaMatrix()
16        Matrix.MakeEnergyMatrix()
17        a_gpu = gpuarray.to_gpu(Matrix.EnergyMatrix)
18        vr_gpu, w_gpu = linalg.eig(a_gpu, 'N', 'V')
19        Matrix.EigenValues = w_gpu.get()
20        Matrix.EigenVectors = np.transpose(np.conj(vr_gpu.get()))
21        Matrix.SortEigVecVal()
22        Matrix.PairingPotential()
23        dv= abs(Matrix.deltan) - abs(Matrix.delta);
24        error= np.linalg.norm(dv)/np.linalg.norm(abs(Matrix.deltan));#
           checking convergence of the cooper pair density
25        if(error < Matrix.Maxerror) or (np.linalg.norm(dv) == 0):
26            break
27        Matrix.delta= Matrix.deltan
28
29 def RelaxGPU(Matrix):
30     """
31     Purpose: solve Bdg eigenvalue problem self consistently
32     """
33    linalg.init()
34
35    Matrix.deltainitial=Matrix.delta
36
37    for i in range(Matrix.Nmax):
38        Matrix.MakeDeltaMatrix()
39        Matrix.MakeEnergyMatrix()
40        a_gpu = gpuarray.to_gpu(Matrix.EnergyMatrix)
41        vr_gpu, w_gpu = linalg.eig(a_gpu, 'N', 'V')
42        Matrix.EigenValues = w_gpu.get()
43        Matrix.EigenVectors = np.transpose(np.conj(vr_gpu.get()))
44        Matrix.SortEigVecVal()
45        Matrix.PairingPotential()
46        dv= abs(Matrix.deltan) - abs(Matrix.delta);
47        error= np.linalg.norm(dv)/np.linalg.norm(abs(Matrix.deltan));#
           checking convergence of the cooper pair density
48        print('iteration number: %d, error: %f, total energy: %f\n'%(i,error,
           sum(np.real(Matrix.EigenValuesP))))
49        if(error < Matrix.Maxerror) or (np.linalg.norm(dv) == 0):
50            break
51        Matrix.delta= Matrix.deltan
```


Now we also show one of the [SampleScript.py](#). In particular it is the one for skyrmion-vortex pairs.

```

1 import numpy as np
2
3 from SCF_BDG import MakeMatrices, Plot, RelaxCPU
4
5 Matrix=MakeMatrices.MakeMatrix()
6 p = Plot.Plot()
7 p.ClearFolder()
8
9 # =====
10 #Initialize Matrix
11
12 #Set max iterations and error (in pairing potential):
13 Matrix.Nmax=30
14 Matrix.Maxerror= 0.00001
15
16 #System size parameters:
17 Matrix.Nx=23
18 Matrix.Ny=23
19
20 #System parameters (dimensionless):
21 Matrix.alphaR= 0.5 # Rashba SOC
22 Matrix.mu= -4 # Chemical potential
23 Matrix.h0= 0.4 # strength of exchange field
24 Matrix.thermE= 0.001 #thermal energy, kB*T
25 Matrix.epsilonD= 2 #Debye frequency
26 Matrix.bx=0#bx=0 (open BCs along x); bx=1 (periodic BCs along x)
27 Matrix.by=0
28
29 #Make magnetic skyrmion texture:
30 Matrix.Sq=1
31 Matrix.Sw=1
32 NxMid= int((Matrix.Nx-1)/2)
33 NyMid= int((Matrix.Ny-1)/2)
34 Matrix.Srx= [NxMid]# center of skyrmion
35 #Matrix.Srx= [NxMid-3,NxMid+3] #for 2 Skyrmions
36 Matrix.Sry= [NyMid]
37 #Matrix.Sry= [NyMid-3,NyMid+3] #for 2 Skyrmions
38 Matrix.randPointsX= 4#distance from boundary of sample to ferromagnetic
   block
39 Matrix.randPointsY= 4
40 Matrix.thetaSkyrm= 0*np.pi/2 # thetaSkyrm= 0 (Neel skyrmion); thetaSkyrm=
   pi/2 (Bloch skyrmion)
41 Matrix.MakeSkyrmTextures() # for homogeneous magnetization Matrix.
   MakeFerroBlock()
42
43 #Initialize pairing potential:
44 Matrix.g= 5.0 #attractive coupling strength

```

A. Appendix

```
45 Matrix.delta= np.ones(Matrix.Nx*Matrix.Ny, dtype=complex) #Homogenous
    order-parameter field
46 Matrix.Vq=1.0 #vorticity
47 Matrix.thetaVortex= 0*np.pi/2 # thetaVortex= 0 ("Neel" vortex);
    thetaVortex= pi/2 ("Bloch" vortex)
48 Matrix.Vrx= [NxMid] #Position of vortex
49 #Matrix.Vrx= [NxMid-3,NxMid+3] #for 2 vortices
50 Matrix.Vry= [NyMid] #Position of vortex
51 #Matrix.Vry= [NyMid-3,NyMid+3] #for 2 vortices
52 Matrix.MakeVortices() #Insert a vortex in the pair potential
53
54 #Make constant matrices:
55 Matrix.MakeConstantMatrices()
56 Matrix.MakeHMatrix()
57
58 # =====
59 # Relax and save result
60
61 RelaxCPU.RelaxPlot(Matrix,p)
62
63 #Print the lowest energy states
64 N_2=int(len(Matrix.EigenValues)/2)
65 print('\n')
66 print('lowest energy values:\n')
67 print(np.real(Matrix.EigenValues[N_2-5:N_2+5]))
68 print('\n')
69
70 #Save data:
71 Matrix.Save("Matrix")
72 p.ExportMagnetisationProfileVTK(Matrix)
73 p.DensityPlotMajoranasCombined(Matrix,0)
74 p.DensityVectorPlotDelta(Matrix,0)
```

Last but not least, we also show a sample file how we did the phase diagrams in [LowestEnergyMode.py](#).

```
1 import numpy as np
2 import sys
3 import imp
4 from joblib import Parallel, delayed
5 import multiprocessing
6 from tqdm import tqdm
7 from SCF_BDG import MakeMatrices, RelaxGPU
8
9 def processInput(h0,mu):
10
11     Matrix=MakeMatrices.MakeMatrix()
12
13     # =====
14     #Initialize Matrix
15
```

```

16 #Set max iterations and error (in pairing potential):
17 Matrix.Nmax=30
18 Matrix.Maxerror= 0.00001
19
20 #System size parameters:
21 Matrix.Nx=60 #should be an odd (prime) number
22 Matrix.Ny=11 #should be an odd (prime) number
23
24 #System parameters (dimensionless):
25 Matrix.alphaR= 0.3 # Rashba SOC
26 Matrix.mu= mu # Chemical potential
27 Matrix.h0 = h0 # strength of exchange field
28 Matrix.thermE= 0.001 #thermal energy, kB*T
29 Matrix.epsilonD= 2 #Debye frequency
30
31 #Make magnetic skyrmion texture:
32 Matrix.randPointsX = 2 #distance from boundary of sample to
    ferromagnetic block
33 Matrix.randPointsY = 5
34 Matrix.MakeFerroBlock()
35
36 #Initialize pairing potential:
37 Matrix.g= 5.0 #attractive coupling strength
38 Matrix.delta= 1*np.ones(Matrix.Nx*Matrix.Ny, dtype=complex) #Homogenous
    order-parameter field
39
40 #Make constant matrices:
41 Matrix.MakeConstantMatrices()
42 Matrix.MakeHMatrix()
43
44 RelaxGPU.RelaxGPU_NP(Matrix)
45 DeltaMatrix = Matrix.MapFromVectorToMatrix(Matrix.delta)
46 return [Matrix.EigenValues[int(len(Matrix.EigenValues)/2)], np.mean(abs(
    DeltaMatrix[Matrix.randPointsX:Matrix.Nx-Matrix.randPointsX, Matrix.
    randPointsY:Matrix.Ny-Matrix.randPointsY]))]
47
48 # =====
49 # Relax and save result
50
51 h0 = np.linspace(0, 3, 30)
52 mu = np.linspace(-6, 0, 30)
53
54 result = Parallel(n_jobs=4)(delayed(processInput)(h0p, mup) for h0p in
    tqdm(h0) for mup in tqdm(mu))
55
56 np.savez("LowestEnergyMode_test", mu, h0, result)

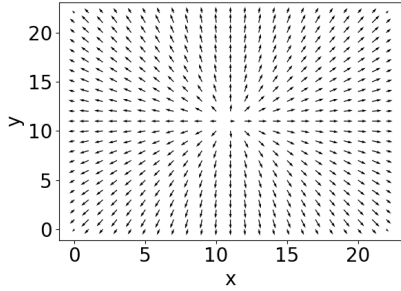
```

We should mention that we calculated the phase diagram in steps of 0.1 for h_0^* and in 0.2 steps for μ^* . The resources needed to do this was 4 cores of CPU and 3-4 GB of graphic card memory. With that we needed 6-12 hours to calculate the diagram depending on the system we considered.

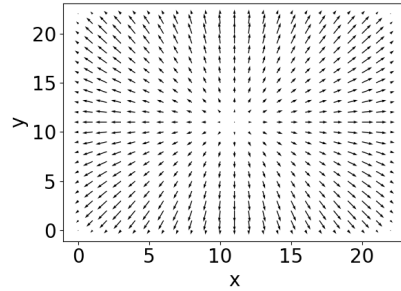
A.4. Relaxation

Here we show the relaxation progress of a skyrmion-vortex pair shown in chapter 4.6. We see in figure A.1 that from $\xi = 10^{-4}$ to $\xi = 10^{-5}$ there is only a small change in the total energy E_{tot}^* as well as in the superconducting gap Δ^* . Because of this we chose $\xi = 10^{-4}$ as a termination condition for the relaxation.

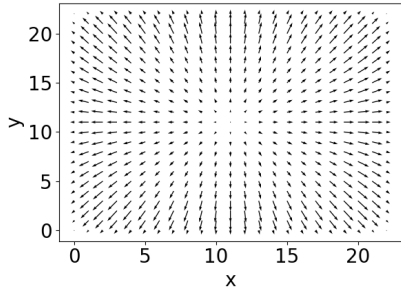
(a) initial, $E_{\text{tot}}^* = 4449.804150$



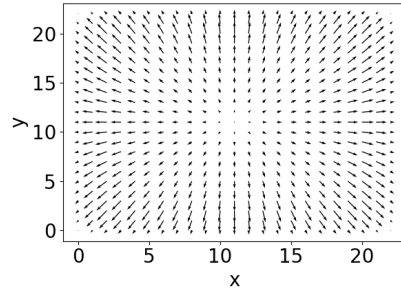
(b) $\xi = 10^{-1}$, $E_{\text{tot}}^* = 4252.748002$



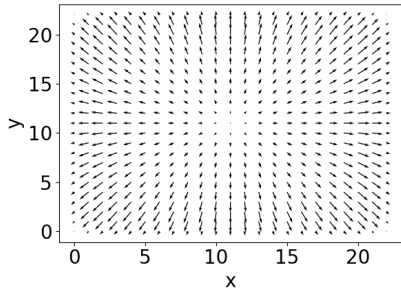
(b) $\xi = 10^{-2}$, $E_{\text{tot}}^* = 4250.125666$



(b) $\xi = 10^{-3}$, $E_{\text{tot}}^* = 4249.886317$



(b) $\xi = 10^{-4}$, $E_{\text{tot}}^* = 44249.857121$



(b) $\xi = 10^{-5}$, $E_{\text{tot}}^* = 4249.854824$

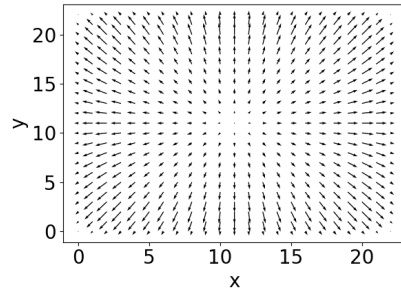


Figure A.1.: Relaxation progress of a skyrmion-vortex pair.

A.5. Procedure for the phase diagrams

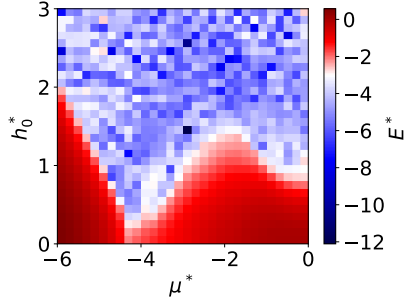
In this chapter we explain how the phase diagram in chapter 4.3 was done. We will also explain why only in one phase diagram we could add the condition for being topological into the plot. Then we also address the problem that we can not include conditions in every phase diagram.

In order to see where the system is topological we plotted the lowest energy mode of the system since it is assumed that this goes to zero inside the topological region. The plot showing this can be seen in figure A.2 (a) and (b). We chose to take small steps in the parameters in the order of 0.1. With that we are able to see a pretty clear separation between topological and non topological. The system does not converge in the same manner for different parameter sets which results in small fluctuations inside the topological region. These fluctuations also stay after we interpolated the data to get a smoother picture.

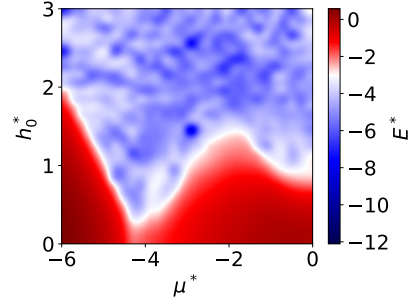
Next we added the condition in equation 4.3 inside the plot. For that we first analyse the superconducting gap $|\Delta^*|$ inside the ferromagnetic region. We chose the same steps as before and plot it with and without interpolation in figure A.2 (c) and (d). There we can see that the superconducting gap is higher the lower the magnetic field h_0^* and the higher the chemical potential μ^* is. This makes sense since higher magnetic fields destroy the superconductivity and lower chemical potential means that less bands are occupied resulting in a lower superconductivity. Next we use the condition to produce the plot in figure A.2 (e) and (f). There the green region means that we are topological and the purple region means that we are not topological. The shape shown in this plot was then interpolated and the border was drawn inside the plot with the lowest energy mode. Another possibility to analyse where the system is topological is to consider specific regions in the ferromagnet instead of the whole ferromagnet.

Finally, we discuss the problems with the condition for the other configurations. For the Kitaev Chain on a two dimensional superconductor in chapter 4.2 the condition in equation 3.11 does not apply here since we consider two different shapes for the ferromagnet and the superconductor. In [32] they derived a condition using the Pfaffian of the matrix but this condition also does not to apply for this system. It is only for a fully one dimensional system as Cadez et al. mentioned. The only possibility to get the right condition would be to calculate the Pfaffian of the considered matrix directly but we were not able to do this in the limited time. For the vortex, skyrmion and skyrmion-vortex pair system in chapter 4.4, 4.5 and 4.6 we occurred a different problem. Since these systems are not translationally invariant, we cannot change to momentum space. Since the condition in 3.11 was derived by going to momentum space it is not applicable. Also, the method of calculating the pfaffian Cadez et al. used in [32] considered going to momentum space. However, it could be that there is a way to get the condition from the pfaffian of a real space matrix but we could not find one during this thesis. As a result this means that we need to derive a general condition for being topological in real space for these systems.

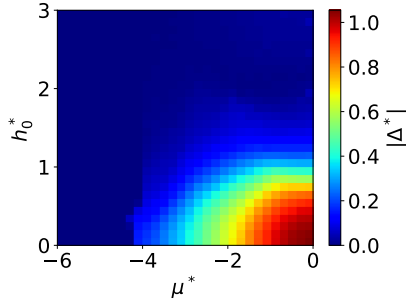
(a) without interpolation



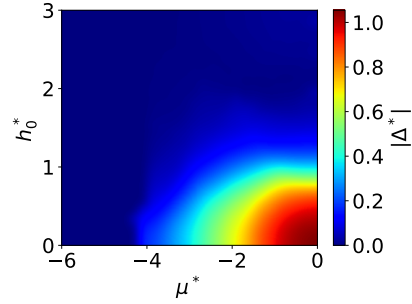
(b) with interpolation



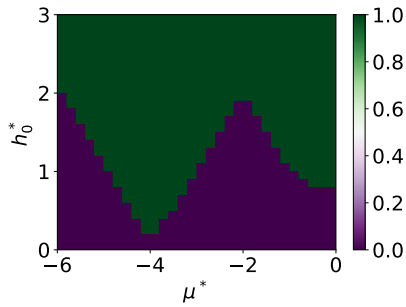
(c) without interpolation



(d) with interpolation



(e) without interpolation



(f) with interpolation

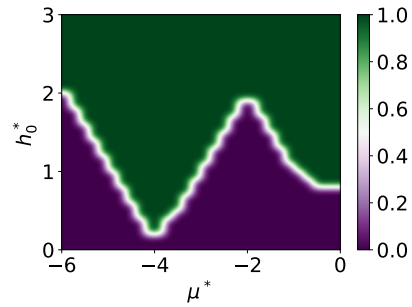


Figure A.2.: The first two pictures show the lowest energy mode ((a) without interpolation and (b) with interpolation). In (d) and (e) we show the mean value of the superconducting gap Δ^* inside the ferromagnetic region without and with interpolation respectively. Then we show when the system is topological (green) and not topological (purple) in (b) and (c).

Bibliography

- [1] G. E. Moore, IEEE Solid-State Circuits Newsl. **20**, 33 (2006).
- [2] Richard P Feynman, Int. J. Theor. Phys. **21**, 467 (1982), arXiv:quant-ph/9508027 .
- [3] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997), arXiv:quant-ph/9508027 .
- [4] Quantum flagship, <https://qt.eu/>, accessed: 2019-03-13.
- [5] Ibm q, <https://www.research.ibm.com/ibm-q/>, accessed: 2019-02-07.
- [6] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, R. Barends, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Fowler, B. Foxen, E. Jeffrey, J. Kelly, E. Lucero, A. Megrant, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, H. Neven, and J. M. Martinis, Science **199**, 195 (2018), arXiv:1709.06678v1 .
- [7] J. I. Cirac and P. Zoller, Phys. Rev. Lett. **74**, 4091 (1995), arXiv:quant-ph/0305129 .
- [8] L. Childress and R. Hanson, MRS Bull. **38**, 134 (2013), arXiv:1112.1950 .
- [9] D. Loss and D. P. DiVincenzo, Phys. Rev. A - At. Mol. Opt. Phys. **57**, 120 (1998), arXiv:9701055 [cond-mat] .
- [10] J. K. Pachos, *Introduction to Topological Quantum Computation* (Cambridge University Press, Cambridge, 2012).
- [11] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma, Rev. Mod. Phys. **80**, 1083 (2008), arXiv:0707.1889 .
- [12] E. Majorana and L. Maiani, Nuovo Cim. (engl. Transl. Soryushiron Kenkyu) **63**, 149 (1981).
- [13] A. Y. Kitaev, Physics-Uspekhi **44**, 131 (2001), arXiv:cond-mat/0010440 .
- [14] L. Fu and C. L. Kane, Phys. Rev. Lett. **100**, 096407 (2008), arXiv:0707.1692 .
- [15] J. D. Sau, R. M. Lutchyn, S. Tewari, and S. Das Sarma, Phys. Rev. Lett. **104**, 1 (2010), arXiv:0907.2239 .
- [16] J. Alicea, Y. Oreg, G. Refael, F. von Oppen, and M. P. A. Fisher, Nat. Phys. **7**, 412 (2011), arXiv:1006.4395 .
- [17] T.-P. Choy, J. M. Edge, A. R. Akhmerov, and C. W. J. Beenakker, Phys. Rev. B **84**, 195442 (2011), arXiv:1108.0419 .

- [18] V. Mourik, K. Zuo, S. M. Frolov, S. R. Plissard, E. P. A. M. Bakkers, and L. P. Kouwenhoven, *Science* **336**, 1003 (2012).
- [19] H. Kim, A. Palacio-Morales, T. Posske, L. Rózsa, K. Palotás, L. Szunyogh, M. Thorwart, and R. Wiesendanger, *Sci. Adv.* **4**, eaar5251 (2018).
- [20] G. Yang, P. Stano, J. Klinovaja, and D. Loss, *Phys. Rev. B* **93**, 224505 (2016), arXiv:1602.00968 .
- [21] F. Jonietz, S. Muhlbauer, C. Pfleiderer, A. Neubauer, W. Munzer, A. Bauer, T. Adams, R. Georgii, P. Boni, R. A. Duine, K. Everschor, M. Garst, and A. Rosch, *Science* **330**, 1648 (2010).
- [22] S. Woo, K. Litzius, B. Krüger, M.-Y. Im, L. Caretta, K. Richter, M. Mann, A. Krone, R. M. Reeve, M. Weigand, P. Agrawal, I. Lemesh, M.-A. Mawass, P. Fischer, M. Kläui, and G. S. D. Beach, *Nature Materials* **15**, 501 EP (2016).
- [23] Topology in condensed matter, <https://topocondmat.org/>, accessed: 2019-01-24.
- [24] M. Leijnse and K. Flensberg, *Semicond. Sci. Technol.* **27**, 124003 (2012), arXiv:1206.1736 .
- [25] V. Lahtinen and J. K. Pachos, *SciPost Phys.* **3**, 1 (2017), arXiv:1705.04103 .
- [26] A. Y. Kitaev, *Ann. Phys. (N. Y.)*. **303**, 2 (2003), arXiv:quant-ph/9707021 .
- [27] D. A. Ivanov, *Phys. Rev. Lett.* **86**, 268 (2001), arXiv:0005069 [cond-mat] .
- [28] P. DE GENNES, *Superconductivity of Metals and Alloys*, Advanced Book Classics (Addison-Wesley Publishing Company, 1966).
- [29] K. M. Hals, M. Schechter, and M. S. Rudner, *Phys. Rev. Lett.* **117**, 017001 (2016), arXiv:1603.07550 .
- [30] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, *Phys. Rev.* **108**, 1175 (1957).
- [31] M. Sato, Y. Takahashi, and S. Fujimoto, *Phys. Rev. B* **82**, 134521 (2010).
- [32] T. Čadež and P. D. Sacramento, *J. Phys. Condens. Matter* **28**, 1 (2016), arXiv:1506.07909v2 .
- [33] J. Li, B. A. Bernevig, A. H. MacDonald, Z. Wang, T. Neupert, and A. Yazdani, *Nat. Commun.* **7**, 1 (2016).
- [34] T. Liu, J. J. He, and F. Nori, *Physical Review B* **98**, 10.1103/physrevb.98.245413 (2018).
- [35] K. Björnson and A. M. Black-Schaffer, *Phys. Rev. B* **88**, 024501 (2013), arXiv:1304.0981 .

-
- [36] K. Björnson and A. M. Black-Schaffer, Phys. Rev. B - Condens. Matter Mater. Phys. **91**, 1 (2015).
- [37] E. W. J. Straver, J. E. Hoffman, O. M. Auslaender, D. Rugar, and K. A. Moler, Appl. Phys. Lett. **93**, 172514 (2008).
- [38] K. Everschor-Sitte, J. Masell, R. M. Reeve, and M. Kläui, Journal of Applied Physics **124**, 240901 (2018).
- [39] A. Abanov and V. Pokrovsky, Phys. Rev. B - Condens. Matter Mater. Phys. **58**, R8889 (1998), arXiv:cond-mat/9801114 .
- [40] K. Hashimoto, *Non-Universal Superconducting Gap Structure in Iron-Pnictides Revealed by Magnetic Penetration Depth Measurements* (Springer Japan, 2013).
- [41] SCF_BDG, https://github.com/jnothelf/SCF_BDG, accessed: 2019-02-28.
- [42] L. E. Givon, T. Unterthiner, N. B. Erichson, D. W. Chiang, E. Larson, L. Pfister, S. Dieleman, G. R. Lee, S. van der Walt, B. Menn, T. M. Moldovan, F. Bastien, X. Shi, J. Schlüter, B. Thomas, C. Capdevila, A. Rubinsteyn, M. M. Forbes, J. Frelinger, T. Klein, B. Merry, N. Merrill, L. Pastewka, L. Y. Liu, S. Clarkson, M. Rader, S. Taylor, A. Bergeron, N. H. Ukani, F. Wang, W.-K. Lee, and Y. Zhou, scikit-cuda 0.5.2: a Python interface to GPU-powered libraries (2018).
- [43] paraview, <https://www.kitware.com/platforms/paraview>, accessed: 2019-02-28.

Acknowledgements

I would first like to thank my thesis advisors Dr. Karin Everschor-Sitte and Prof. Dr. Matteo Rizzi for giving me the opportunity to write about a self-chosen topic. The door to their office was always open whenever I ran into a trouble spot or had a question about my research or writing. They consistently allowed this thesis to be my own work, but steered me in the right direction whenever they thought I needed it. Nevertheless they gave me enough space to enforce my own will.

I would also like to thank Prof. Dr. Kjetil Hals I visited during my thesis. I really enjoyed the discussions about the topic and had a wonderful time in Norway.

Moreover I would also like to acknowledge Prof. Dr. Mathias Kläui as the second reader of this thesis, and I am gratefully indebted for his very valuable comments on this thesis. Furthermore I want to thank him for the possibility to be a part of measuring travels which gave me great insights in experimental magnetisation imaging techniques at synchrotrons.

Next I thank my fellow students Nico Kerber, Pascal Praß, Franziska Martin, Christian Matthé, Martin Müller and Alexey Kivel for accompanying me on this journey. I had great pleasure during the intermissions at the cafeteria. The discussions there always cheered me up and motivated me to continue working. Especially I want to thank Nico Kerber and Kai Litzius with whom I had a very nice and funny time during the working day.

I would like to extend my thanks to Kai Litzius, Nico Kerber, Ben McKeever and Franziska Martin for proof-reading my thesis. Furthermore, I thank Tabea Kühner providing me with helpful suggestions to improve my language both grammatically and stylistically. Additionally, I thank Venkata Krishna Bharadwaj for helping me with every computer problem I faced during the thesis.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here.

This paper was not previously presented to another examination board and has not been published.

Mainz the March 18, 2019

Jonas Nothhelfer