Delft University of Technology

Bachelor thesis

Numerical simulations for type II superconductors

Finite Element Method for the time-dependent Ginzburg-Landau equations

Tobias Bonsen

supervised by
Dr.ir. F.J. Vermolen
Dr.ir. K. van Hoogdalem

November 14, 2017

Abstract

Superconductivity was discovered in 1911 and since then it has become indispensable in a wide range of fields. It is often accompanied by strong magnetic fields which can do away with the superconducting properties of the material. This process is described by the Ginzburg-Landau theory of superconductivity. In this report, this theory is discussed at length. The result is a system of two coupled, time-dependent partial differential equations that can be solved using numerical methods. A finite element method is constructed using standard Lagrangian and curl-conforming Nédélec elements. Numerical simulations were performed with Lagrangian and Nédélec elements in COMSOL and MATLAB respectively. Using Lagrangian elements delivers flawed results. Using Nédélec elements should improve these results but so far only parts of the problem have successfully been solved using these elements.

Contents

1 Introduction

The phenomenon of superconductivity was discovered in 1911 by Kamerlingh Onnes in Leiden. After he succeeded in producing liquid Helium, he was researching the properties of several metals at low temperatures (just a few Kelvin). He discovered that below a certain temperature, the metal's electrical resistance completely vanished. Since then, a lot of research has been conducted into this phenomenon and nowadays superconducting materials at higher temperatures are also available.

The property of no resistance means that very high currents can be achieved, which is useful in a wide range of fields. For example, MRI machines use superconducting magnets to create images of a person's body. Another important application is the usage of superconductors to create a Josephson junction. This consists of two superconductors, coupled by a weak link, and are the building blocks of very sensitive sensors or single-electron transistors. These junctions are also integral in quantum computing.

Superconductors are often used to achieve high currents, but according to the Maxwell equations, this results in high magnetic fields in and near the material as well. Superconductors have the property of excluding magnetic fields up to a certain strength. When the field exceeds this strength, it intrudes the material.

This process is modelled by a set of equations called the *Ginzburg-Landau equations*. These are a set of two coupled non-linear partial differential equations that can't be solved generally using analytical methods. Hence numerical approximations are needed. This is done by using the *Finite Element Method*, where a domain is divided into small elements and the solutions are approximated on each of these elements.

Such a model was already constructed by Alstrøm et al. [1] using the COMSOL Multiphysics [2] software, but the obtained results turn out to be flawed. The goal of this report is to get a good understanding of the Ginzburg-Landau model and to improve the existing FEM model using MATLAB.

To be able to develop a FEM model and properly interpret the results, some of the theory behind superconductivity will be discussed in chapter 2 and the process of creating a Finite Element model will be introduced. In Chapter 3, some of the results of Alstrøm et al. [1] will be reproduced using COMSOL Multiphysics and the flaws of these results will be discussed along with their possible causes. Then a FEM will be created for the Ginzburg-Landau equations in MATLAB [3]. In chapter 4, numerical simulations will be done using the program and the results will be compared to the ones obtained by using COMSOL. Finally, several conclusions will be drawn in chapter 5 and recommendations for future research will be made in chapter 6.

2 Theory

In this first section, some theoretical background will be discussed. Firstly, the derivation of the Ginzburg-Landau equations will be treated briefly and some insight will be given into the physical meaning of the terms that occur in these equations. Next, the equations will be simplified through normalization and gauge invariance. Finally, some general theory about differential equations such as the Ginzburg-Landau and a method to solve them will be discussed. As a main guideline for the physics section of this chapter, the book *Introduction to superconductivity* by M. Tinkham [4] was used. For the mathematical part, the book *Numerical methods in scientific computing* by J. van Kan et al. [5] was used as a reference.

2.1 Phenomena in Superconductivity

Superconductivity was first discovered by Kamerlingh Onnes in 1911 in his laboratory in Leiden. Just 3 years after he had succeeded in producing liquid Helium, he used this discovery to study the properties of metals at temperatures of just a few Kelvin. He discovered that the electrical resistance of certain metals completely vanished when cooled below a critical temperature, T_c, which differs for different materials. Experiments on superconducting current loops have shown that the lower bound for their decay is in the order of 10^5 years.[4]

Another important property of superconducting materials is that they exclude magnetic fields even when the sample was already in a magnetic field *before* it was cooled through T_c. This phenomenon was discovered by Meissner and Ochsenfeld and is called the *Meissner effect.* However, as shown in Figure 2.1, the magnetic field is able to intrude the superconducting sample to a certain depth, called the *penetration depth* λ. This effect is broken when the sample becomes subject to a certain *critical magnetic field* H_c.

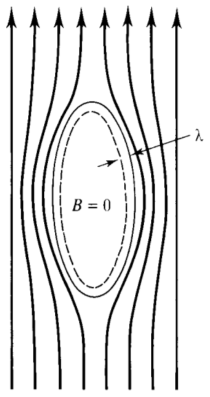


Figure 2.1: The Meissner effect: the applied magnetic field is screened from the interior of the superconductor by a penetration depth λ. [4]

2.2 Theories of superconductivity

After the discovery of superconductivity by Kamerlingh Onnes, much research has been conducted into this phenomenon, both on an experimental and a theoretical level. This research resulted into the development of several theories of superconductivity.

The first model that gave a good description of superconductivity was provided by the brothers F. and H. London [6]. They proposed two equations based on a phenomenological derivation.

$$\mathbf{E} = \frac{\partial}{\partial t}\left(\Lambda \mathbf{J}_s\right), \tag{2.1}$$

$$\mathbf{h} = -c\,\nabla \times \left(\Lambda \mathbf{J}_s\right), \tag{2.2}$$

where

$$\Lambda = \frac{4\pi\lambda_L^2}{c^2} = \frac{m}{n_s e^2}. \tag{2.3}$$

In these equations $\mathbf{E}$ and $\mathbf{h}$ are respectively the electric field and magnetic flux, $\mathbf{J}_s$ is the superconducting current density, e and m are the electron charge and mass, c is the speed of light, n_s is the *number density of superconducting electrons* and λ_L is a new parameter called the *London penetration depth.*

These equations, combined with the Maxwell equations, predicted that n_s would vary continuously from zero at the critical temperature to a limiting value of the order of the total number of conduction electrons for lower temperatures. The Meissner effect was also described by their model, predicting that a magnetic field would be exponentially screened from the interior of a superconductor with a certain penetration depth λ_L, given by:

$$\lambda_L = \left(\frac{mc^2}{4\pi n_s e^2}\right)^{1/2}. \tag{2.4}$$

The London equations provided a good description for superconducting phenomena. Inserting the maximal value for n_s, namely the total number of conducting electrons n, an ideal theoretical limit $\lambda_L(0)$ for the penetration depth as $T \to 0$ is found. However, experimental data found that the penetration depth λ is always larger than $\lambda_L(0)$. This called for a new concept, the coherence length, which was introduced by Pippard. [7]

Pippard proposed a nonlocal generalization of the London theory. The basis for this principle was the postulate that the superconducting current at a point $\mathbf{r}$ depends on the magnetic vector potential $\mathbf{A(r')}$ in a volume of radius ξ_0 around $\mathbf{r}$. This ξ_0 was called the *coherence length.*

He derived this new coherence length from a uncertainty argument: since only electrons within kT_c of the Fermi energy can participate in a phenomenon which occurs at T_c, their momentum range will be: $\Delta p \approx kT_c/v_F$ where v_F is the Fermi velocity, T_c is the critical temperature and k is the Boltzmann constant. This results in:

$$\Delta x \gtrsim \hbar v_F / kT_c,$$

where $\hbar$ is the reduced constant of Planck. This leads to the definition of the coherence length:

$$\xi_0 = a\frac{\hbar v_F}{kT_c}, \tag{2.5}$$

with a a constant in the order of 1 yet to be determined. Usually, $\xi_0 \gg \lambda$ and it represents the smallest size of a wave packet that the superconducting charge carriers can form (approximately). For a vector potential that does not maintain its full value over a volume of radius ξ_0, one would thus expect a weakened supercurrent. In the experiments, $\mathbf{A(r)}$ decreased sharply over a length $\lambda \ll \xi_0$, a weakened supercurrent resulted in an increased field penetration, and thus an increased apparent penetration depth. Pippard found that his new concept could fit the experimental data with $a = 0.15$, which was later validated by the microscopic theory.

Subsequent experiments on superconducting materials established the existence of an energy gap Δ between the ground state and the quasi-particle excitations of the system. Measurements on the specific heat of these materials showed that the minimum excitation energy per particle was about $1.5kT_c$. At about the same time, measurements of electromagnetic absorption in the region of $\hbar\omega \sim kT_c$ could be interpreted in terms of an energy gap of 3 to 4 times kT_c. This would mean the excitations were produced by pairs.

At this point, in 1957, BCS theory came along. Bardeen, Cooper and Schrieffer [8] postulated that weak attractions between electron's, caused by electron-phonon interaction , causes the formation of electron pairs, called *Cooper pairs* at low temperatures. These pairs, with equal and opposite momentum and spin, can be seen as the superconducting charge carriers that were anticipated in the phenomenological theories. The predictions of this theory, for example the size of the band gap, were in agreement with the experimental data.

2.3 Ginzburg-Landau theory of superconductivity

In 1950, seven years before the introduction of BCS theory, another phenomenological theory was produced by Ginzburg and Landau [9]. It focused on the superconducting electrons rather than on the excitations and it introduced a complex valued order parameter ψ. This parameter describes the local density of superconducting electrons by the following relation:

$$n_s = |\psi(x)|^2. \tag{2.6}$$

The main idea in deriving the so-called Ginzburg-Landau equations was that they were a result of using a variational principle of the free energy. Under the assumption that $|\psi|$ is small and so is its gradient, the free energy density of the system can be expanded in powers of $|\psi|^2$:

$$f = f_{n0} + \alpha|\psi|^2 + \frac{\beta}{2}|\psi|^4 + \frac{1}{2m^*}\left|\left(\frac{\hbar}{i}\nabla - \frac{e^*}{c}\mathbf{A}\right)\psi\right|^2 + \frac{h^2}{8\pi}, \tag{2.7}$$

where α and β are to be determined constants, m^* and e^* are the effective charge carrier mass and charge respectively and h is Planck's constant.

For $\psi = 0$, this reduces to $f = f_{n0} + h^2/8\pi$ which is the free energy of the non-superconducting state with $f_{n0}(T) = f_{n0}(0) - \frac{1}{2}\gamma T^2$. The remaining terms describe superconducting behaviour. They will now be discussed, along with their physical meaning.

The second and third term on the right hand side are a series expansion of the free energy in powers of $|\psi|^2$ or n_s, in which only the first two terms are retained. This is a sufficient approximation as long as the temperature does not deviate too much from T_c, where $|\psi|^2 \rightarrow 0$. It is immediately evident that in this expansion, β has to be greater than zero, for if this is not the case, the minimum free energy occurs at very large $|\psi|^2$, where the approximation clearly does not hold.

Now there are two possible situations, as shown in Figure 2.2. For $\alpha > 0$ the minimum free energy is found for $|\psi|^2 = 0$, corresponding to the normal state. For $\alpha < 0$, the minimum is found at $|\psi|^2 = -\frac{\alpha}{\beta}$, which gives the following minimum free energy(in the absence of fields and gradients):

$$f_s - f_n = \frac{-\alpha^2}{2\beta}. \tag{2.8}$$

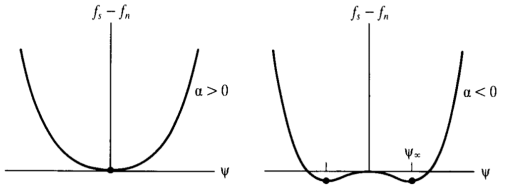


Figure 2.2: The free energy for cases $\alpha > 0$ and $\alpha < 0$, corresponding to respectively the normal and superconducting state. [4]

From this it becoomes clear that α is negative for $T < T_c$, the superconducting state and positive for $T > T_c$, the normal state. Performing a Taylor's series expansion of $\alpha(T)$ about T_c, neglecting higher order terms gives:

$$\alpha(T) = \alpha'\left(\frac{T}{T_c} - 1\right) \quad \alpha' > 0. \tag{2.9}$$

The final term in Equation (2.7) deals with fields and gradients. To make discussing this term more manageable, $\psi = |\psi|e^{i\phi}$ is used. This leaves the final term as:

$$\frac{1}{2m^*}\left[\hbar^2(\nabla|\psi|)^2 + \left(\hbar\nabla\phi - \frac{e^*\mathbf{A}}{c}\right)^2 |\psi|^2\right]. \tag{2.10}$$

In this equation, the first term gives the energy associated with gradients in the order parameter. The second term stands for the kinetic energy of the supercurrent in gauge-invariant form. Further **A** stands for the magnetic vector potential and is related to the magnetic and electric fields by:

$$\mathbf{B} = \nabla \times \mathbf{A}, \quad \mathbf{E} = -\nabla\phi - \frac{\partial \mathbf{A}}{\partial t}, \tag{2.11}$$

where ϕ is the electric potential(a scalar field).[10]

In the London theory discussed before, the gauge is constant so the term $\nabla\phi$ disappears. Equating the remainder to the kinetic energy for a supercurrent based on the London theory ($\mathbf{A}^2/8\pi\lambda_{eff}^2$), the following effective penetration depth is found:

$$\lambda_{eff}^2 = \frac{m^*c^2}{4\pi|\psi|^2 e^{*2}}, \tag{2.12}$$

which is in agreement with Equation (2.4), considering $n_s = |\psi(x)|^2$, except now the mass and charge are replaced by their effective counterparts. These effective values were determined by looking at the experimental data. It turns out that the combination $e^* = 2e$, $m^* = 2m$ and $n_s = \frac{1}{2}n$ fitted the data best. Eventually these values were also verified by derivation from the microscopic theory, corresponding to the Cooper pairs mentioned before.

2.3.1 The GL Differential Equations

The minimization of the free energy in Equation (2.7) was already discussed in the absence of fields, currents or gradients. Now, if these are imposed, $\psi(\mathbf{r}) = |\psi(\mathbf{r},t)|e^{i\phi(\mathbf{r},t)}$ has to minimize the overall free energy, given by the volume integral of Equation (2.7). Carrying out this process turns out to be very difficult and is beyond the scope of this project. Assuming a gapless density of state spectrum, Gor'kov and Eliashberg[11] were able to obtain the time-dependent Ginzburg-Landau equations given below:

$$\frac{\hbar^2}{2m^*D}\left(\frac{\partial}{\partial t} + i\frac{e^*}{\hbar}\phi\right)\psi = -\alpha\psi - \frac{\beta}{2}|\psi|^2\psi - \frac{1}{2m^*}\left(\frac{\hbar}{i}\nabla - \frac{e^*}{c}\mathbf{A}\right)^2\psi. \tag{2.13}$$

$$\sigma\left(\frac{\partial \mathbf{A}}{\partial t} + \nabla\phi\right) = \frac{e^*\hbar}{2m^*i}\left(\psi^*\nabla\psi - \psi\nabla\psi^*\right) - \frac{e^{*2}}{m^*}|\psi|^2\mathbf{A} - \frac{1}{\mu_0}\nabla\times\nabla\times\mathbf{A}. \tag{2.14}$$

In order for the problem to be well-defined, appropriate boundary conditions need to be specified. The three conditions that are required are listed below.

$$\left(\frac{\hbar}{i}\nabla\psi - \frac{e^*}{c}\mathbf{A}\psi\right)\cdot\mathbf{n} = 0, \quad \text{on } \partial\Omega, \tag{2.15}$$

which is related to the superconducting current. It says no superconducting current is allowed to pass the surface. Later on, it was shown that a term $\frac{i\hbar}{b}\psi$ should be added to the right hand side of this equation to account for a proximity effect.

$$\left(\frac{\partial\mathbf{A}}{\partial t} + \nabla\phi\right)\cdot\mathbf{n} = 0, \quad \text{on } \partial\Omega, \tag{2.16}$$

which restricts the regular current on the boundary. It says $\mathbf{E}\cdot\mathbf{n} = 0$ on $\partial\Omega$ so no normal current $\mathbf{J}_n = \sigma\mathbf{E}$ passes through the surface.

$$\nabla\times\mathbf{A} = \mathbf{B}_a, \quad \text{on } \partial\Omega, \tag{2.17}$$

where $\mathbf{B}_a$ is the applied magnetic field on the boundary.

The great advantage of these equations is that they are able to describe the intermediate state where both the superconducting and normal states exist, that is near H_c. The interface between two such states is shown schematically in Figure 2.3.

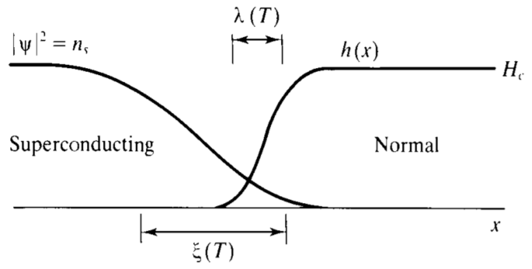


Figure 2.3: Interface between a superconducting and normal state. Here λ is the penetration depth and ξ is the coherence length, which will be discussed below. Here h denotes the one dimensional magnetic flux. [4]

2.3.2 The Ginzburg-Landau Coherence length

An important result of the differential equations is the *Ginzburg-Landau coherence length.* The argument that leads to this parameter goes as follows: consider the simplified case where no fields are present. ψ can then be taken to be real and the *normalized wave function* $f = \psi/\psi_\infty$ can be introduced, where $\psi_\infty^2 = -\alpha/\beta > 0$. Equation (2.34) then becomes(in one dimension):

$$\frac{d^2f}{dx^2} + f - f^3 = 0. \tag{2.18}$$

This shows that ψ varies over a volume with the characteristic length of:

$$\xi^2(T) = \frac{\hbar^2}{2m^*|\alpha(T)|} \propto \frac{1}{1 - T/T_c}. \tag{2.19}$$

This is called the Ginzburg-Landau coherence length. It is generally not the same as the Pippard coherence length ξ_0 that has been discussed before. To see how they are related a linearized solution of Equation (2.18) is found. For this, $f(x) = 1 + g(x)$ where $g(x) \ll 1$ is substituted to find:

$$g(x) \sim e^{\pm\sqrt{2}x/\xi(T)}. \tag{2.20}$$

This shows that a small disturbance in the wave function ψ from ψ_∞ will decay in a length of order $\xi(T)$.

Using results from the microscopic BCS theory, the value of $\xi(T)$ was calculated in terms of ξ_0. For pure superconductors at very low temperatures ($\ll T_c$) this resulted in $\xi(T) \approx \xi_0$. At temperatures near T_c, this led to:

$$\xi(T) = 0.74 \frac{\xi_0}{(1 - T/T_c)^{1/2}}. \tag{2.21}$$

Finally, a very important dimensionless parameter κ is introduced, using both the coherence length and another length that is closely related to the London penetration depth. This new length is also a result from BCS theory and is denoted by λ_{eff}. The dimensionless parameter κ is now defined as follows:

$$\kappa = \frac{\lambda_{\text{eff}}(T)}{\xi(T)}. \tag{2.22}$$

Using the numerical results for pure superconductors this definition leads to:

$$\kappa = 0.96\, \frac{\lambda_L(0)}{\xi_0}. \tag{2.23}$$

2.4 Type I and Type II Superconductors

The parameter κ from the last section plays an important role in the behavior of superconducting materials. To see the influence of this parameter on the behavior of a superconductor, two cases are considered: $\kappa \ll 1$ and $\kappa \gg 1$. Superconductors with these values for κ are respectively called *type I and type II superconductors.* For these two cases, the interface between normal and superconducting states is depicted in one dimension in Figure 2.4.

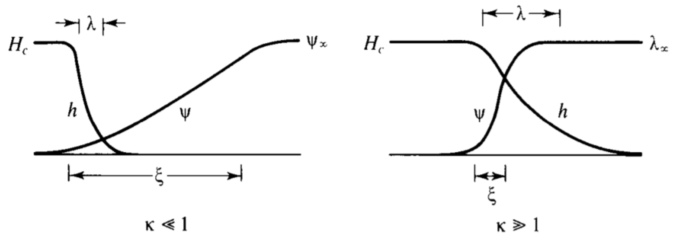


Figure 2.4: The interface between normal and superconducting states for type I (left) and type II (right) superconductors. [4]

These types react differently to an external magnetic field. To see why an additional concept is introduced: the *surface energy* γ. This is a phenomenological energy term associated with the NS interface. This term is derived from a free energy minimization argument and is:

$$\gamma = \frac{H_C^2}{8\pi}\delta, \tag{2.24}$$

where $\delta \approx \xi - \lambda$. For type I superconductors δ is positive, which leads to a positive surface energy associated with the boundary. The interfaces will arrange themselves in a way to minimize the overall energy. In this case this means a minimization of the interface area. An example of this for an infinite flat slab is shown below. An external magnetic field B_a is applied. This magnetic field is screened from the interior of superconducting regions, but some holes arise where the magnetic field penetrates the sample. These are regions in the normal state and are arranged in such a way that the interface area is minimal. These regions are in this case of macroscopic dimensions.

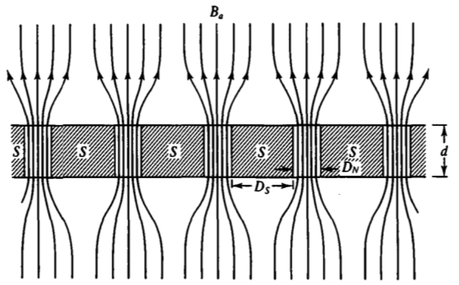


Figure 2.5: Field penetration for an infinite slab with $\kappa \ll 1$. The NS interface area is minimized. The normal region thickness D_N is of macroscopic size. [4]

For type II superconductors however, δ becomes negative and so does the surface energy. This leads to an overall maximization of the interface area. This is achieved through having a great number of microscopic non-superconducting holes in the sample. In theory the area would become infinite with an infinite number of infinitely small holes. However, there is a limit to the size of these holes. This limit is determined by *magnetic flux quantization*, which prescribes the minimal magnetic flux through a hole. This magnetic flux quantum is defined as:

$$\Phi_0 = \frac{hc}{2e}. \tag{2.25}$$

The resulting non-superconducting holes are called Abrikosov vertices and contain a single flux quantum each. An example of such an arrangement is shown below. The holes are now on the microscopic scale.

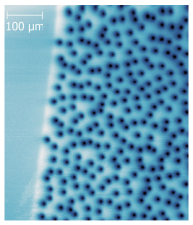


Figure 2.6: Abrikosov vertices in in 200 nm thick YBCO film taken by Scanning SQUID Microscopy after field cooling at 6.93 T to 4 K. [12]

2.5 Normalization and Gauge Invariance

To make the mathematics in the remainder of this report more clear, the GL-equations will be normalized and later a gauge will be chosen. For the normalization, the following transformations are used, where the primes mark dimensionless variables[1]:

$$\begin{gathered}(x,y,z,t) = \left(\lambda x', \lambda y', \lambda z', \frac{\xi^2}{D}t'\right), \qquad \mathbf{A} = \frac{c\hbar}{e^*\xi}\mathbf{A}' \\ \psi = \sqrt{-\frac{\alpha}{\beta}}\psi', \quad \phi = -\alpha D\kappa^2\sqrt{\frac{2\mu_0}{b}}\phi', \quad \sigma = \frac{1}{\mu_0 D\kappa^2}\sigma'.\end{gathered} \tag{2.26}$$

Substituting these dimensionless variables, the normalized GL-equations are:

$$\left(\frac{\partial}{\partial t} + i\kappa\phi\right)\psi = -\left(\frac{i}{\kappa}\nabla + \mathbf{A}\right)^2\psi - \psi - |\psi|^2\psi. \tag{2.27}$$

$$\sigma\left(\frac{\partial \mathbf{A}}{\partial t} + \nabla\phi\right) = \frac{1}{2i\kappa}\left(\psi^*\nabla\psi - \psi\nabla\psi^*\right) - |\psi|^2\mathbf{A} - \nabla\times\nabla\times\mathbf{A}. \tag{2.28}$$

and the boundary conditions transform into:

$$\left(\frac{\hbar}{i}\nabla\psi + \mathbf{A}\psi\right)\cdot\mathbf{n} = 0, \quad \text{on } \partial\Omega. \tag{2.29}$$

$$\left(\frac{\partial\mathbf{A}}{\partial t} + \nabla\phi\right)\cdot\mathbf{n} = 0, \quad \text{on } \partial\Omega. \tag{2.30}$$

$$\nabla\times\mathbf{A} = \mathbf{B}_a, \quad \text{on } \partial\Omega. \tag{2.31}$$

These equations have the property of gauge invariance. This means any particular gauge can be chosen, leading to the same outcome in the measurable physical quantities. This freedom of choice is due to the fact that quantities like $\mathbf{A}$ are not directly measurable. Only the magnetic field $\mathbf{B} = \nabla\times\mathbf{A}$ is measurable, and therefore it is fixed. This leaves infinite choices for $\mathbf{A}$, which is called *gauge freedom*. One can perform so called *gauge transformations* on the different potentials, without changing the physical outcome. In this case the gauge transformation is defined by a scalar function $\chi(x, y, z, t)$: [1][13]

$$\tilde{\psi} = \psi e^{i\kappa\chi}, \quad \tilde{\mathbf{A}} = \mathbf{A} + \nabla\chi, \quad \tilde{\phi} = \phi - \frac{\partial\chi}{\partial t}. \tag{2.32}$$

To obtain the final version of the GL-equations that will later be numerically solved, a convenient gauge is chosen. This gauge is the one with zero electrical potential, that is $\tilde{\phi} = 0$. This means that:

$$\frac{\partial\chi}{\partial t} = \phi. \tag{2.33}$$

Applying this transformation, the final version of the dimensionless time-dependent Ginzburg-Landau differential equations is found. The tildes are omitted for ease.

$$\frac{\partial\psi}{\partial t} = -\left(\frac{i}{\kappa}\nabla - \mathbf{A}\right)^2\psi + \psi - |\psi|^2\psi. \tag{2.34}$$

$$\sigma\frac{\partial\mathbf{A}}{\partial t} = \frac{1}{2ik}\left(\psi^*\ \nabla\psi - \psi\ \nabla\psi^*\right) - |\psi|^2\mathbf{A} - \nabla\times\ \nabla\times\ \mathbf{A}. \tag{2.35}$$

$$\nabla\psi\cdot\mathbf{n} = 0, \quad \text{on } \partial\Omega. \tag{2.36}$$

$$\nabla\times\mathbf{A} = \mathbf{B}_a, \quad \text{on } \partial\Omega. \tag{2.37}$$

$$\mathbf{A}\cdot\mathbf{n} = 0, \quad \text{on } \partial\Omega. \tag{2.38}$$

2.6 Partial Differential Equations

A *Partial Differential Equation (PDE)* is an equation in which one or more multivariate functions and their partial derivatives with respect to these different variables appear. Such equations describe most physical processes and can be solved analytically only in very special cases. The GL-equations form a perfect example of such a physical process. In this case, the functions are ψ and $\mathbf{A}$ and they are both dependent on the spatial variables and time. Unfortunately, in this case there is no way to derive a solution to this problem through analytic methods. Therefore, a solution will have to be found numerically. There are several numerical methods that are used in solving PDE's. These include the finite element method (FEM), finite volume methods (FVM) and finite difference methods (FDM). In this report, the finite element method is used for the spatial variables and a finite difference method will be used for the time derivatives. Both of these methods will be described below.

2.7 Finite Element Method

The *finite element method*, FEM for short, is a numerical method for solving problems in a wide range of fields. It subdivides the problem into smaller problems which are called elements. FEM formulates the problem as a system of algebraic equations, through which the solution is approximated at a discrete number of points over the domain.[14]

In the process of developing a finite element model, the first step is to express the problem in its weak formulation. In this section, the weak formulation for the time-dependent GL-equations will be obtained. Next, the unknown functions or variables in the problem are approximated by a linear combination of a set of basis functions. The choice of these functions depends on the type of problem and has to do with certain function spaces, which will be briefly discussed after. Furthermore, the proper choice of basis functions for the problem at hand will be discussed.

2.7.1 Weak formulation

Once a partial differential equation is expressed in its so-called *weak formulation*, it is no longer required to hold absolutely and the solutions to the obtained problem are only with respect to certain *test functions*.[15] This means that some of the requirements implied in the differential equation such as differentiability are no longer required to hold. For example, the solution of the heat equation in its weak form is only required to be differentiable once, while the original formulation the solution has to be twice differentiable. [5]

To obtain the weak formulation of a partial differential equation, it is multiplied by an arbitrary test function η in the so called *solution space* Σ. Eventually the solution will be approximated by functions from the same space. This choice of the same space for test and basis functions is what makes the FEM in this report a *standard Galerkin method*. The proper choice of this space will be discussed in the next section. The equation is then integrated over the domain Ω.

Weak formulation for the time-dependent GL-equations

This procedure will now be carried out for the time-dependent Ginzburg-Landau equations. For the first equation, this results in:

$$\int_\Omega \frac{\partial\psi}{\partial t}\,\eta\, d\Omega = \int_\Omega \left(\frac{1}{\kappa^2}\Delta\psi - \frac{i}{\kappa}\nabla\cdot(\mathbf{A}\psi) - \frac{i}{\kappa}\mathbf{A}\cdot\nabla\psi - \mathbf{A}^2\psi + \psi - |\psi|^2\psi\right)\,\eta\, d\Omega,$$

which leads to:

$$\begin{aligned}\int_\Omega \frac{\partial\psi}{\partial t}\,\eta\, d\Omega = &\int_\Omega \frac{1}{\kappa^2}\Delta\psi\;\eta\; d\Omega - \int_\Omega \frac{i}{\kappa}\nabla\cdot(\mathbf{A}\psi)\;\eta\; d\Omega\\ &-\int_\Omega \frac{i}{\kappa}\mathbf{A}\cdot\nabla\psi\;\eta\; d\Omega + \int_\Omega \left(1-\mathbf{A}^2-|\psi|^2\right)\psi\;\eta\; d\Omega.\end{aligned}$$

Through Green's first identity[16] the first term on the right hand side is rewritten and the product rule is used to expand the $\nabla\cdot(\mathbf{A}\psi)$-term:

$$\begin{aligned}\int_\Omega \frac{\partial\psi}{\partial t}\,\eta\, d\Omega = &-\frac{1}{\kappa^2}\int_\Omega \nabla\eta\cdot\nabla\psi\; d\Omega + \frac{1}{\kappa^2}\oint_{\partial\Omega}\eta\,(\nabla\psi\cdot\mathbf{n})\,d\,(\partial\Omega)\\ &-\int_\Omega \frac{i}{\kappa}(\nabla\cdot\mathbf{A})\psi\;\eta\; d\Omega - \int_\Omega \frac{2i}{\kappa}\mathbf{A}\cdot\nabla\psi\;\eta\; d\Omega\\ &+\int_\Omega \left(1-\mathbf{A}^2-|\psi|^2\right)\psi\;\eta\; d\Omega.\end{aligned}$$

Boundary condition (2.36) then completes the weak formulation of Equation (2.34):

$$\begin{aligned}\int_\Omega \frac{\partial\psi}{\partial t}\,\eta\, d\Omega = &-\frac{1}{\kappa^2}\int_\Omega \nabla\eta\cdot\nabla\psi\; d\Omega + -\int_\Omega \frac{i}{\kappa}(\nabla\cdot\mathbf{A})\psi\;\eta\; d\Omega\\ &-\int_\Omega \frac{2i}{\kappa}\mathbf{A}\cdot\nabla\psi\;\eta\; d\Omega + \int_\Omega \left(1-\mathbf{A}^2-|\psi|^2\right)\psi\;\eta\; d\Omega.\end{aligned} \tag{2.39}$$

For the second equation which is vector valued, a vector valued test function $\boldsymbol{\eta}$ has to be used, which has to be part of a different function space, which will be discussed later on. The inner product of this function with both the left and right hand side of the equation is taken and the result is integrated over Ω:

$$\begin{aligned}\sigma\int_\Omega \frac{\partial\mathbf{A}}{\partial t}\cdot\boldsymbol{\eta}\; d\Omega = &\frac{1}{2ik}\int_\Omega (\psi^*\;\nabla\psi - \psi\;\nabla\psi^*)\cdot\boldsymbol{\eta}\; d\Omega\\ &-\int_\Omega |\psi|^2\,\mathbf{A}\cdot\boldsymbol{\eta}\; d\Omega - \int_\Omega (\nabla\times\nabla\times\mathbf{A})\cdot\boldsymbol{\eta}\; d\Omega.\end{aligned}$$

To deal with the last term on in this equation, a rule for the curl-curl is needed that is similar to the green identity that was used for the Laplacian in Equation (2.34). This rule is the following:

Theorem 1. *For smooth functions $\boldsymbol{u}$ and $\boldsymbol{v}$ on a domain Ω the following integration by parts formula holds:*

$$\int_\Omega (\nabla \times \boldsymbol{u}) \cdot \boldsymbol{v} \; d\Omega = \int_\Omega \boldsymbol{u} \cdot (\nabla \times \boldsymbol{v}) \, d\Omega - \oint_{\partial\Omega} (\boldsymbol{u} \times \boldsymbol{n}) \cdot \boldsymbol{v} \; d\left(\partial\Omega\right),$$

where $\boldsymbol{n}$ is the outward pointing normal vector on $\partial\Omega$.

Proof. The statement is derived from a standard identity in vector calculus:

$$\nabla \cdot (\boldsymbol{u} \times \boldsymbol{v}) = (\nabla \times \boldsymbol{u}) \cdot \boldsymbol{v} - (\nabla \times \boldsymbol{v}) \cdot \boldsymbol{u}.$$

This is integrated over the domain to obtain:

$$\int_\Omega \nabla \cdot (\boldsymbol{u} \times \boldsymbol{v}) \; d\Omega = \int_\Omega \left((\nabla \times \boldsymbol{u}) \cdot \boldsymbol{v} - (\nabla \times \boldsymbol{v}) \cdot \boldsymbol{u} \right) \; d\Omega.$$

Using Gauss's theorem this leads to:

$$\oint_{\partial\Omega} \boldsymbol{n} \cdot (\boldsymbol{u} \times \boldsymbol{v}) \; d\left(\partial\Omega\right) = \int_\Omega \left((\nabla \times \boldsymbol{u}) \cdot \boldsymbol{v} - (\nabla \times \boldsymbol{v}) \cdot \boldsymbol{u} \right) \; d\Omega.$$

Because a scalar triple product is invariant under a circular shift, the following also holds:

$$\boldsymbol{n} \cdot (\boldsymbol{u} \times \boldsymbol{v}) = \boldsymbol{v} \cdot (\boldsymbol{n} \times \boldsymbol{u}) = \boldsymbol{u} \cdot (\boldsymbol{v} \times \boldsymbol{n}).$$

Combining these two results obtains:

$$\int_\Omega \boldsymbol{v} \cdot (\boldsymbol{n} \times \boldsymbol{u}) \; d\Omega = \int_\Omega \left((\nabla \times \boldsymbol{u}) \cdot \boldsymbol{v} - (\nabla \times \boldsymbol{v}) \cdot \boldsymbol{u} \right) \; d\Omega.$$

This leads to the required equality. □

We apply Theorem (1) to the previous result with $\boldsymbol{u} = \nabla \times \mathbf{A}$ and $\boldsymbol{v} = \boldsymbol{\eta}$ to obtain:

$$\begin{aligned}\sigma \int_\Omega \frac{\partial \mathbf{A}}{\partial t} \cdot \boldsymbol{\eta} \, d\Omega = \frac{1}{2ik} \int_\Omega (\psi^* \;\nabla\psi - \psi \;\nabla\psi^*) \cdot \boldsymbol{\eta} \; d\Omega - \int_\Omega |\psi|^2 \, \mathbf{A} \cdot \boldsymbol{\eta} \; d\Omega \\ - \int_\Omega (\nabla \times \mathbf{A}) \cdot (\nabla \times \boldsymbol{\eta}) \, d\Omega + \oint_{\partial\Omega} ((\nabla \times \mathbf{A}) \times \boldsymbol{n}) \cdot \boldsymbol{\eta} \, d\left(\partial\Omega\right).\end{aligned}$$

Boundary condition (2.37) then leaves us with:

$$\begin{aligned}\sigma \int_\Omega \frac{\partial \mathbf{A}}{\partial t} \cdot \boldsymbol{\eta} \, d\Omega = \frac{1}{2ik} \int_\Omega (\psi^* \;\nabla\psi - \psi \;\nabla\psi^*) \cdot \boldsymbol{\eta} \; d\Omega - \int_\Omega |\psi|^2 \, \mathbf{A} \cdot \boldsymbol{\eta} \; d\Omega \\ - \int_\Omega (\nabla \times \mathbf{A}) \cdot (\nabla \times \boldsymbol{\eta}) \, d\Omega + \oint_{\partial\Omega} (\mathbf{B}_a \times \boldsymbol{n}) \cdot \boldsymbol{\eta} \, d\left(\partial\Omega\right).\end{aligned} \tag{2.40}$$

2.7.2 Function spaces

In deriving the weak formulation of the GL-equations, some assumptions are made about the used test functions. For example, in the derivation of Equation (2.39), it is assumed that $\nabla\eta$ exists and is square integrable over the domain. These requirements determine the solution space, the set of functions the test and later basis functions are chosen from. In deriving the weak formulation of the GL-equations, two function spaces turn up. They are called *Sobolev spaces* and have to do with the existence and integrability of the curl and gradient. First, an auxiliary space of all *Lebesque integrable* functions is defined as:

$$L^2(\Omega) = \left\{ v(\mathbf{x}) : \Omega \to \mathbb{R} : \int_\Omega |v|^2 d\Omega < \infty \right\}, \tag{2.41}$$

The Sobolev space related to the first GL-equation then becomes:

$$H^1(\Omega) = \left\{ v(\mathbf{x}) \in L^2(\Omega, \mathbb{R}^d) : \frac{\partial v}{\partial x_i} \in L^2(\Omega, \mathbb{R}^d) \right\}, \tag{2.42}$$

for the first equation. Here L^2 denotes the space of all square integrable functions. For the second equation the curl of the test functions are required to exist and be square integrable. For complete clarity, a distinction has to be made between the rotation in 2D and 3D. The used definitions for the curl of a vector valued function $\boldsymbol{w} : \Omega \to \mathbb{R}^d$ are:

$$\begin{aligned} &\textbf{2D:}\ \ \text{curl}\ \boldsymbol{w} = \frac{\partial w_2}{\partial x} - \frac{\partial w_1}{\partial y} \\ &\textbf{3D:}\ \ \text{curl}\ \boldsymbol{w} = \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ w_1 & w_2 & w_3 \end{vmatrix} = \begin{bmatrix} w_{3,y} - w_{2,z} \\ w_{1,z} - w_{3,x} \\ w_{2,x} - w_{1,y} \end{bmatrix}. \end{aligned} \tag{2.43}$$

Now, the Sobolev spaces related to the curl are:

$$H^{curl}(\Omega) = \begin{cases} \left\{ \boldsymbol{v}(\mathbf{x}) \in L^2(\Omega, \mathbb{R}^2) : \text{curl}\ \boldsymbol{v} \in L^2(\Omega) \right\}, & \text{if } d = 2 \\ \left\{ \boldsymbol{v}(\mathbf{x}) \in L^2(\Omega, \mathbb{R}^3) : \text{curl}\ \boldsymbol{v} \in L^2(\Omega, \mathbb{R}^3) \right\}, & \text{if } d = 3 \end{cases}. \tag{2.44}$$

It should be clear that $H^1(\Omega) \subseteq H^{curl}(\Omega)$. The test functions for the first and second GL-equation will respectively have to be chosen from spaces H^1 and H^{curl}.

2.7.3 Basis functions

Eventually, the approximation of the problem will be made by a linear combination of basis functions, which are chosen from the same function space as the test functions that are used. So, eventually, a typical approximation will be of the form:

$$u = \sum_{i=1}^{n} a_i \eta_i, \tag{2.45}$$

where u is the unknown in the original problem, η_i are the basis functions with their corresponding coefficients a_i. These coefficients are the *degrees of freedom (dof)* of the problem. To approximate the solution of the problem, Equation (2.45) is then substituted into the weak formulation of the problem. Finally, substituting the same basis functions one by one as test function leads to a system of linear equations that can be solved for the unknown coefficients a_i.

It is the goal of this section to find appropriate basis functions for each of the GL-equations.

Mesh

Before the choice of basis functions can be discussed, the concept of a *mesh* needs to be explained. The reason for this is that the definition of the basis functions relies on this mesh. A mesh is a partition of the entire domain Ω. A part of this partition is called an *element* and how this mesh is constructed is dependent on the dimension. The simplest example is a uniform mesh in the 1D case, which for $\Omega = [0, 1]$ is shown in Figure 2.7.

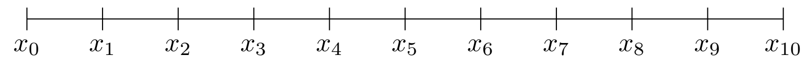


Figure 2.7: Uniform 11 point mesh on $\Omega = [0, 1]$, where $0 = x_0 < x_1 < \ldots < x_{n+1} = 1$.

Hence the total interval is divided into N subintervals, or elements $e_k = [x_{k-1}, x_k]$. In general an *element* is a simplex in $\mathbb{R}^d$. The union of all elements in the mesh should form the entire domain. In 2D, the most popular elements are triangular elements and in 3D tetrahedral elements are often used. In 2D, a possible mesh for $\Omega = [-1, 1] \times [-1, 1]$ is shown in Figure 2.8. The elements are now formed by triangular control volumes.

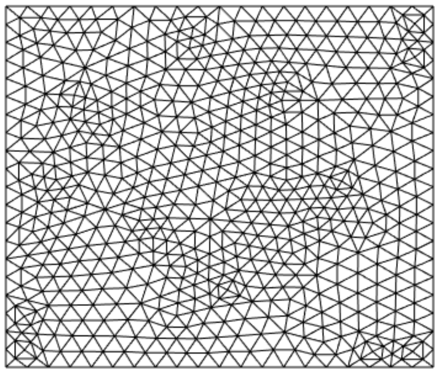


Figure 2.8: Example of a 2D mesh on the domain $\Omega = [-1,1] \times [-1,1]$. The mesh was constructed using the pde mesher tool of MATLAB.[3]

As can be seen, this mesh is no longer uniform over the domain. This is because the mesh was constructed by the pde mesher of MATLAB [3]. This program uses *Delaunay triangulation* to create the mesh. [17] This algorithm maximizes the minimal angle of all triangles in the mesh. This avoids discretization errors and large errors in the derivative of the solution. [18] The result is a set of N_p points $\mathbf{x}_i$ and N_e edges r_m between these points, forming the N triangular elements e_k.

Based on the constructed mesh, basis functions are formulated. They are often defined element-wisely and are nonzero only in a particular set of elements. Basis functions can be defined point-wisely or edge-wisely, which means they will have a dof related to a certain point or edge of the mesh. The way this works will be the subject of the remainder of this section.

Lagrangian $\mathbb{P}_1$ basis

If the approximated solution exists in the space $H^1(\Omega)$, a common choice of basis functions is *Lagrangian polynomials*, that are defined element-wisely. Depending on the required accuracy, a certain order of polynomials can be chosen. The most simple choice is first order. The basis functions are then called linear Lagrangian interpolating polynomials and to define them the following subspace of $H^1(\Omega)$ is introduced:

$$V^1(\Omega) = \left\{ v \in H^1(\Omega) : v|_{e_k} \in \mathbb{P}_1(e_k), 1 \leq k \leq N \right\}, \tag{2.46}$$

where $\mathbb{P}_m(e_k)$ denotes the space of all polynomials of order m on element e_k.

The Lagrangian basis functions $l_i(\mathbf{x})$ are then defined by the following [5]:

$$\begin{aligned} &(1)\ l_i(\mathbf{x}) \in V^1(\Omega) \\ &(2)\ l_i(\mathbf{x}_j) = \delta_{ij}, \end{aligned} \tag{2.47}$$

where $i, j \in 1, ..., N_p$. This means that each point in the mesh corresponds to a basis function and the coefficient of this basis function is the value of the unknown that is solved for in point x_i. A linear Lagrangian polynomial in 1D is shown in Figure 2.9.

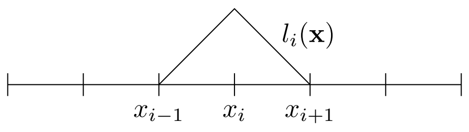


Figure 2.9: Linear Lagrangian polynomial on a uniform mesh in 1D.

This corresponds to the following functions on the domain:

$$l_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}] \\ 0, & \text{elsewhere.} \end{cases} \tag{2.48}$$

In 2D, the same definition (2.47) is used. Again, these polynomials are defined element-wisely. A typical triangular element is shown in Figure 2.10.

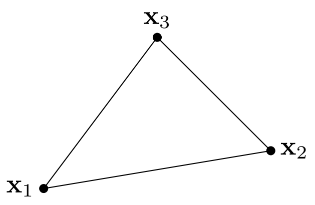


Figure 2.10: Element of a 2D mesh.

A linear polynomial in 2D is defined by:

$$l_i(\mathbf{x}) = b_0^i + b_1^i x + b_2^i y. \tag{2.49}$$

To define the linear Lagrangian polynomials in 2D, the parameters b_j^i should be computed. Using Definition (2.47), this is done by solving the following system of linear equations (which can easily be done):

$$\begin{bmatrix} 1 & x^1 & y^1 \\ 1 & x^2 & y^2 \\ 1 & x^3 & y^3 \end{bmatrix} \begin{bmatrix} b_0^1 & b_0^2 & b_0^3 \\ b_1^1 & b_1^2 & b_1^3 \\ b_2^1 & b_2^2 & b_2^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.50}$$

The resulting polynomial is shown in Figure 2.11.

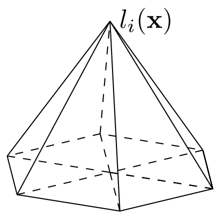


Figure 2.11: Linear Lagrangian polynomial on a 2D mesh.

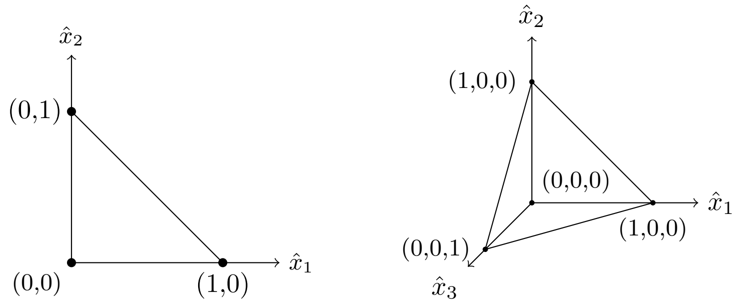
Nédélec linear basis
For solutions that exist in $H^{curl}(\Omega)$, one could use the same set of nodal basis functions, since $H^1(\Omega) \subseteq H^{curl}(\Omega)$. However, to find a more accurate approximation of the solution, different so-called *curl-conforming* basis functions can be used. These basis functions span a bigger part of $H^{curl}(\Omega)$ than a Lagrangian basis would. The most commonly used curl-conforming basis is called the *Nédélec basis* and will be defined in this section. [19]

Tangential continuity

A very important assumption that was made in Theorem (1) is that $\boldsymbol{u} \times \boldsymbol{n}$ is continuous over the domain. If this is not the case, the integral would not be well-defined. One could divide the domain into different sub-domains and carry out the same integration by parts formula and obtain additional terms in the process. From this follows that the tangential component of $\boldsymbol{u}$ should be continuous over the domain. This is a less strict requirement than in the case of $H^1(\Omega)$ conforming elements, where the functions are required to be continuous component-wise.

Reference elements

Like the Lagrangian polynomials, the Nédélec basis functions are defined element-wisely. The most convenient way to do this is to first define them on a certain reference element $\hat{K}$, and then to map these functions onto the elements e^k in the real domain Ω. The reference elements $\hat{K}$ for $d = 2$ and $d = 3$ are shown in Figure 2.12.



(a) Unit triangle. (b) Unit tetrahedron.

Figure 2.12: Reference elements $\hat{K}$ in 2D and 3D. [20]

The coordinates in these reference elements can then be mapped onto coordinates in the real domain by the following transformation:

$$\mathbf{x} = F_{e_k}(\hat{\mathbf{x}}) = B_{e_k}\hat{\mathbf{x}} + b_{e_k}, \tag{2.51}$$

where B_{e_k} is a $d \times d$ matrix and b_{e_k} is a $d \times 1$ vector.

Function spaces

As stated before, the most important requirement on curl-conforming elements is their tangential continuity. To obtain a proper basis, the goal of this section is therefore to find a set of basis functions that are tangentially continuous over the element edges.

In the construction of the Lagrange basis, each basis function was related to a certain node in the mesh. In the case of Nédélec elements, each function will be related to an edge in the mesh. To find these functions, it will be useful to find a set of linearly independent functions that each have a nonzero tangential component on one of the edges. To find general functions that meet this requirement, some additional function spaces are defined:

$$\tilde{\mathbb{P}}_m(\hat{K}) = \left\{ p(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \hat{x}_1^{a_1^j} \hat{x}_2^{a_2^j} \dots \hat{x}_d^{a_d^j} : \alpha \in \mathbb{R}, n \in \mathbb{N}, \sum_{i=1}^{d} a_i^j = m \ \forall j = 1 \dots n \right\}, \tag{2.52}$$

which is the space of homogeneous polynomials of degree m. It has dimension $m+1$ for $d = 2$, so in the case of $m = 1$ an additional auxiliary space is required to in the end find the three independent basis functions related to each edge in the reference element [19]:

$$\mathcal{S}^m(\hat{K}) = \left\{ \mathbf{p} \in \left(\tilde{\mathbb{P}}_m(\hat{K})\right)^d : \mathbf{p} \cdot \hat{\mathbf{x}} = 0, \forall \hat{\mathbf{x}} \in \hat{K} \right\}. \tag{2.53}$$

The dimension of this space is m for $d = 2$ and $m(m+2)$ for $d = 3$. This space is needed because it contains functions that have nonzero tangential component on the edge from $(1,0)$ to $(0,1)$ in the reference element, while it is perpendicular to the other two edges (in the case of $d=2$). The function space with functions that have nonzero tangential components on each of the three edges can then be defined as:

$$\mathcal{R}^m = \left(\mathbb{P}_{m-1}(\hat{K})\right)^d \oplus \mathcal{S}^m. \tag{2.54}$$

This is the space from which the Nédélec basis functions will be chosen. In this report, the discussion will be limited to the case where $d=2$. In this case, the following theorem holds:

Theorem 2. *[19] For $d=2$, $\mathcal{R}_m$ can be written as:*

$$\mathcal{R}^m = \left(\mathbb{P}_{m-1}(\hat{K})\right)^d \oplus \tilde{\mathbb{P}}_{m-1}(\hat{K}) \begin{bmatrix} \hat{x}_2 \\ -\hat{x}_1 \end{bmatrix}. \tag{2.55}$$

Proof. The dimension of the space $\tilde{\mathbb{P}}_{m-1}(\hat{K})$ in 2 variables is m. As stated before, so is the dimension of $\mathcal{S}^m$ for $d=2$. To prove the stated equivalent, all that remains to be demonstrated is:

$$\tilde{\mathbb{P}}_{m-1}(\hat{K}) \begin{bmatrix} \hat{x}_2 \\ -\hat{x}_1 \end{bmatrix} \subseteq \mathcal{S}^m.$$

To prove this, take $p \in \tilde{\mathbb{P}}_{m-1}(\hat{K})$. The following then holds:

$$p \begin{bmatrix} \hat{x}_2 \\ -\hat{x}_1 \end{bmatrix} \cdot \hat{\mathbf{x}} = 0,$$

so $p \begin{bmatrix} \hat{x}_2 \\ -\hat{x}_1 \end{bmatrix} \in \mathcal{S}_m$ and this concludes the proof. □

In this report, linear Nédélec basis functions will be used. This means order $m=1$ is chosen and this results in the following function space:

$$\mathcal{R}^1 = \left\langle \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \hat{x}_2 \\ -\hat{x}_1 \end{bmatrix} \right\rangle. \tag{2.56}$$

Nédélec basis functions of first order
Based on function space $\mathcal{R}^1$, the Nédélec basis functions of first order will now be constructed. This is done by first defining a set of degrees of freedom. These degrees of freedom are related to the edges of the reference element. The numbering and orientation of these edges is shown in Figure 2.13.

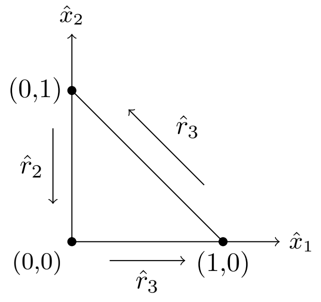


Figure 2.13: Orientation and numbering of edges in the 2D reference element.

The set of degrees of freedom related to these edges for $u \in \mathcal{R}^1$ then becomes:

$$\hat{A} = \left\{ \hat{\alpha}(u) = \int_{\hat{r}_i} \hat{t}_i \cdot u \; d\hat{s}, i \in \{1,2,3\} \right\}, \tag{2.57}$$

where $\hat{t}_i$ is the tangential unit vector of the edge $\hat{r}_i$. Finally, the Nédélec basis functions on the reference element $\hat{\boldsymbol{\eta}}_i$ are defined by:

$$\begin{aligned} &(1)\ \hat{\boldsymbol{\eta}}_i(\mathbf{x}) \in \mathcal{R}^1(\Omega) \\ &(2)\ \hat{\alpha}_i(\hat{\boldsymbol{\eta}}_j) = \delta_{ij}, \end{aligned} \tag{2.58}$$

which results in the following three basis functions on reference element $\hat{K}$:

$$\hat{\boldsymbol{\eta}}_1 = \begin{bmatrix} -\hat{x}_2 \\ \hat{x}_1 \end{bmatrix}, \; \hat{\boldsymbol{\eta}}_2 = \begin{bmatrix} -\hat{x}_2 \\ \hat{x}_1 - 1 \end{bmatrix}, \; \hat{\boldsymbol{\eta}}_3 = \begin{bmatrix} 1 - \hat{x}_2 \\ \hat{x}_1 \end{bmatrix}. \tag{2.59}$$

Piola transformation
To obtain basis functions on a general element e_k, the functions on the reference element have to be mapped onto the real domain Ω. Normally, one would expect this is done by:

$$\boldsymbol{\eta}(\mathbf{x}) = \left(\hat{\boldsymbol{\eta}} \circ F_{e_k}^{-1}\right)(\mathbf{x}), \tag{2.60}$$

but the problem with this approach is that the pull-back of a $H^{curl}(\hat{K})$-function isn't necessarily a part of $H^{curl}(e_k)$. [19] Instead, the following transformation, called a *Piola transformation* is used. It is defined by the following:

$$\boldsymbol{\eta}(\mathbf{x}) = \mathcal{P}_{e_k}(\hat{\boldsymbol{\eta}}) = \left(\hat{D}F_{e_k}^{-T} \hat{\boldsymbol{\eta}}\right) \circ F_{e_k}^{-1}(\mathbf{x}), \tag{2.61}$$

where $\hat{D}F_{e_k}$ is the Jacobian of the element map. Using this definition and the element map from equation (2.51), the following is obtained for the Nédélec basis functions on the element e_k:

$$\boldsymbol{\eta}(\mathbf{x}) = B_{e_k}^{-T} \left(\hat{\boldsymbol{\eta}} \circ F_{e_k}^{-1}\right)(\mathbf{x}). \tag{2.62}$$

The rotation of these functions, which is needed in the weak formulation of the GL-equations, is mapped onto element e_k as follows:

$$\text{curl } \boldsymbol{\eta}(\mathbf{x}) = \frac{1}{\det B_{e_k}} \text{ curl} \left(\hat{\boldsymbol{\eta}} \circ F_{e_k}^{-1} \right) (\mathbf{x}). \tag{2.63}$$

Tangential continuity
The last step in obtaining the Nédélec basis functions of first order that are globally defined on the entire domain Ω, is assuring they satisfy tangential continuity. The basis functions were defined by the tangential unit vectors $\hat{t}_i$ on the reference element $\hat{K}$. In order for the global basis functions to be well defined, the used element mapping needs to assure that the tangential unit vectors elements that share an edge have the same orientation on that edge, which is not necessarily so in case of the Piola mapping discussed in the previous section. To make this right, additional signs need to be added to certain basis functions that will assure tangential continuity on the entire domain. For two neighbouring elements e_k and e_l, that share an edge r_m, the following mapping is used:

$$\boldsymbol{\eta}_m(\mathbf{x}) = \begin{cases} \frac{B_{e_k}}{|\det B_{e_k}|} \left(\hat{D} F_{e_k}^{-T} \hat{\boldsymbol{\eta}}_i \right) \circ F_{e_k}^{-1}(\mathbf{x}), & \text{on } e_k \\ -\frac{B_{e_l}}{|\det B_{e_l}|} \left(\hat{D} F_{e_k}^{-T} \hat{\boldsymbol{\eta}}_j \right) \circ F_{e_l}^{-1}(\mathbf{x}), & \text{on } e_l \\ 0, \quad \text{elsewhere.} \end{cases} \tag{2.64}$$

Here i and j are the number of the edge r_m on the reference element $\hat{K}$ for respectively real elements e_k and e_l.

This defines all linear Nédélec basis functions on the entire domain Ω. This result will later be used for modelling the second Ginzburg-Landau equation in its weak formulation (Equation (2.40)).

2.7.4 Numerical integration

As has been shown in Section 2.7.1, the weak formulation consists of definite integrals over a certain domain. To be able to compute these integrals in MATLAB, numerical *quadrature rules* are required, which approximate these integrals. The general form of such a rule on a domain Ω is shown in Equation (2.65). [5]

$$\int_\Omega f(\mathbf{x}) \, d\Omega \approx \sum_{i=1}^{M} w_i f(\mathbf{x}_i), \tag{2.65}$$

where M is the used number of integration points and w_i are the weights of the integration points $\mathbf{x}_i$. There are many choices for the integration points. A common group is known as the *Gaussian quadrature rules*. These are constructed to yield an exact result for polynomials of a certain degree. [21] The integration points and weights are found by using Equation (2.65) and assuming

f is a general polynomial of a certain degree. More on this process can be read in [22] or [23]. There are plenty of MATLAB programs that provide the set of integration points and weights for a certain order Gaussian quadrature.

2.8 Finite difference method

As stated before, for the time derivatives in the Ginzburg-Landau equations a finite difference method will be used. This means the time derivative of functions will be approximated by evaluating the function on discrete times, using a constant time interval dt. There is a wide range of possible methods to do this, but in this report a simple *backwards difference method* will be used. The time derivative of a function will then be approximated as follows:

$$\frac{\partial f}{\partial t}(x,y,z,t) = \frac{f(x,y,z,t) - f(x,y,z,t-dt)}{dt}. \tag{2.66}$$

3 Finite Element Method for the GL-equations

The goal of this chapter is to develop an accurate numerical method for solving the time-dependent Ginzburg-Landau equations that were presented in the previous chapter. The simulations will be performed in 2D. To this end, the finite element method, that has been introduced in the previous chapter, will be used. Firstly, the program COMSOL Multiphysics[2] will be used to develop this model and some results will be presented. However, it turns out that these results are flawed. The likely cause of these flaws will be discussed and the remainder of this chapter is devoted to obtaining a more accurate model using different basis functions in MATLAB[3].

3.1 COMSOL Multiphysics

COMSOL Multiphysics is a finite element analysis, solver and simulation software package for various physics and engineering applications, especially coupled phenomena, or multiphysics. In addition to conventional physics-based user interfaces, COMSOL Multiphysics also allows entering coupled systems of partial differential equations (PDE's).[24] This functionality was used to model the time-dependent Ginzburg-Landau equations.

3.1.1 GL-equations in COMSOL

COMSOL uses piecewise linear basis functions to approximate all degrees of freedom on the points of a generated mesh. To write the equations in the proper form, both ψ and $\mathbf{A}$ are divided into two components. For ψ these are the real and imaginary parts and for $\mathbf{A}$ these are the x- and y-component. This leaves four independent variables $u_{1\ldots4}$. The following substitutions are made [1]:

$$\psi = u_1 + u_2 i, \quad \mathbf{A} = \begin{bmatrix} u_3 \\ u_4 \end{bmatrix}. \tag{3.1}$$

Substituting this into Equation (2.34), the following is obtained:

$$\frac{d(u_1+u_2 i)}{dt} = -\left(\frac{i}{\kappa}\nabla + \mathbf{A}\right)^2 (u_1+u_2 i) + \psi - |u_1+u_2 i|^2 (u_1+u_2 i). \tag{3.2}$$

Simplifying and splitting the final equation into the real and imaginary part results in:

$$\frac{du_1}{dt} = \frac{\Delta u_1}{\kappa^2} + \frac{2}{\kappa}(u_3 u_{2,x} + u_4 u_{2,y}) + \frac{1}{\kappa}(u_{3,x} + u_{4,y})u_2 - (u_3^2+u_4^2)u_1 - u_1 - (u_1^2+u_2^2)u_1, \tag{3.3}$$

$$\frac{du_2}{dt} = \frac{\Delta u_2}{\kappa^2} - \frac{2}{\kappa}(u_3 u_{1,x} + u_4 u_{1,y}) - \frac{1}{\kappa}(u_{3,x} + u_{4,y})u_1 - (u_3^2 + u_4^2)u_2 - u_2 - (u_1^2 + u_2^2)u_2. \tag{3.4}$$

To implement the second equation along with the boundary conditions a fifth auxiliary variable u_5 is introduced, satisfying the equation:

$$\nabla \cdot \begin{bmatrix} u_3 \\ u_4 \end{bmatrix} = u_{3,x} + u_{4,y} + u_5. \tag{3.5}$$

To implement the boundary conditions, in the second equation, a term $\nabla \times \mathbf{B}_a$ is added where $\mathbf{B}_a = (0, 0, B_a)$.

$$\sigma \frac{du_3}{dt} = \frac{1}{\kappa}(u_1 u_{2,x} - u_2 u_{1,x}) - (u_1^2 + u_2^2)u_3 + \frac{d}{dx}(u_{4,x} - u_{3,y}) - \mathbf{B}_a, \tag{3.6}$$

$$\sigma \frac{du_4}{dt} = \frac{1}{\kappa}(u_1 u_{2,x} - u_2 u_{1,x}) - (u_1^2 + u_2^2)u_4 + \frac{d}{dy}(-u_{4,x} + u_{3,y}) + \mathbf{B}_a. \tag{3.7}$$

The general form of a PDE in COMSOL is shown below. A Neumann boundary condition is used, which is also given below.

$$\mathbf{e}_a \frac{d^2\mathbf{u}}{dt^2} + \mathbf{d}_a \frac{d\mathbf{u}}{dt} + \nabla \cdot \mathbf{\Gamma} = \mathbf{F}. \tag{3.8}$$

In Equation (3.6), the matrix $\mathbf{e}_a$ contains only zeros. The remaining matrices and vectors are inserted as:

$$-\mathbf{n} \cdot \mathbf{\Gamma} = \mathbf{G}, \quad \text{on } d\Omega. \tag{3.9}$$

$$\mathbf{d}_a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \sigma & 0 & 0 \\ 0 & 0 & 0 & \sigma & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.10}$$

$$\mathbf{\Gamma} = \begin{bmatrix} [-u_{1,x}/\kappa^2, -u_{1,y}/\kappa^2]^T \\ [-u_{2,x}/\kappa^2, -u_{2,y}/\kappa^2]^T \\ [0, u_{4,x} - u_{3,y} - B_a]^T \\ [-u_{4,x} + u_{3,y} + B_a, 0]^T \\ [u_3, u_4]^T \end{bmatrix}, \tag{3.11}$$

$$\mathbf{F} = \begin{bmatrix} 2(u_3 u_{2,x} + u_4 u_{2,y})/\kappa + (u_{3,x} + u_{4,y})u_2/\kappa - (u_3^2 + u_4^2)u_1 + u_1 - (u_1^2 + u_2^2)u_1 \\ -2(u_3 u_{1,x} + u_4 u_{1,y})/\kappa - (u_{3,x} + u_{4,y})u_1/\kappa - (u_3^2 + u_4^2)u_2 + u_2 - (u_1^2 + u_2^2)u_2 \\ (u_1 u_{2,x} - u_2 u_{1,x})/\kappa - (u_1^2 + u_2^2)u_3 \\ (u_1 u_{2,y} - u_2 u_{1,y})/\kappa - (u_1^2 + u_2^2)u_4 \\ u_{3,x} + u_{4,y} + u_5 \end{bmatrix}. \tag{3.12}$$

3.1.2 Numerical simulations

Since the focus in this report is on type II superconductors, this type will also be used in the simulations. Therefore, a $\kappa > 1/\sqrt{2}$ is chosen. Furthermore, to investigate the magnetic field penetration on the domain properly, the applied magnetic field has to be near the critical field H_c. σ is chosen to be 1 and it turned out that a dt of 0.1 s was required for convergence of the results. In summary:

- $\kappa = 4$
- $B_a = 1$
- $\sigma = 1$
- $dt = 0.1$

The equations were solved on the geometry and mesh of Figure 3.1. The dimensions were chosen such that Abrikosov vortices should arise on the domain. An indentation is added to see what effect this has on the field penetration.

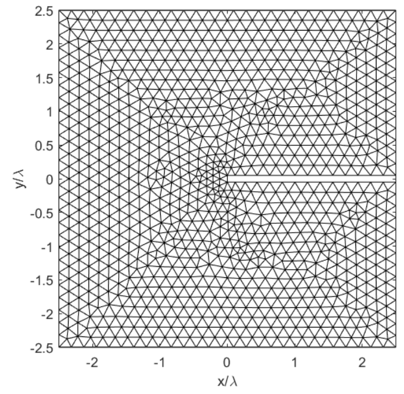


Figure 3.1: Mesh on a 2 by 2 square with an indentation.

The results are shown in Figures 3.2 and 3.3. The modulus of the order parameter and the magnetic field in the z-direction are plotted for certain points in time.

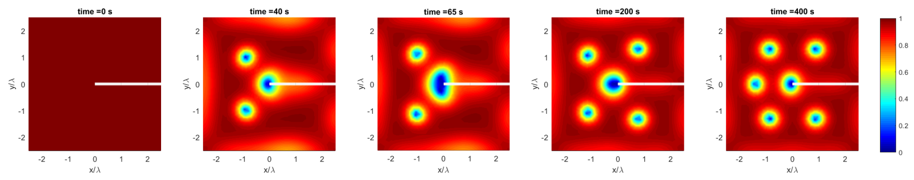


Figure 3.2: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of the GL equations using COMSOL.

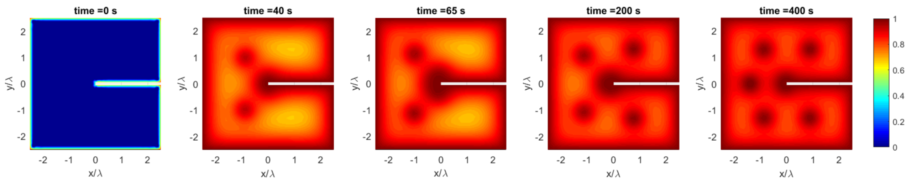


Figure 3.3: Magnetic field in the z-direction as results from the time-dependent solution of the GL equations using COMSOL.

Discussion of the results

As it can be seen, at first $|\psi|$ equals 1 on the entire domain, which corresponds to the superconducting state. As time progresses, the applied magnetic field starts to intrude the sample and vortices arise where the order parameter is close to zero. These vortices originate at the end of the indentation, where they grow until they enter and find another spot on the domain. Finally no more vortices arise and the vortices are slightly relocated and finally the system is at a steady-state, which corresponds to the last plot in Figure 3.2.

Most of these results agree with measurements and predictions. However, there are some flaws. The most important one is the vortex entry. As it can be seen in the third plot, it takes a relatively large vortex to enter the material. This behaviour is nonphysical and it turns out that for more complex geometries, problems become even larger. [25]

A possible solution for this problem would be to refine the mesh, but it turns out that the *mesh convergence* for this particular problem is very slow, indicating that for accurate results an unrealistically high resolution has to be used. A different, and better approach is to use a different set of basis functions, which are curl-conforming. An example of these elements are Nédélec elements, which have been discussed in Section 2.7.3 and in the remainder of this chapter such a model will be constructed using MATLAB.

3.2 FEM implementation in MATLAB

In the previous4 chapter, the time-dependent Ginzburg-Landau equations were expressed in their weak formulations(Equations (2.39) and (2.40)). The goal of this section is to finalize this groundwork and develop a finite element model for the problem using MATLAB[3].

3.2.1 Discretization

As stated in Section 2.7.3, the final step in constructing a finite element model for the problem is inserting the approximations for $\mathbf{A}$ and ψ and then insert the basis functions one by one as test functions to create a system of equations.

First GL-equation
Since the equations are coupled, in solving each of the equations some assumptions need to be made. In the first equation, $\mathbf{A}$ and $|\psi|^2$ are assumed to be known. They will be approximated by an iterative process. The basis that is used for ψ is the Lagrangian linear basis, which has been discussed at length in Section 2.7.3. The approximation then becomes:

$$\psi(\mathbf{x}) \approx \sum_{i=1}^{N_p} \psi_i(t) l_i(\mathbf{x}), \tag{3.13}$$

where we recall N_p is the number of points in the mesh, $l_i(\mathbf{x})$ is the Lagrangian basis function related to the point $\mathbf{x}_i$ and $\psi_i(t)$ is the time-dependent coefficient of that basis function. By recalling Definition (2.47), it is noted that this is also the approximation value of $\psi(\mathbf{x}_i)$. Inserting this into Equation (2.39) and replacing the test function by $l_j(\mathbf{x})$ for $j = 1, 2, \ldots, N_p$ results in:

$$\int_\Omega \frac{d\left(\sum_{i=1}^{N_p} \psi_i(t) l_i(\mathbf{x})\right)}{dt}\, l_j(\mathbf{x})\; d\Omega = -\frac{1}{\kappa^2}\int_\Omega \nabla l_j(\mathbf{x}) \cdot \nabla\left(\sum_{i=1}^{N_p} \psi_i(t) l_i(\mathbf{x})\right)\, d\Omega - \int_\Omega \frac{i}{\kappa}(\nabla\cdot\mathbf{A})\left(\sum_{i=1}^{N_p} \psi_i(t) l_i(\mathbf{x})\right) l_j(\mathbf{x})\; d\Omega$$
$$- \int_\Omega \frac{2i}{\kappa}\mathbf{A}\cdot\nabla\left(\sum_{i=1}^{N_p} \psi_i(t) l_i(\mathbf{x})\right) l_j(\mathbf{x})\; d\Omega + \int_\Omega \left(1 - \mathbf{A}^2 - |\psi|^2\right)\left(\sum_{i=1}^{N_p} \psi_i(t) l_i(\mathbf{x})\right) l_j(\mathbf{x})\; d\Omega.$$

Simplifying:

$$\frac{d\sum_{i=1}^{N_p} \psi_i(t)}{dt}\int_\Omega l_i(\mathbf{x}) l_j(\mathbf{x})\; d\Omega = -\frac{1}{\kappa^2}\sum_{i=1}^{N_p}\psi_i(t)\int_\Omega \nabla l_j(\mathbf{x})\cdot\nabla l_i(\mathbf{x})\; d\Omega - \frac{i}{\kappa}\sum_{i=1}^{N_p}\psi_i(t)\int_\Omega (\nabla\cdot\mathbf{A})\, l_i(\mathbf{x}) l_j(\mathbf{x})\; d\Omega$$
$$-\frac{2i}{\kappa}\sum_{i=1}^{N_p}\psi_i(t)\int_\Omega (\mathbf{A}\cdot\nabla l_i(\mathbf{x}))\, l_j(\mathbf{x})\; d\Omega + \sum_{i=1}^{N_p}\psi_i(t)\int_\Omega \left(1 - \mathbf{A}^2 - |\psi|^2\right) l_i(\mathbf{x}) l_j(\mathbf{x})\; d\Omega.$$

Finally, a backwards finite difference method in the time-domain is used to obtain:

$$\frac{\sum_{i=1}^{N_p} \psi_i(t) - \sum_{i=1}^{N_p} \psi_i(t-dt)}{dt} \int_\Omega l_i(\mathbf{x}) l_j(\mathbf{x})\, d\Omega$$
$$= -\frac{1}{\kappa^2} \sum_{i=1}^{N_p} \psi_i(t) \int_\Omega \nabla l_j(\mathbf{x}) \cdot \nabla l_i(\mathbf{x})\, d\Omega - \frac{i}{\kappa} \sum_{i=1}^{N_p} \psi_i(t) \int_\Omega (\nabla \cdot \mathbf{A})\, l_i(\mathbf{x}) l_j(\mathbf{x})\, d\Omega$$
$$- \frac{2i}{\kappa} \sum_{i=1}^{N_p} \psi_i(t) \int_\Omega (\mathbf{A} \cdot \nabla l_i(\mathbf{x}))\, l_j(\mathbf{x})\, d\Omega + \sum_{i=1}^{N_p} \psi_i(t) \int_\Omega \left(1 - \mathbf{A}^2 - |\psi|^2\right) l_i(\mathbf{x}) l_j(\mathbf{x})\, d\Omega.$$

Simplifying:

$$\sum_{i=1}^{N_p} \left(\int_\Omega l_i(\mathbf{x}) l_j(\mathbf{x})\, d\Omega + \left(\frac{1}{\kappa^2} \int_\Omega \nabla l_j(\mathbf{x}) \cdot \nabla l_i(\mathbf{x})\, d\Omega + \frac{i}{\kappa} \int_\Omega (\nabla \cdot \mathbf{A})\, l_i(\mathbf{x}) l_j(\mathbf{x})\, d\Omega + \frac{2i}{\kappa} \int_\Omega (\mathbf{A} \cdot \nabla l_i(\mathbf{x}))\, l_j(\mathbf{x})\, d\Omega \right.\right.$$
$$\left.\left. - \int_\Omega \left(1 - \mathbf{A}^2 - |\psi|^2\right) l_i(\mathbf{x}) l_j(\mathbf{x})\, d\Omega \right) dt \right) \psi_i(t) = \sum_{i=1}^{N_p} \left(\int_\Omega l_i(\mathbf{x}) l_j(\mathbf{x})\, d\Omega \right) \psi_i(t - dt). \tag{3.14}$$

This is a system of N_p equations with N_p unknowns $\psi_i(t)$ for each t, which can be solved in MATLAB by using an iterative process and defining an initial values $\psi_i(t = 0)$. As stated before, in this process $\mathbf{A}$ and $|\psi|^2$ are assumed to be known and are taken as their values from the previous iteration step.

Second GL-equation

In solving the second equation, ψ and $|\psi|^2$ are assumed to be known and will be approximated in the same way as before. The solution for $\mathbf{A}$ of the weak form of the second Equation (2.40) exists in the $H^{curl}(\Omega)$ space and therefore $\mathbf{A}$ is approximated by using the Nédélec basis discussed in Section 2.7.3. The approximation then becomes:

$$\mathbf{A}(\mathbf{x}) \approx \sum_{i=1}^{N_e} A_i(t) \boldsymbol{\eta}_i(\mathbf{x}), \tag{3.15}$$

where N_e is the number of edges in the mesh, $\boldsymbol{\eta}_i(\mathbf{x})$ is the Nédélec basis function related to the edge r_i and $A_i(t)$ is the time-dependent coefficient of that basis function. Now the same process is carried out, the approximation is substituted into Equation (2.40), along with $\boldsymbol{\eta}_j(\mathbf{x})$ for $j = 1, 2, \ldots, N_e$ as test functions. This results in:

$$\sigma \int_\Omega \frac{d\left(\sum_{i=1}^{N_e} A_i(t) \boldsymbol{\eta}_i(\mathbf{x})\right)}{dt} \cdot \boldsymbol{\eta}_j(\mathbf{x})\, d\Omega = \frac{1}{2ik} \int_\Omega (\psi^* \nabla\psi - \psi \nabla\psi^*) \cdot \boldsymbol{\eta}_j(\mathbf{x})\, d\Omega - \int_\Omega |\psi|^2 \left(\sum_{i=1}^{N_e} A_i(t) \boldsymbol{\eta}_i(\mathbf{x}) \right) \cdot \boldsymbol{\eta}_j(\mathbf{x})\, d\Omega$$
$$- \int_\Omega \left(\nabla \times \left(\sum_{i=1}^{N_e} A_i(t) \boldsymbol{\eta}_i(\mathbf{x}) \right) \right) \cdot (\nabla \times \boldsymbol{\eta}_j(\mathbf{x}))\, d\Omega + \oint_{d\Omega} (\mathbf{B}_a \times \boldsymbol{n}) \cdot \boldsymbol{\eta}_j(\mathbf{x})\, d(d\Omega).$$

Again applying a backwards finite difference method in the time domain delivers:

$$\sigma \int_\Omega \frac{\left(\sum_{i=1}^{N_e} A_i(t)\boldsymbol{\eta}_i(\mathbf{x}) - \sum_{i=1}^{N_e} A_i(t-dt)\boldsymbol{\eta}_i(\mathbf{x})\right)}{dt} \cdot \boldsymbol{\eta}_j(\mathbf{x})\ d\Omega = \frac{1}{2ik}\int_\Omega (\psi^*\ \nabla\psi - \psi\ \nabla\psi^*)\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\Omega$$
$$-\sum_{i=1}^{N_e} A_i(t)\int_\Omega |\psi|^2\,\boldsymbol{\eta}_i(\mathbf{x})\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\Omega - \sum_{i=1}^{N_e} A_i(t)\int_\Omega (\nabla\times\boldsymbol{\eta}_i(\mathbf{x}))\cdot(\nabla\times\boldsymbol{\eta}_j(\mathbf{x}))\,d\Omega + \oint_{d\Omega}(\mathbf{B}_a\times\boldsymbol{n})\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\,(d\Omega)\,.$$

Simplifying:

$$\sum_{i=1}^{N_e}\left(\sigma\int_\Omega \boldsymbol{\eta}_i(\mathbf{x})\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\Omega + \left(\int_\Omega |\psi|^2\,\boldsymbol{\eta}_i(\mathbf{x})\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\Omega + \int_\Omega (\nabla\times\boldsymbol{\eta}_i(\mathbf{x}))\cdot(\nabla\times\boldsymbol{\eta}_j(\mathbf{x}))\,d\Omega\right)dt\right)A_i(t)$$
$$= \sigma\sum_{i=1}^{N_e} A_i(t-dt)\int_\Omega \boldsymbol{\eta}_i(\mathbf{x})\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\Omega + \left(\frac{1}{2ik}\int_\Omega (\psi^*\ \nabla\psi - \psi\ \nabla\psi^*)\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\Omega + \oint_{d\Omega}(\mathbf{B}_a\times\boldsymbol{n})\cdot\boldsymbol{\eta}_j(\mathbf{x})\ d\,(d\Omega)\right)dt. \tag{3.16}$$

This is a system of N_e equations with N_e unknowns $A_i(t)$ for each t which can be solved in MATLAB by an iterative process by defining initial values $A_i(t=0)$. The values for ψ and $|\psi|^2$ are taken from the solution of the first equation.

3.2.2 Matrix and vector assembly

The last section provides a set of two systems of equations for each t. To be able to solve these systems efficiently in MATLAB, they will now be presented in matrix-vector notation.

Matrix-vector notation

The systems of equation given in Equations (3.14) and (3.16) can respectively be written in matrix-vector notation as:

$$(M + (P+S)\ dt)\boldsymbol{\psi}^t = M\boldsymbol{\psi}^{t-dt}, \tag{3.17}$$

with:

- M a $(N_p \times N_p)$ matrix with elements $M_{ji} = \int_\Omega l_i(\mathbf{x})l_j(\mathbf{x})\ d\Omega$
- P a $(N_p \times N_p)$ matrix with elements $P_{ji} = \int_\Omega \left(\frac{i}{\kappa}\nabla\cdot\mathbf{A} - 1 + \mathbf{A}^2 + |\psi|^2\right) l_i(\mathbf{x})l_j(\mathbf{x})\ d\Omega$
- S a $(N_p \times N_p)$ matrix with elements $S_{ji} = \frac{1}{\kappa^2}\int_\Omega \nabla l_i(\mathbf{x})\cdot\nabla l_j(\mathbf{x})\ d\Omega + \frac{2i}{\kappa}\int_\Omega (\mathbf{A}\cdot\nabla l_i(\mathbf{x}))\, l_j(\mathbf{x})\ d\Omega$
- $\boldsymbol{\psi}^t$ a $(N_p \times 1)$ vector with elements $\boldsymbol{\psi}_i^t = \psi_i(t)$.

And:

$$(\sigma M^{Ned} + (T+Q)\ dt)\boldsymbol{A}^t = \sigma M^{Ned}\boldsymbol{A}^{t-dt} + (\boldsymbol{f}+\boldsymbol{b})\ dt, \tag{3.18}$$

with:

- M^{Ned} a $(N_e \times N_e)$ matrix with elements $M^{Ned}_{ji} = \int_\Omega \boldsymbol{\eta}_i(\mathbf{x}) \cdot \boldsymbol{\eta}_j(\mathbf{x})\ d\Omega$
- T a $(N_e \times N_e)$ matrix with elements $T_{ji} = \int_\Omega |\psi|^2\, \boldsymbol{\eta}_i(\mathbf{x}) \cdot \boldsymbol{\eta}_j(\mathbf{x})\ d\Omega$
- Q a $(N_e \times N_e)$ matrix with elements $Q_{ji} = \int_\Omega (\nabla \times \boldsymbol{\eta}_i(\mathbf{x})) \cdot (\nabla \times \boldsymbol{\eta}_j(\mathbf{x}))\, d\Omega$
- $\boldsymbol{f}$ a $(N_e \times 1)$ vector with elements $\boldsymbol{f}_i = \frac{1}{2ik} \int_\Omega (\psi^*\ \nabla\psi - \psi\ \nabla\psi^*) \cdot \boldsymbol{\eta}_i(\mathbf{x})\ d\Omega$
- $\boldsymbol{b}$ a $(N_e \times 1)$ vector with elements $\boldsymbol{b}_i = \oint_{d\Omega} (\mathbf{B}_a \times \boldsymbol{n}) \cdot \boldsymbol{\eta}_i(\mathbf{x})\ d\,(d\Omega)$
- $\boldsymbol{A}^t$ a $(N_e \times 1)$ vector with elements $\boldsymbol{A}^t_i = A_i(t)$.

Element matrices and vectors
Since the basis functions are defined element-wisely, the integrals in Equations (3.17) and (3.18) will have to be computed per element:

$$\int_\Omega f(\mathbf{x})\ d\Omega = \sum_{k=1}^{N} \int_{e_k} f(\mathbf{x}) d\Omega. \tag{3.19}$$

This appears to be a cumbersome procedure, but in fact only some of the integrals in Equation (3.19) are different from zero. The reason for this is that they all contain the basis functions, which are nonzero only on certain elements of the mesh. In 2D, only three of the basis functions are nonzero on each element.

Because of this, an efficient way of assembling the required matrices and vectors is done by using element matrices and vectors. The matrix, respectively vector entries that need to be computed are in general given by:

$$H_{ij} = \int_\Omega h(\mathbf{x}, \boldsymbol{\phi}_i(\mathbf{x}), \boldsymbol{\phi}_j(\mathbf{x}))\ d\Omega, \tag{3.20}$$

$$\boldsymbol{g}_i = \int_\Omega g(\mathbf{x}, \boldsymbol{\phi}_i(\mathbf{x}))\ d\Omega. \tag{3.21}$$

for some functions g and h.

Here $\boldsymbol{\phi}$ is a general basis function. Equation (3.19) is applied for both cases. On each element e_k, for the vector and matrix case only three and nine integrals are nonzero respectively. This gives raise to the concepts *element matrix* and *element vector*. The element matrix in the 2D case is a (3×3) matrix H^{e_k} and the element vector is a (3×1) vector $\boldsymbol{g}^{e_k}$. Their elements are defined by:

$$H^{e_k}_{ij} = \int_{e_k} h(\mathbf{x}, \boldsymbol{\phi}_i(\mathbf{x}), \boldsymbol{\phi}_j(\mathbf{x}))\ d\Omega, \tag{3.22}$$

$$\boldsymbol{g}^{e_k}_i = \int_{e_k} g(\mathbf{x}, \boldsymbol{\phi}_i(\mathbf{x}))\ d\Omega. \tag{3.23}$$

Finally, the elements of these element matrices and vectors are added to their corresponding element in the large vectors and matrices.

Boundary vector
There is one exception to the general forms stated in Equations (3.20) and (3.21), which is the *boundary vector* $\boldsymbol{b}$. The form of this integral is similar, only now the domain of the integral is $d\Omega$. A similar integration method is used, only now the domain is divided into the K edges that are on the boundary. This *set of boundary edges* will be denoted by $\mathcal{B}(\Omega)$. For 2D simulations, this becomes a line integral along the boundary:

$$\boldsymbol{b}_i = \oint_{d\Omega} (\mathbf{B}_a \times \boldsymbol{n}) \cdot \boldsymbol{\eta}_i(\mathbf{x})\ d\,(d\Omega) = \sum_{r_k \in \mathcal{B}(\Omega)} \int_{r_k} (\mathbf{B}_a \times \boldsymbol{n}) \cdot \boldsymbol{\eta}_i(\mathbf{x})\ d\,(d\Omega)\,. \quad (3.24)$$

Again, only some of the integrals in Equation (3.24) are different from zero. In fact, in this case only one is. This is due to the definition of the Nédélec basis functions, which can be found in Equation (2.58). It is noted that since $\mathbf{B}_a$ is in the z-direction, $\mathbf{B}_a \times \boldsymbol{n}$ will be along the edge, i.e.:

$$\mathbf{B}_a \times \boldsymbol{n} = |\mathbf{B}_a| \boldsymbol{t}. \quad (3.25)$$

Now, using Definition (2.58), this means that on edge r_k, the integral is only nonzero for basis function $\boldsymbol{\eta}_k$. So in this case an element vector is not required and the elements of the boundary vector are simply given by:

$$\boldsymbol{b}_i = \begin{cases} \int_{r_i} (\mathbf{B}_a \times \boldsymbol{n}) \cdot \boldsymbol{\eta}_i(\mathbf{x})\ d\,(d\Omega)\,, & r_i \in \mathcal{B}(\Omega) \\ 0, \quad \text{otherwise.} \end{cases} \quad (3.26)$$

Numerical integration
To compute all of the boundary vector elements, numerical integration is used. The Gaussian integration quadrature, that was discussed in Section 2.7.4, of order 5 is used. The integration points on the reference element for 1D and 2D integrals are shown in Figure 3.4.

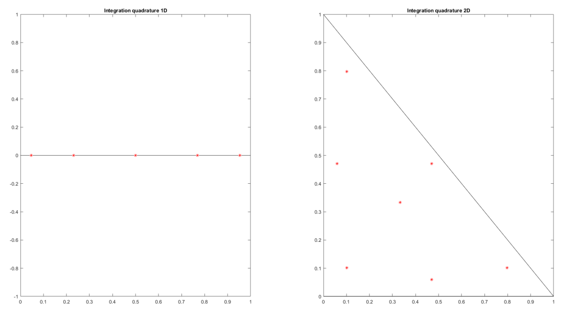


Figure 3.4: Integration quadratures on the reference element for 1D(left) and 2D(right) integrals. [3]

As it has been discussed in the previous chapter, a weight is assigned to the integrand values at each of these points and the results are then summed over all points. The weights for the points above are:

x_i	w_i
0.05	0.12
0.23	0.24
0.50	0.28
0.77	0.24
0.95	0.12

Table 1: 1D quadrature

x_i	y_i	w_i
0.33	0.33	0.11
0.47	0.06	0.07
0.10	0.80	0.06
0.06	0.47	0.07
0.80	0.10	0.06
0.47	0.47	0.07
0.10	0.10	0.06

Table 2: 2D quadrature

The integrals are then computed by using Equation (2.65).

4 Numerical Simulations

In this chapter, numerical simulations using the MATLAB model described in the last chapter will be presented. Since the coding of the entire problem is very sensitive to making small mistakes, the problem is divided into sub-problems. Each of these sub-problems was solved in both COMSOL and MATLAB and the results of these simulations will be compared and discussed in this chapter. The full coupled problem has not been solved in MATLAB yet, but large parts of the separate equations have been numerically solved with success. Therefore, the aim of this chapter is to show what has been achieved so far and give insight into what parts of the code still need improvement.

4.1 First GL equation

The first GL-equation will first be solved. Because of the reasons discussed above, this is done term by term. The same parameters are used as before, so $\kappa = 4$, $\sigma = 1$ and $dt = 0.1$. As an initial value for ψ, the following function is used:

$$\psi_{init} = \frac{1+i}{\sqrt{2}} e^{-(x^2+y^2)} \; \forall \; x, y \in \Omega. \tag{4.1}$$

The sub-problems of this section are all solved on the same mesh, which is again a square. Only now the indentation is removed because this equation solves for the order parameter ψ, and therefore there is no concern for field penetration in this case. The indentation will return in the simulation of the second equation.

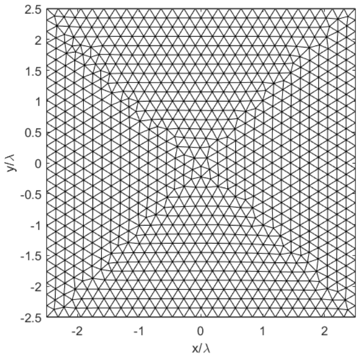


Figure 4.1: Mesh used for the numerical simulations of the first GL equation.

4.1.1 Term-wise solutions

The first problem that was solved is the heat equation, shown in Equation (4.2).

$$\frac{\partial \psi}{\partial t} = \frac{1}{\kappa^2} \Delta \psi. \tag{4.2}$$

The resulting modulus of the order parameter on the domain for five different times are shown in Figures 4.2 and 4.3 for MATLAB and COMSOL respectively.

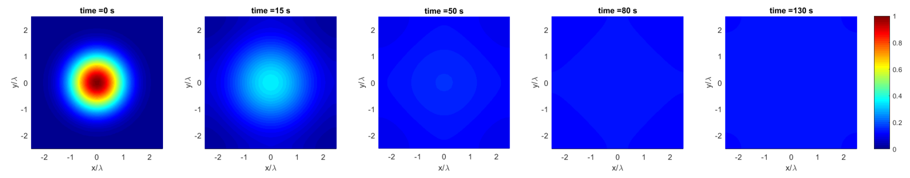


Figure 4.2: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.2) using MATLAB.

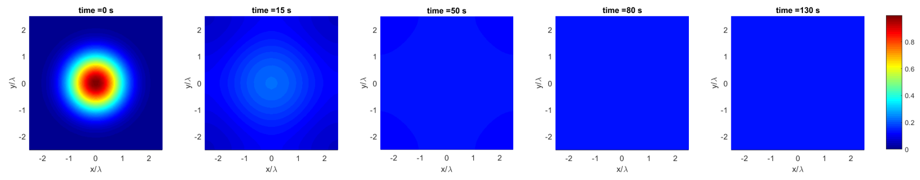


Figure 4.3: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.2) using COMSOL.

As it can be seen, the results are very similar. Over time, $|\psi|$ is spread out over the domain until it is constant. This is the steady-state solution on the right. Small differences are observed on the intermediate time steps. The solution of COMSOL seems to reach the steady-state slightly faster. This could be explained by differences in the finite difference method in the time domain between COMSOL and MATLAB. However, the observed behavior is the same so this is not investigated further.

Next, the following two terms are added:

$$\frac{\partial \psi}{\partial t} = \frac{1}{\kappa^2} \Delta \psi + \psi - |\psi|^2 \psi. \tag{4.3}$$

The results are shown in the same way in Figures 4.4 and 4.5.

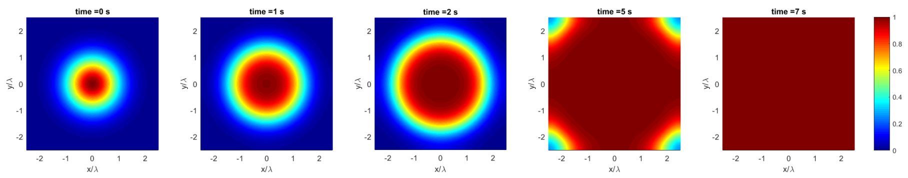


Figure 4.4: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.3) using MATLAB.

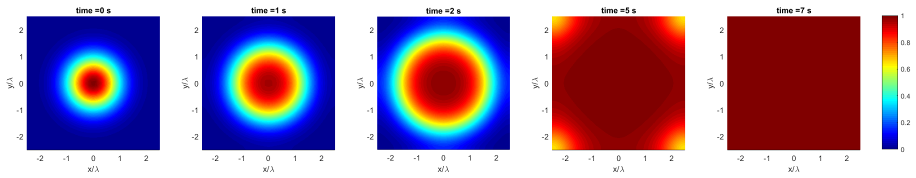


Figure 4.5: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.3) using COMSOL.

The results are again in agreement. The modulus of the order parameter now increases over time and is again spread out over the domain until it is constant. As it can be seen, the steady-state solution is $|\psi|= 1$ everywhere. To see why this result is in agreement with expectation, assume a constant ψ over the domain with a modulus below 1. $\partial\psi/\partial t$ is then positive so the solution is not steady state and $|\psi|$ will increase. The same argument leads to a decrease for $|\psi|> 1$ so the final steady-state solution should indeed be a constant modulus of 1 everywhere. Again, the COMSOL result seems to reach steady-state slightly faster.

Next, the quadratic term involving the magnetic vector potential is added to the equation. In this part of the process, $\mathbf{A}$ is a given function on the domain:

$$\mathbf{A} = [-y, x]^T \ \ \forall\ x, y \in \Omega. \tag{4.4}$$

The equation that will be solved is:

$$\frac{\partial f}{\partial t} = \frac{1}{\kappa^2}\Delta\psi + \psi - |\psi|^2\,\psi - \mathbf{A}^2\psi. \tag{4.5}$$

It turned out that to properly evaluate the time dependent behavior of the solution, a different initial value for ψ should be used. It is given below.

$$\psi_{init} = \frac{1+i}{\sqrt{2}}e^{-10(x^2+y^2)}\ \forall\ x, y \in \Omega. \tag{4.6}$$

With the constant potential given above the time dependent result of solving these equations are shown in Figures 4.6 and 4.7.

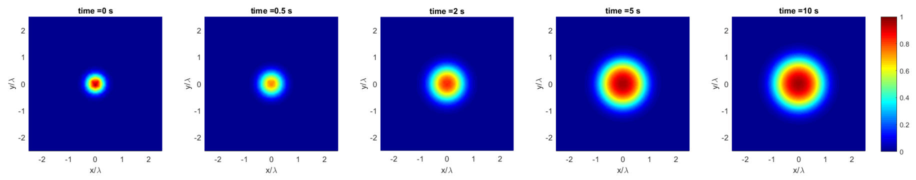


Figure 4.6: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.5) using MATLAB.

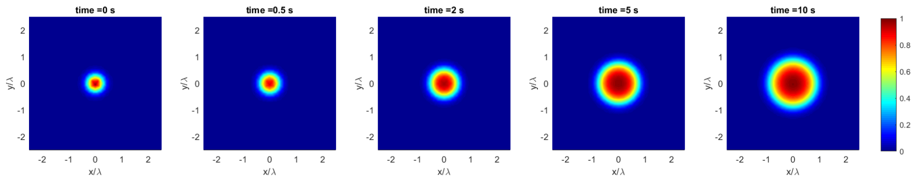


Figure 4.7: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.5) using COMSOL.

Again, similar behavior is observed in both solutions. The steady-state solution is now non-constant over the domain. This is to be expected since the added term favors the decrease of $|\psi|$ and has a magnitude that squares with the distance to the origin. In the origin, this extra term is zero so the solution again approaches $|\psi|= 1$. Far away from the origin, this term is relatively large and so the solution approaches $|\psi|= 0$. The steady-state is the same for both solutions, but again this state is approached faster by COMSOL.

Next, one of the complex cross terms of the equation is also added. The values for $\mathbf{A}$ and the initial value of ψ are the same as in the previous step.

$$\frac{\partial \psi}{\partial t} = \frac{1}{\kappa^2}\Delta\psi + \psi - |\psi|^2\,\psi - \mathbf{A}^2\psi - \frac{i}{\kappa}\mathbf{A}\cdot\nabla\psi. \tag{4.7}$$

The results are:

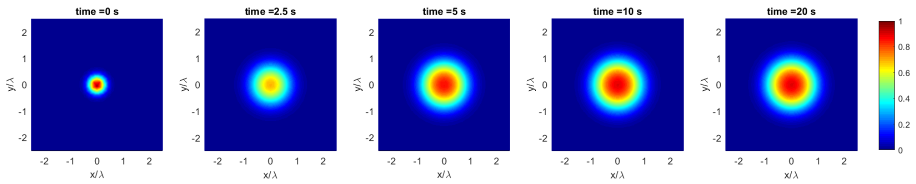


Figure 4.8: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.7) using MATLAB.

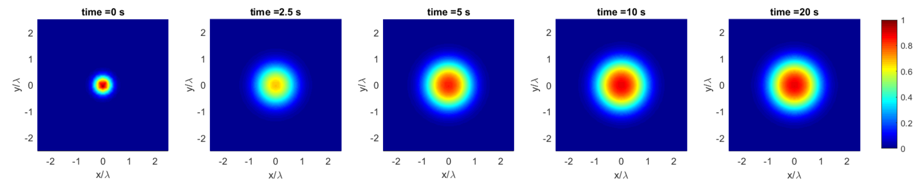


Figure 4.9: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.7) using COMSOL.

The results are similar to the previous ones, and again MATLAB and COMSOL agree.

Finally, the other complex term is added. Notice the used $\mathbf{A}$ is divergence free so the last term is negligible.

$$\frac{\partial\psi}{\partial t} = \frac{1}{\kappa^2}\Delta\psi + \psi - |\psi|^2\psi - \mathbf{A}^2\psi - \frac{2i}{\kappa}\mathbf{A}\cdot\nabla\psi - \frac{i}{\kappa}\psi\nabla\cdot\mathbf{A}. \tag{4.8}$$

In presenting the results, larger time intervals are now used to obtain:

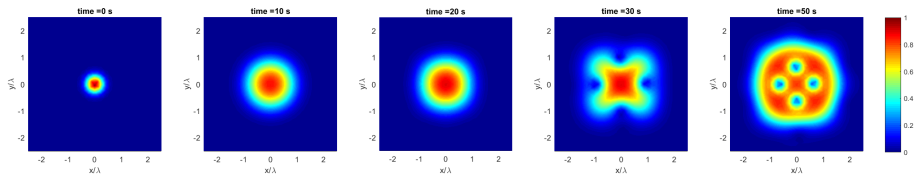


Figure 4.10: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.8) using MATLAB.

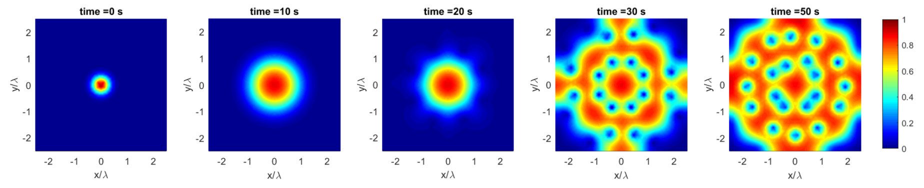


Figure 4.11: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of Equation (4.8) using COMSOL.

In this final step, differences are observed between the solutions. Until a time of about 20 s the solutions agree, but after that vortices of low $|\psi|$ start to arise on the domain. The size of these vortices is the same for both solutions, but in the solution of COMSOL, the amount of vortices is higher. The steady-states of these solutions now also differ. In the MATLAB solution the final amount of vortices is four, while in COMSOL it is twenty. Furthermore, the solution of

COMSOL is not symmetrical while the one of MATLAB is. The cause of these differences is still unclear and should be investigated further.

4.1.2 Additional tests

To gain a better insight to the behavior of the different solvers and to find possible mistakes, some additional tests were performed on the solutions of the entire first GL equation. The results of these tests are discussed in this section.

As it has been discussed, when $\mathbf{A} = \mathbf{0}$ on the domain, the steady-state solution should be $|\psi|= 1$ everywhere on the domain. To check this, the first GL equation is solved with zero magnetic vector potential and a perturbed initial ψ, given by:

$$\psi_{init} = \frac{1+i}{\sqrt{2}} \left(1 + e^{-(x^2+y^2)}\right) \ \forall \ x, y \in \Omega. \tag{4.9}$$

The used mesh and parameters are the same as before. The results are shown in Figures 4.12 and 4.13.

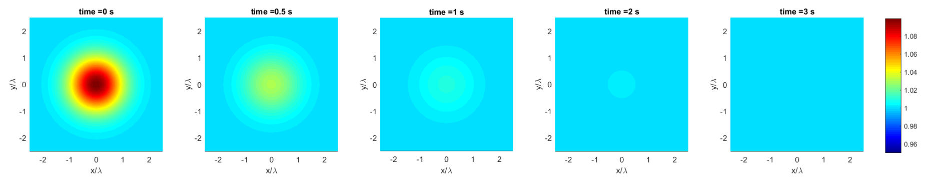


Figure 4.12: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of the first GL equation with zero magnetic vector potential and a perturbed initial ψ using COMSOL.

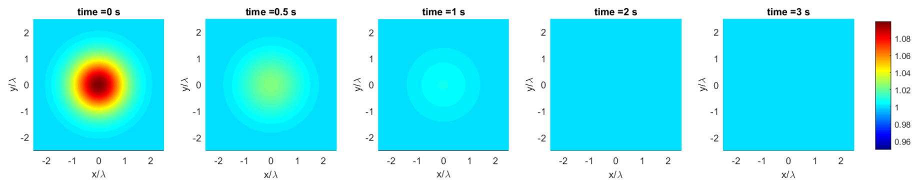


Figure 4.13: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of the first GL equation with zero magnetic vector potential and a perturbed initial ψ using MATLAB.

As it can be seen, the results are similar and as expected: the modulus of the order parameter decreases and reaches its steady-state when it is 1 on the entire domain.

As it has been seen in the previous section, differences between the COMSOL and MATLAB results only arise when the complex terms involving the magnetic vector potential are added. Therefore, more extensive tests have been performed on these differences.

Firstly, the behavior of the solvers for a type I superconductor, i.e. $\kappa < 1$, is checked. With the same given magnetic vector potential as before and a constant initial ψ, this should result in a steady-state that has $|\psi| = 0$ everywhere. This means the field penetrates the entire sample and the superconducting properties are lost. The mesh and other parameters stay the same and the chosen value of $\kappa = 0.25$. The results are shown below.

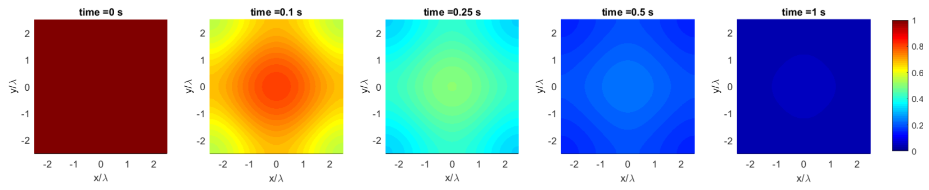


Figure 4.14: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of the first GL equation with a constant magnetic field and $\kappa = 0.25$ using COMSOL.

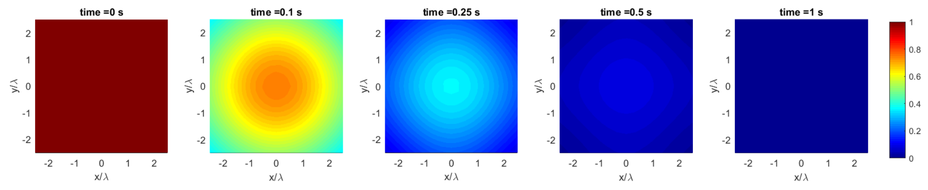


Figure 4.15: Modulus of the order parameter ($|\psi|$) as results from the time-dependent solution of the first GL equation with a constant magnetic field and $\kappa = 0.25$ using MATLAB.

The results are as expected. Due to the low κ, the complex terms that cause the vortices to arise are less dominant so the steady-state solution is indeed zero everywhere.

Finally, a test was ran on the full equation for type II superconductors on a longer time interval. All parameters and the mesh are the same as before, only now a constant initial ψ is used. In Figures 4.16 and 4.17, the results are shown.

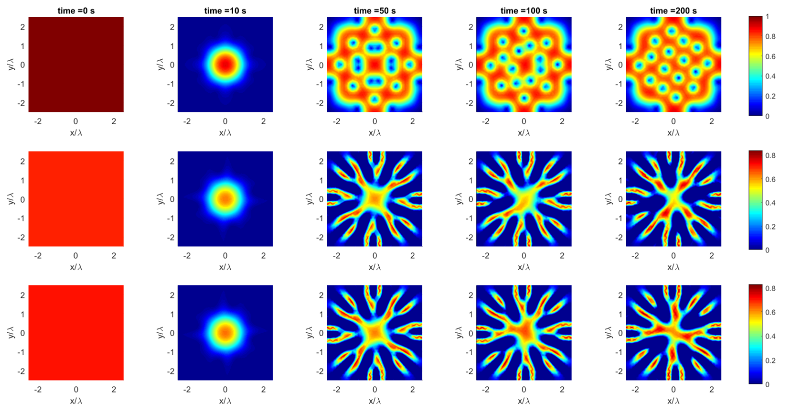


Figure 4.16: Modulus(top), real part(middle) and imaginary part(bottom) of the order parameter ($|\psi|$) as results from the time-dependent solution of the first GL equation with a constant magnetic field and $\kappa = 4$ using COMSOL.

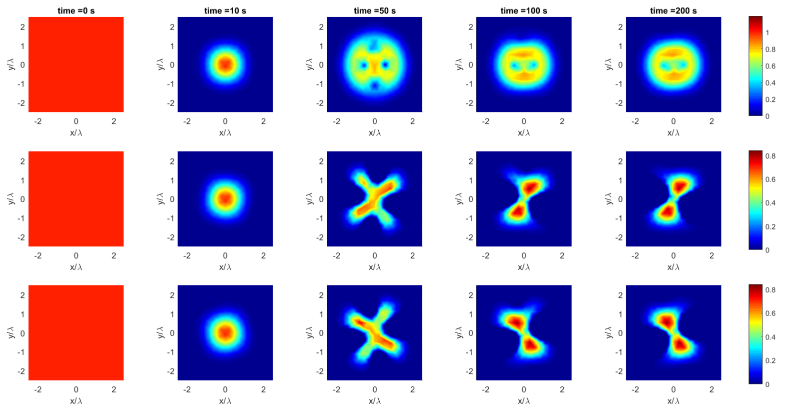


Figure 4.17: Modulus(top), real part(middle) and imaginary part(bottom) of the order parameter ($|\psi|$) as results from the time-dependent solution of the first GL equation with a constant magnetic field and $\kappa = 4$ using MATLAB.

Same as before, big differences are observed. Vortices arise but in the COMSOL the amount is higher than in the MATLAB results. It seems in MATLAB the region of non-zero $|\psi|$ is contained within a certain radius from the origin, while in COMSOL this region stretches to the boundaries of the domain. The reason for these differences is yet unclear and should be investigated further. Possibly, it could have been caused by a difference in the computation of the matrix en vector entries. At the moment no statement can be made on which result is physically correct. The COMSOL result is trusted more as it was already used for a succesful model of the fully coupled GL equations.

4.2 Second GL equation

The same process is performed for the second GL-equation. Now, different basis functions are used in the programs. MATLAB uses, as described, Nédélec basis functions to approximate $\mathbf{A}$ while COMSOL uses the Lagrangian linear basis for both components of $\mathbf{A}$. Three partial problems where solved by both programs and the results are presented below. The parameters and initial values that were used are:

- $\kappa = 4$
- $\sigma = 1$
- $B_a = 1$
- $dt = 0.01$
- $\mathbf{A}_{init} = 0 \; \forall \; x, y \in \Omega$
- $\psi = x + yi \; \forall \; x, y \in \Omega$

This again corresponds to a type II superconductor with an applied magnetic field that is near H_c. To evaluate the penetration of this field, the indentation is again added to the geometry and the mesh becomes:

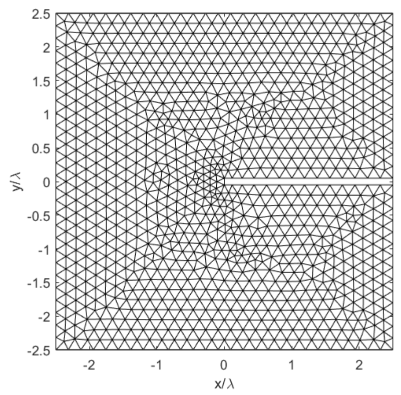


Figure 4.18: Mesh used for the numerical simulations of the second GL equation.

4.2.1 Term-wise solutions

The first problem that was solved is:

$$\sigma \frac{\partial \mathbf{A}}{\partial t} = -\nabla \times \ \nabla \times \ \mathbf{A}. \tag{4.10}$$

Shown in the figures below is the curl of $\mathbf{A}$, which is the magnetic field in the z-direction, for five different points in time.

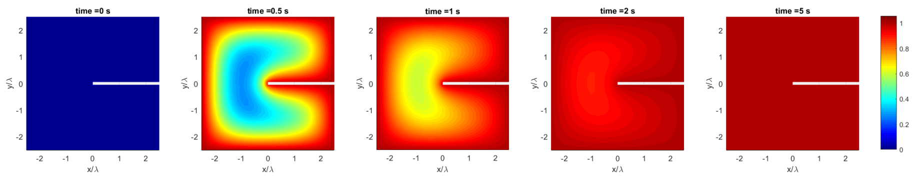


Figure 4.19: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of Equation (4.10) using COMSOL.

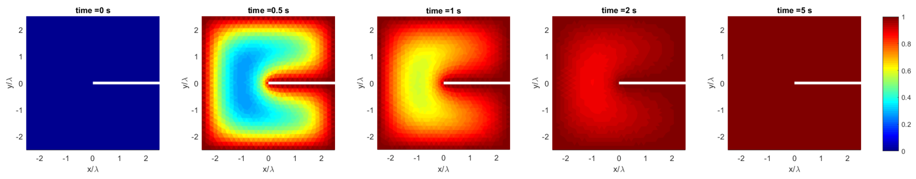


Figure 4.20: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of Equation (4.10) using MATLAB.

As it can be seen, the results are very similar. The magnetic field penetrates the entire domain, which is the physical result, as the superconducting shielding terms are not yet taken into account.

Now, $|\psi|$ is added to the equation:

$$\sigma \frac{\partial \mathbf{A}}{\partial t} = -\nabla \times \ \nabla \times \ \mathbf{A} - |\psi|^2 \, \mathbf{A}. \tag{4.11}$$

The results are:

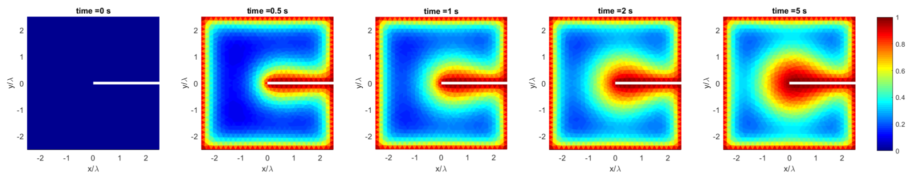


Figure 4.21: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of Equation (4.11) using MATLAB.

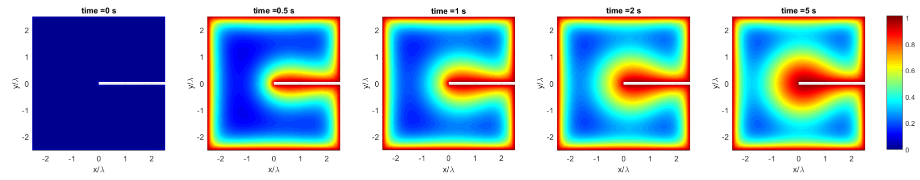


Figure 4.22: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of Equation (4.11) using COMSOL.

Again, the magnetic field intrudes the domain. Only now a steady-state is reached where the magnetic field does not penetrate the sample entirely, but it is screened from certain parts of the interior. Close to the origin this screening effect is weaker, as it is proportional to the square of the distance to the origin, and therefore the field is able to penetrate deeper into the sample in this region. The results from the COMSOL and MATLAB implementations are the same.

Finally, the equation is completed by adding the final term:

$$\sigma \frac{\partial \mathbf{A}}{\partial t} = \frac{1}{2ik} \left(\psi^* \ \nabla \psi - \psi \ \nabla \psi^*\right) - |\psi|^2 \mathbf{A} - \nabla \times \ \nabla \times \ \mathbf{A}. \tag{4.12}$$

The results are:

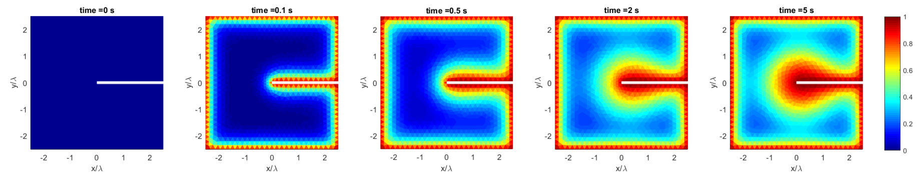


Figure 4.23: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of Equation (4.12) using MATLAB.

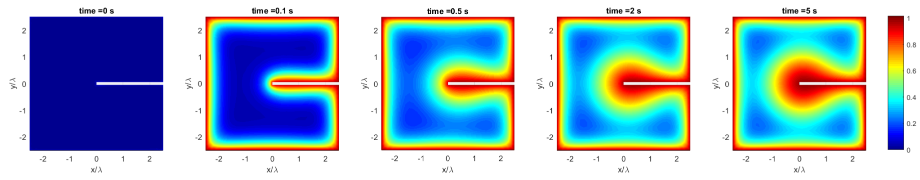


Figure 4.24: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of Equation (4.12) using COMSOL.

The results are similar to the previous ones and are again in agreement. Adding the last term in this case has a very small effect on the results. This could be explained by the fact that the spatial differentials of ψ are relatively small.

4.2.2 Additional tests

As a final test on the implementation of the second GL equation, the penetration of a magnetic field in an entirely superconducting sample is simulated. For this, the same values as before are used, except for the given ψ. This now becomes:

$$\psi = \frac{1+i}{\sqrt{2}} \;\forall\; x, y \in \Omega. \tag{4.13}$$

Also, a larger geometry is used:

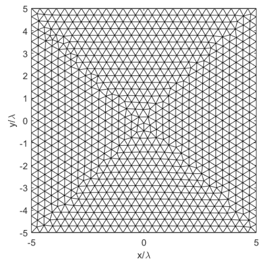


Figure 4.25: Mesh for the test on the penetration of a magnetic field into a superconducting sample.

As it can be seen in the results below, the results are in agreement. The magnetic field is screened from the interior of the sample. The penetration depth is around λ, which is in agreement with the physical expectations.

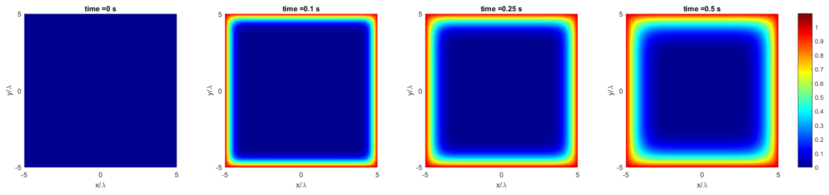


Figure 4.26: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of the second GL equation for a superconducting sample using COMSOL.

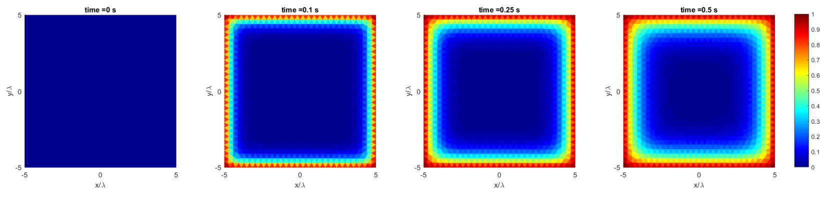


Figure 4.27: $B_z = \nabla \times \mathbf{A}$, according to the time-dependent solution of the second GL equation for a superconducting sample using MATLAB.

5 Conclusions

The behavior of superconductors in the presence of magnetic fields is accurately described by the time dependent Ginzburg-Landau equations. Using finite element methods with Lagrangian elements of the first order in 2D to solve these equations shows that vortices of non-superconducting material arise when the applied magnetic field exceeds the critical magnetic field for type II superconductors.

Results obtained by using Lagrangian elements turn out to be flawed, so other curl-conforming elements need to be used, which are provided by the Nédélec basis. Using these elements, the second GL-equation and parts of the first GL equation have been modelled successfully. A working MATLAB model using these elements for the full problem has not yet been achieved.

6 Future Research

To obtain a working model for the time dependent GL equations, more research has to be done into the differences in the results that were presented in this report. In this final section, some suggestions are made for possible causes of these differences.

The differences arose when the factor in front of the complex term of the first GL equation was changed from 1 to 2. Research could be done into what value between 1 and 2 is the tipping point for these differences. Also, a mesh convergence study can be performed on both the COMSOL and the MATLAB model to see if the results change on an even finer mesh. The same can be done for higher order elements.

Furthermore, research can be done into the way MATLAB handles linear systems of equations with complex degrees of freedom, as is the case in the first GL equation. In the implementation in COMSOL, the real and imaginary parts were divided into two dof's, but in MATLAB this was not the case and the way MATLAB handles this might be a cause for the differences.

Finally, in the MATLAB code the coupling of the two equations was initially done in a very rough manner by just taking the values from the previous iteration as given in solving the next time step. An iterative scheme can be added to refine these results.

Once a working model has been achieved, this model will need to be validated with physical measurements. When this is done, the model can be used to make predictions on the behavior of actual superconducting components and eventually it can be extended to three spatial dimensions.

Appendices

A Description of the MATLAB code

Finally, to get an idea of the actual implementation in MATLAB, some of the used code will be presented. First, the main code will be briefly discussed and then an example is given on how the large matrices and vectors are assembled.

```
%Loading mesh data
load('Mesh_squarewithslit_COMSOL.mat')
x = coordinates(:,1); y = coordinates(:,2);
n = length(coordinates(:,1));

%get elements edgewise
[elems2edges, edges2nodes] = get_edges(elems2nodes);
ne=length(edges2nodes(:,1));

%calculate affine transformations for elements
[B_K,b_k,B_K_det] = affine_transformations(coordinates,elems2nodes)
    ;
% signs in 2d for the Nedelec basis functions defined on edges
signs = signs_edges(elems2nodes);

%Defining parameters and initial values
topology = 3; topologybnd = 2; dt = 0.1; niter = 25; Q_p = 1;dim=2;
kappa=4;sigma=1;Ba=1;
Ainit=zeros(ne,1);
Psi_init=ones(n,1);
Psi(:,1)=Psi_init;
A=zeros(ne,niter+1);
A(:,1)=Ainit;

%Get integration quadrature for the unit triangle
[ ip, w, nip ] = intquad( 5, 2 );
%Get integration quadrature for the interval [0,1]
[ip1D,w1D]=intquad1D(5,0,1);

%Computation of solutions
Buildmatricesandvectors;
Buildboundaryvector;
iter=1;
Generate_Psifunc
for iter = 1:niter
    disp(['Progress: ',num2str(iter/niter*100),'%']);
    Generate_Afunc;
    Buildmatricesandvectors_eq1;
    Psi(:,iter+1) = (M+(S)*dt) \ (M*Psi(:,iter));
    Generate_Psifunc;
    Buildmatricesandvectors_eq2;
    A(:,iter+1) = (sigma*Mned+(K+T)*dt) \ (sigma*Mned*A(:,iter)+(f+
        b)*dt);
end;
```

As it can be seen on the previous page, the first part of the code is dedicated to constructing the mesh, calculating the transformations from the reference element onto the elements in the mesh, and defining the basis. Then, all parameters and the initial values for ψ and $\mathbf{A}$ are defined. Here, $\psi_{init} = 1$ is chosen on the entire domain, which corresponds to a superconducting state, and $\mathbf{A}_{init} = 0$. The applied magnetic field $B_a = 1$ on the entire boundary. The integration quadratures of specified order are defined and finally the solutions are computed.

For this computation, first the boundary vector and some matrices are constructed that are independent of time. The "Generate_func" commands define functions to compute the values for ψ and $\mathbf{A}$ respectively for each time step. Then, for each of the defined time steps, the time dependent matrices and vectors are constructed and the systems of equations are solved by using MATLAB's "\"-command.

As stated before the large matrices and vectors are constructed by using element matrices and vectors. The code that is used for this process is shown for the matrix M from Equation (3.17). The large matrix is assembled by the code below. Here "elems2nodes" denotes a $(N_p \times 3)$ matrix with containing the elements of the mesh by their nodes.

```
M                = sparse(N_p,N_p);

for i = 1:length(elems2nodes(:,1)) % for all internal elements
        GenerateElementMatrix; % Melem

    for ind1 = 1:topology
        for ind2 = 1:topology
            M(elems2nodes(i,ind1),elems2nodes(i,ind2))  = M(
                elems2nodes(i,ind1),elems2nodes(i,ind2)) + Melem(
                ind1,ind2);
        end;
    end;
end;
```

In this code, "GenerateElementMatrix" computes the elementmatrix for element e_i of the mesh. The code for this is shown below. Here l(i,index) denotes the index'th lagrangian linear basis function on the element.

```
Melem=zeros(3,3);
for index1 = 1:topology
    for index2 = 1:topology
        Melem(index1,index2)=abs(det(B_K(:,:,i)))*sum(w.*(l(ip,
            index1).*l(ip,index2)));
    end;
end;
```

References

[1] Pedersen N. F. Madsen S. Sørensen M. P. Alstrøm, T. S. Magnetic Flux Lines in Complex Geometry Type-II Superconductors Studied by the Time Dependent Ginzburg-Landau Equation. *Acta Applicandae Mathematicae*, 2010.

[2] COMSOL. -. COMSOL, Inc., Burlington, Massachusetts, USA, -.

[3] MATLAB. *version 7.10.0 (R2016b)*. The MathWorks Inc., Natick, Massachusetts, USA, 2016.

[4] M. Tinkham. *Introduction to Superconductivity*. McGraw-Hill, 1996.

[5] Segal A. Vermolen F. van Kan, J. *Numerical Methods in Scientific Computing*. VSSD, 2005.

[6] F. London and H. London. The electromagnetic equations of the supraconductor. *Proceedings: Mathematical, Physical and Engineering Sciences 149(866):71*, 1935.

[7] A.B. Pippard. The coherence concept in superconductivity. *Physica*, 19(1):765 – 774, 1953.

[8] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. Theory of superconductivity. *Phys. Rev.*, 108:1175–1204, Dec 1957.

[9] V. L. Ginzburg and L. D. Landau. *On the Theory of Superconductivity*, pages 113–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[10] Wikipedia. Magnetic potential — Wikipedia, the free encyclopedia, 2017. [Online; accessed 27-June-2017].

[11] Éliashberg G.M. Gor'kov, L.P. Dynamical properties of gapless superconductors. *Zh. Eksperim. i Teor. Fiz. 54, 612*, 1968.

[12] Pan A. V. Wang X. R. Fedoseev S. A. Hilgenkamp H. Wells, F. S. Analysis of low-field isotropic vortex glass containing vortex groups in YBa˙2Cu˙3O˙7x thin films visualized by scanning SQUID microscopy. *Scientific Reports, 5 8677*, 2015.

[13] R.G. Brown. Gauge transformations, 2014. [Online; accessed 27-June-2017].

[14] Wikipedia. Finite element method — Wikipedia, the free encyclopedia, 2017. [Online; accessed 27-June-2017].

[15] E.W. Weisstein. Gaussian quadrature – mathworld, 2017. [Online; accessed 27-June-2017].

[16] Wikipedia. Green's identities — Wikipedia, the free encyclopedia, 2017. [Online; accessed 27-June-2017].

[17] Wikipedia. Delaunay triangulation — Wikipedia, the free encyclopedia, 2017. [Online; accessed 29-October-2017].

[18] J.R. Shewchuk. *Lecture Notes on Delaunay Mesh Generation.* University of California at Berkeley, 1999.

[19] A. Schneebeli. An H(curl;Ω)-conforming FEM: Nédélec's elements of first type. . *Technical report*, 2003.

[20] Valdman J. Anjam, T. Fast MATLAB assembly of FEM matrices in 2D and 3D: Edge elements. 2015.

[21] Wikipedia. Gaussian quadrature — Wikipedia, the free encyclopedia, 2017. [Online; accessed 30-June-2017].

[22] D.A. Dunavant. High degree efficient symmetrical gaussian quadrature rules for the triangle. *Int. J. Num. Methods*, 1985.

[23] Wikipedia. Weak formulation — Wikipedia, the free encyclopedia, 2017. [Online; accessed 30-June-2017].

[24] Wikipedia. Comsol multiphysics — Wikipedia, the free encyclopedia, 2017. [Online; accessed 27-June-2017].

[25] T. Garaud. Benchmarking dynamical processes in Ginzburg-Landau theory. *KTH-Royal institute of Technology, Stockholm*, 2016.