**Master Thesis** 

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

Jonas Nothhelfer

submitted: March 18, 2019 last correction: June 5, 2019



First referee:Dr. Karin Everschor-Sitte and Prof. Dr. Matteo RizziSecond 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.

 $\begin{array}{l} \textbf{Abelian anyon} \\ \ket{\psi_1,\psi_2} = e^{i\theta} \ket{\psi_2,\psi_1}, \ \theta \neq 0, \pi \end{array}$ 

**non-Abelian anyon**  $|\psi_1,\psi_2\rangle = \boldsymbol{M} |\psi_2,\psi_1\rangle, \ \boldsymbol{M} \text{ is a matrix}$ 

### Abstract

In this thesis we investigate the existence of Majorana zero modes in ferromagnetsuperconductor 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 nonabelian 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 Bogoliubovde 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 skyrmionvortex 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
Α.	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

# 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  $p_x \pm i p_y$ -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 skyrmionvortex 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 loosing 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 pertubations. 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 pertubations.

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 noncontractible 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  $\lambda_1$  that does not encircle another particle can be continuously deformed to a path  $\lambda_2$  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  $\lambda_2$  describing two particle exchanges is continuously deformable to  $\lambda_1$  by taking it behind or front of the right-most particle, and in turn  $\lambda_1$  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  $\lambda_2$  and  $\lambda_1$  are topologically inequivalent since  $\lambda_2$  can not be deformed through the right-most particle, while  $\lambda_1$  is still contractible to a point. Hence, the paths now have different topology and different statistical quantum evolutions can be assigned to each [10].

This results in the condition

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

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  $\lambda_2$  can not be continuously deformed (the path is not allowed to pass through the encircled particle) to the contractable path  $\lambda_1$ . 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:  $|\Psi(\lambda_2)\rangle \neq |\Psi(\lambda_1)\rangle = |\Psi(0)\rangle$ .

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  $p_x \pm i p_y$  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]

$$H_{\text{chain}} = -\mu \sum_{i=1}^{N} n_i - \sum_{i=1}^{N-1} (t c_i^{\dagger} c_{i+1} - \Delta c_i c_{i+1} + h.c.)$$
(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  $\mu$ . 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  $\Delta$ . A more detailed description of the different pairings can be found in Appendix A.1. The pairing  $\Delta$  and the hopping t are assumed to be the same for all sites. In the following, we will assume that the superconducting phase  $\phi$  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

$$c_i = rac{1}{2}(\gamma_{i,1} + i\gamma_{i,2})$$
  
 $c_i^{\dagger} = rac{1}{2}(\gamma_{i,1} - i\gamma_{i,2}),$ 

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

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

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

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

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

$$H_{\text{chain}} = -it \sum_{i=1}^{N-1} \gamma_{i,2} \gamma_{i+1,1}$$
(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  $\gamma$  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  $H = \frac{1}{2}C^{\dagger}H_{BdG}C$  with C a column vector containing all creation and annihilation operators,  $C = (c_1, ..., c_N, c_1^{\dagger}, ..., c_N^{\dagger})^T$ , where the first part of the vector corresponds to holes, and the second part corresponds to electrons. The  $2N \times 2N$  matrix  $H_{BdG}$ can be written in a compact way using Pauli matrices  $\tau_i$  in particle-hole space, and denoting with  $|n\rangle$  a column basis vector  $(0, ..., 1, ..., 0)^T$  corresponding to the *n*-th site of the chain. With these definitions the Bogoliubov de Gennes Hamiltonian is then given by

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

Note that this Hamiltonian has particle-hole symmetry since  $\mathcal{P}H_{BdG}\mathcal{P}^{-1} = -H_{BdG}$ 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,  $\mu$  has to be smaller than 2t. To be in the second phase, however, requires  $\mu$  to be bigger than 2t.



Figure 2.3.: Energy spectrum of (2.5) plotted in terms of  $\mu$ . Both energies are rescaled in terms of t. Two zero energy modes appear for  $\mu$  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.



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,  $\Delta$  and  $\mu$  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

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

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, ..., N - 1. Values of k, which differ by  $2\pi$ , are equivalent. Using this, the Bogoliubov-de Gennes Hamiltonian in momentum space reads

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

in the basis  $\boldsymbol{\psi} = (\psi, \psi^{\dagger}) = \langle k | C$ . 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:

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

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 d(k). These components can be read off from equation (2.6) and assume the form

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

From this we see in figure 2.5 that as k goes from zero to  $2\pi$ , the path that the endpoint of the vector d(k) traces out is an ellipse of radius 2 in z-direction and radius  $2\Delta$  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  $\nu$ . 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 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,  $\nu =$  undefined,  $\mu = -2t$  (c) topological phase,  $\nu = 1$ ,  $\mu = 0$  (d) in between the two phases,  $\nu =$  undefined,  $\mu = 2t$ (e) trivial phase,  $\nu = 0$ ,  $\mu = 2.5t$ . Here  $\nu$  is an integer which describes how many times the loop winds around the origin and  $\mu$  corresponds to the chemical potential. Hence, with  $\mu$  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  $\mu$  with respect to the bottom of the band:

$$\mu \to \mu - 2t.$$

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



(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  $\Delta$  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  $\mu$  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:

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

17

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  $\sigma_i$ . 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  $\mu$ . 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  $\mathcal{THT}^{\dagger} = \mathcal{H}$  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

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

where we now extended our basis to  $\boldsymbol{\psi} = (\psi_{\uparrow}, \psi_{\downarrow}, \psi_{\uparrow}^{\dagger}, \psi_{\downarrow}^{\dagger})$ ,  $\tau$  are the Pauli matrices in particle-hole space and  $\sigma$  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  $\mu$ , 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

$$\mathcal{H} = egin{pmatrix} \mathcal{H}_0 & oldsymbol{\Delta} \ oldsymbol{\Delta^*} & -\mathcal{H}_0^* \end{pmatrix} \,,$$

where  $\mathcal{H}_0$  describes the part of our Hamiltonian where only  $\mu$  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

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

where  $\mathcal{T} = -i\sigma_y \mathcal{K}$ ,  $\mathbf{\Delta}' = 2\Delta i\sigma_y k$  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

(a) no Majorana modes B/t = 0.05



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

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

(b) no Majorana modes B/t = 0.2

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

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

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

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

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

whereas equation (2.7) can be written as

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

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

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

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  $\boldsymbol{c} = (c_{\uparrow}, c_{\downarrow}, c_{\downarrow}^{\dagger}, -c_{\uparrow}^{\dagger})$ , leading to

$$H = \sum_{i} \boldsymbol{c}_{i}^{\dagger}(t\tau_{z})\boldsymbol{c}_{i+1} + \boldsymbol{c}_{i}^{\dagger}(-\mu\tau_{z} + \boldsymbol{B}_{i}\cdot\boldsymbol{\sigma} + \Delta\tau_{x})\boldsymbol{c}_{i}$$

where  $\sigma$  are the Pauli matrices in spin space and  $\tau$  are the matrices in particle hole space. Now the magnetic field **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

$$H = -\sum_{i} (B_{0} + \mu + t \frac{|\beta_{i}|^{2} + |\beta_{i-1}|^{2}}{2B_{0}})c_{i}^{\dagger}c_{i} + \sum_{i} \left[t\alpha^{*}c_{i}^{\dagger}c_{i+1} + t^{2}\frac{\beta_{i+1}^{*}\beta_{i}}{2B_{0}}c_{i}^{\dagger}c_{i+2} + \left(\frac{1}{2B_{0}} - \frac{1}{2\mu}\right)\Delta t\beta_{i}c_{i}c_{i+1} + h.c.\right]$$

$$(2.11)$$

where  $\alpha_n$  and  $\beta_n$  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 selfconsistency condition of the pairing  $\Delta$ . 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.

## **B** 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

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

where  $\psi_{\alpha}^{\dagger}(\vec{r})$  is the real-space creation operator for a spin  $\alpha$ ,  $h_{\alpha\beta}(\vec{r})$  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 approxiamtion is given by

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

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  $\gamma_n^{\dagger}$  and annihilation  $\gamma_n$  operators, inducing particle-hole symmetry in which the Hamiltonian is diagonal.

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

From now on we will discontinue writing the dependence of  $\vec{r}$  explicitly. Here  $\gamma_n$  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)

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

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

$$\begin{bmatrix} H_{\text{diag}}, \gamma_n \end{bmatrix} = -\varepsilon_n \gamma_n$$
  
$$\begin{bmatrix} H_{\text{diag}}, \gamma_n^{\dagger} \end{bmatrix} = \varepsilon_n \gamma_n^{\dagger}$$
  
(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

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

Comparing the factors in front of the  $\gamma$  operators this obeys the following matrix equation

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

25

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  $\Delta$ . 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  $\Delta$  which will lead us to the self consistency condition.

#### 3.2. Self-consistency condition

Next we calculate the pairing  $\Delta$  with the equation we got from the mean field approximation

$$\Delta = rac{V}{2} \left\{ \langle \psi_{\uparrow} \psi_{\downarrow} 
angle - \langle \psi_{\downarrow} \psi_{\uparrow} 
angle 
ight\}$$

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

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

In the mean field approximation we already mentioned that the energy  $\varepsilon_n$  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  $\omega_D$  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  $\Delta$  (for example choosing the same pairing  $\Delta$  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) selfconsistently. The index i denotes the number of iterations and  $\xi$  defines the error when the iteration is fullfilled. We set the error to  $10^{-4}$ 

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

$$\frac{|\Delta_i - \Delta_{i-1}|}{|\Delta_i|} < \xi, \tag{3.7}$$

where the index *i* describes the number of the iteration. We choose the value of the error  $\xi$  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:

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

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f_{i+a} - f_{i-a}}{2a}$$
$$\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} \approx \frac{f_{i+a} - 2f_i + f_{i-a}}{a^2}$$

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

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

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

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

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

$$H = -t \sum_{\langle ij \rangle} \left( \boldsymbol{c}_{i}^{\dagger} \boldsymbol{c}_{j} + h.c. \right) - \mu \sum_{i} \boldsymbol{c}_{i}^{\dagger} \boldsymbol{c}_{i} - \sum_{i} \boldsymbol{c}_{i}^{\dagger} \left( \boldsymbol{h}_{i} \cdot \boldsymbol{\sigma} \right) \boldsymbol{c}_{i} + i \frac{\hbar \alpha_{\mathrm{R}}}{2a} \sum_{\langle ij \rangle} \left( \boldsymbol{c}_{i}^{\dagger} \hat{\boldsymbol{z}} \cdot \left( \hat{\boldsymbol{d}}_{ij} \times \boldsymbol{\sigma} \right) \boldsymbol{c}_{j} + h.c. \right) + \sum_{i} \left( \Delta_{i} c_{i\uparrow}^{\dagger} c_{i\downarrow}^{\dagger} + h.c. \right).$$

$$(3.9)$$

Here  $c_i^{\dagger} = (c_{i\uparrow}^{\dagger}, c_{i\downarrow}^{\dagger})$ , where  $c_{i\alpha}^{\dagger}$  creates an electron with spin  $\alpha$  at lattice site i = (x, y). The symbol  $\langle ij \rangle$  indicates a summation over nearest-neighbor lattice sites and  $d_{ij}$  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

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

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

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

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

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

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  $\psi_i^{\dagger} = (\psi_{i\uparrow}^{\dagger}, \psi_{i\downarrow}^{\dagger})$ . We will aquire the gap closing condition by diagonalising this Hamiltonian which results in

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

This means that we get a gap closing if

$$\varepsilon(\mathbf{k})^{2} + (\alpha_{\rm R}^{*})^{2} \mathcal{L}_{0}(\mathbf{k})^{2} + (h_{z}^{*})^{2} + |\Delta^{*}|^{2} = \sqrt{\varepsilon(\mathbf{k})^{2}(\alpha_{\rm R}^{*})^{2} \mathcal{L}_{0}(\mathbf{k})^{2} + [\varepsilon(\mathbf{k})^{2} + |\Delta^{*}|^{2}](h_{z}^{*})^{2}}$$

which is equivalent to

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

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



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  $\mu$ . The parameters used here are:  $\alpha_{\rm R}^* = 0.3$ ,  $h_0^* = 2$ ,  $N_x = 25$  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:

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

In this definition  $\Phi$  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 a 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



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  $\mu^*$ . (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  $\mu^*$ . 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

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  $\Delta^*$  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 pertubations.

#### 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	$\omega_D$	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  $\xi$ . 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  $\xi$  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  $\omega_D$  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  $\mu^*$  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  $\mu^*$  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
(b) Superconducting gap



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  $\Delta^*$ . The arrows denote the phase of  $\Delta^*$  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 superconductor is a 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

(a) Probability density

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  $\alpha_R^*$ . 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.



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  $\mu^*$  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

$$(4+\mu^*)^2 + (\Delta^*)^2 = (h_0^*)^2, \quad (\mu^*)^2 + (\Delta^*)^2 = (h_0^*)^2, \quad (4-\mu^*)^2 + (\Delta^*)^2 = (h_0^*)^2.$$
(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  $x \to -x$  or  $y \to -y$ . 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  $\mu^*$  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  $\alpha$ ) and  $\mu^* = -4$  and  $h_0 = 0.4$  (point  $\beta$ ). 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



(a) Probability density

(b) Superconducting gap

Figure 4.16.: (a,b) correspond to point  $\alpha$  and (b,c) correspond to point  $\beta$ . (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  $\Delta^*$ . The arrows denote the phase of  $\Delta^*$  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  $\mu^*$ 



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  $\mu^*$  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  $\Delta^*$ . The arrows denote the phase of  $\Delta^*$  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

$$q = rac{1}{4\pi} \int \mathrm{d}x \; \mathrm{d}y \; oldsymbol{m} \cdot (\partial_x oldsymbol{m} imes \partial_y oldsymbol{m})$$

where 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.



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

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

where  $h_0^*$  is the strength of the magnetisation,  $\Phi = q\phi + \varphi$  where  $\phi$  is the polar coordinate in real space and  $\varphi$  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 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  $\Delta^*$ . We look at the black point for Majorana modes. Far away from this region, we will loose the localization. For a winding number q = 1 we do not find any localised 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  $\alpha_{\rm R}^* = 0.5$ ,  $N_x = 23$  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  $\Delta^*$ . 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 Néel skyrmion with winding number one on top of a superconductor with a vortex.

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  $\mu^*$  for  $\alpha_{\rm R}^* = 0.5$ ,  $N_x = 23$  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  $\Delta^*$ . The arrows denote the phase of  $\Delta^*$  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 skyrmionvortex 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 ferromagnetsuperconductor 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 spartially 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.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 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  $\psi$  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;

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

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 land odd for odd values of l,  $g_l(k) = (-1)^l g_l(k)$ , and superconductors with l = 0, 1, 2, ...are labelled as s, p, d, ...-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  $s_z$ :

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

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,

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

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  $i\boldsymbol{\sigma}\sigma_y = (i\sigma_x\sigma_y, i\sigma_y\sigma_y, i\sigma_z\sigma_y)$ ,

$$\Psi_{\text{triplet}}^{l} = (\boldsymbol{d}(k) \cdot \boldsymbol{\sigma}) i \sigma_{y} = (d_{x}(k)\sigma_{x}, d_{y}(k)\sigma_{y}, d_{z}(k)\sigma_{z}) i \sigma_{y}$$
$$= \begin{pmatrix} -d_{x}(k) + i d_{y}(k) & d_{z}(k) \\ d_{z}(k) & d_{x}(k) + i d_{y}(k) \end{pmatrix}$$

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

$$\Delta_{k}^{\text{s-wave}} = \Delta_{0} g(k) i \sigma_{y}$$
$$\Delta_{k}^{\text{p-wave}} = \Delta_{0} \left( d(k) \cdot \boldsymbol{\sigma} \right) i \sigma_{y}$$

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  $d(k) = (k_x \pm ik_y, i(k_x \pm ik_y), 0)$  is called  $p_x + ip_y$ pairing but we also use 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

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

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

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

#### 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.

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

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

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

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

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

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

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

```
import numpy as np
1
    import matplotlib.pyplot as plt
2
    import matplotlib
3
4
\mathbf{5}
    # ______
    # Definitions and functions
6
7
    # Sets the size of the plot
8
    matplotlib.rcParams.update({'font.size':20})
9
10
    # Define Pauli Matrices
11
    pauli = [[[1, 0], [0, 1]], [[0, 1], [1, 0]], [[0, -1j], [1j, 0]], [[1,
12
        0],[0,-1]]]
13
14
    def MuMatrix(k,mu):
    ....
15
16
    Purpose:
    Makes a Matrix corresponding to the onsite part in the Hamiltonian
17
18
    Definitions:
19
    k: Parameter that controls the momentum of the particles
20
    mu: Parameter for controlling the chemical potential of the system
21
    ......
22
    return (k**2-mu)*np.tensordot(pauli[3],np.identity(2),axes=0)
23
24
25
    def DeltaMatrix(D,k):
26
27
    Purpose:
    Makes a Matrix corresponding to the pairing part in the Hamiltonian
28
29
    Definitions:
30
    D: Parameter for controlling the BCS type pairing of electrons
31
    k: Parameter that controls the momentum of the particles
32
33
    0.0.0
34
    return 2*D*k*np.tensordot(pauli[2],np.identity(2),axes=0)
35
    def MagneticFieldMatrix(B):
36
    .....
37
38
    Purpose:
    Makes a Matrix corresponding to the hopping part in the Hamiltonian.
39
40
    Comment:
41
    All energies are scaled in terms of the hopping prameter t, therefore
42
        we choose it equals 1 here.
43
    Definitions:
44
    B: Parameter which controls the magnetic field
45
46
47
    return -B*np.tensordot(pauli[3],pauli[3],axes=0)
48
    def H_Bulk(k,mu,D,B):
49
```

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

68

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

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

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

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

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

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

```
SCF_BDG
demo
  🛑 2D Block on a 2D superconductor
    SampleScript.py
  Kitaev Chain
    SampleScript.py
  Kitaev Chain with 2D superconductor
    SampleScript.py
  Skyrmion-Vortex pairs
    SampleScript.py
  Vortex in superconductor
    SampleScript.py
  Phase Diagram
    LowestEnergyMode.py
SCF BDG
  ____init___.py
  MakeMatrices.py
  Plot.py
  RelaxCPU.py
  RelaxGPU.py
  🛑 vtk.py
  📥 xml.py
📥 setup.py
 README.md
```

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 Make-Matrices.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 Make-Matrices.py.

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

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

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

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

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

```
231
            VY[i,j] = h[nh+1]
           VZ[i,j] = h[nh+2]
232
233
       self.h= self.h0*h
234
       self.VX=VX
235
       self.VY=VY
236
       self.VZ=VZ
237
238
     def MakeSkyrmTextures(self):
239
       ....
240
       Purpose: Make a skyrmion texture. Which is a part of the H matrix
241
       If thetaSkyrm=0, the function returns Neel skyrmion, if thetaSkyrm=pi
242
           /2 the function return a Bloch skyrmion.
       ....
243
244
       x1= self.randPointsX
245
       x2= self.Nx - self.randPointsX
246
       y1= self.randPointsY
247
       y2= self.Ny - self.randPointsY
248
249
250
       sizel= x2 - x1
       size2= y2 - y1
251
252
       11= 0.1*np.double(min([size1, size2]))
253
254
       h= np.zeros(3*self.Nx*self.Ny)
255
256
       VX= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
257
       VY= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
258
       VZ= np.zeros(shape=(self.Nx, self.Ny), dtype=complex)
259
260
       for k in range(len(self.Srx)):
261
         for i in range(x1,x2):
262
           for j in range(y1,y2):
263
              nh= 3*i + 3*j*self.Nx
264
              x= np.double(i-self.Srx[k])
265
              y= np.double(j-self.Sry[k])
266
              z= x**2 + y**2 - 11**2
267
268
              N = x * * 2 + y * * 2 + 11 * * 2
269
              mxy= np.sin(self.Sp*np.arccos(z/N))
270
              mz= -self.Sq*np.cos(self.Sp*np.arccos(z/N))
              if (i==x1 or j==y1 or i==x2-1 or j==y2-1):
271
272
                h[nh] = 0.0
                h[nh+1] = 0.0
273
                h[nh+2] = -self.Sq
274
              else:
275
                h[nh] = h[nh] + np.sin(self.Sw*np.arctan2(x,y)+self.thetaSkyrm)
276
                    *mxv
                h[nh+1] = h[nh+1] + np.cos(self.Sw*np.arctan2(x,y)+self.
277
                    thetaSkyrm) *mxy
278
                h[nh+2] = h[nh+2] + mz
279
              VX[i,j] = h[nh]
```

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

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

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

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

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

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

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

Next we show RelaxCPU.py.

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

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

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\ \mbox{\#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
```

```
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
    # _______
   #Initialize Matrix
14
15
```

```
16
    #Set max iterations and error (in pairing potential):
17
    Matrix.Nmax=30
    Matrix.Maxerror= 0.00001
18
19
    #System size parameters:
20
    Matrix.Nx=60 #should be an odd (prime) number
21
    Matrix.Ny=11 #should be an odd (prime) number
22
23
    #System parameters (dimensionless):
24
    Matrix.alphaR= 0.3 # Rashba SOC
25
    Matrix.mu= mu # Chemical potential
26
    Matrix.h0 = h0 # strength of exchange field
27
    Matrix.thermE= 0.001 #thermal energy, kB*T
28
    Matrix.epsilonD= 2 #Debye frequency
29
30
    #Make magnetic skyrmion texture:
31
    Matrix.randPointsX = 2 #distance from boundary of sample to
32
        ferromagnetic block
    Matrix.randPointsY = 5
33
    Matrix.MakeFerroBlock()
34
35
    #Initialize pairing potential:
36
    Matrix.g= 5.0 #attractive coupling strength
37
    Matrix.delta= 1*np.ones(Matrix.Nx*Matrix.Ny,dtype=complex) #Homogenous
38
        order-parameter field
39
40
    #Make constant matrices:
^{41}
    Matrix.MakeConstantMatrices()
    Matrix.MakeHMatrix()
42
43
44
    RelaxGPU.RelaxGPU_NP(Matrix)
    DeltaMatrix = Matrix.MapFromVectorToMatrix(Matrix.delta)
45
    return [Matrix.EigenValues[int(len(Matrix.EigenValues)/2)],np.mean(abs(
46
        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 \text{ mu} = \text{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  $\mu^*$ . 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  $E_{\text{tot}}^*$  as well as in the superconducting gap  $\Delta^*$ . Because of this we chose  $\xi = 10^{-4}$  as a termination condition for the relaxation.



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 chaper 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  $\mu^*$  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 skyrmionvortex pair system in chapter 4.4, 4.5 and 4.6 we occured 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.



(b) with interpolation

(a) without 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  $\Delta^*$  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 Society Newsletter 11, 33 (2006).
- [2] R. P. Feynman, International Journal of Theoretical Physics 21, 467 (1982).
- [3] P. W. Shor, SIAM Journal on Computing 26, 1484 (1997).
- [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, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya, R. Barends, B. Burkett, Y. Chen, Z. Chen, A. Fowler, B. Foxen, M. Giustina, R. Graff, E. Jeffrey, T. Huang, J. Kelly, P. Klimov, E. Lucero, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, H. Neven, and J. M. Martinis, Science 360, 195 (2018).
- [7] J. I. Cirac and P. Zoller, Physical Review Letters 74, 4091 (1995).
- [8] L. Childress and R. Hanson, MRS Bulletin 38, 134 (2013).
- [9] D. Loss and D. P. DiVincenzo, Physical Review A 57, 120 (1998).
- [10] J. K. Pachos, *Introduction to Topological Quantum Computation* (Cambridge University Press, 2009) p. 9780511792908.
- [11] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. D. Sarma, Reviews of Modern Physics 80, 1083 (2008).
- [12] E. Majorana and L. Maiani, Nuovo Cim. (engl. Transl. Soryushiron Kenkyu) 63, 149 (1981).
- [13] A. Y. Kitaev, Physics-Uspekhi 44, 131 (2001).
- [14] L. Fu and C. L. Kane, Physical Review Letters **100**, 096407 (2008).
- [15] J. D. Sau, R. M. Lutchyn, S. Tewari, and S. D. Sarma, Physical Review Letters 104, 040502 (2010).
- [16] J. Alicea, Y. Oreg, G. Refael, F. von Oppen, and M. P. A. Fisher, Nature Physics 7, 412 (2011).
- [17] T.-P. Choy, J. M. Edge, A. R. Akhmerov, and C. W. J. Beenakker, Physical Review B 84, 195442 (2011).
- [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, Science Advances 4, eaar5251 (2018).
- [20] G. Yang, P. Stano, J. Klinovaja, and D. Loss, Physical Review B 93, 224505 (2016).
- [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, Semiconductor Science and Technology 27, 124003 (2012).
- [25] V. Lahtinen and J. Pachos, SciPost Physics 3, 021 (2017).
- [26] A. Kitaev, Annals of Physics **303**, 2 (2003).
- [27] D. A. Ivanov, Physical Review Letters 86, 268 (2001).
- [28] P. DE GENNES, Superconductivity of Metals and Alloys, Advanced Book Classics (Addison-Wesley Publishing Company, 1966).
- [29] K. M. Hals, M. Schecter, and M. S. Rudner, Physical Review Letters 117, 017001 (2016).
- [30] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, Physical Review 108, 1175 (1957).
- [31] M. Sato, Y. Takahashi, and S. Fujimoto, Physical Review B 82, 134521 (2010).
- [32] T. Čadež and P. D. Sacramento, Journal of Physics: Condensed Matter 28, 495703 (2016).
- [33] J. Li, T. Neupert, Z. Wang, A. H. MacDonald, A. Yazdani, and B. A. Bernevig, Nature Communications 7, 12297 (2016).
- [34] T. Liu, J. J. He, and F. Nori, Physical Review B 98, 245413 (2018).
- [35] K. Björnson and A. M. Black-Schaffer, Physical Review B 88, 024501 (2013).
- [36] K. Björnson and A. M. Black-Schaffer, Physical Review B 91, 214514 (2015).
- [37] E. W. J. Straver, J. E. Hoffman, O. M. Auslaender, D. Rugar, and K. A. Moler, Applied Physics Letters 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. L. Pokrovsky, Physical Review B 58, R8889 (1998).
- [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. Merill, 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. Espacially 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 June 5, 2019

Jonas Nothhelfer