UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

A Critical Analysis of Design Flaws in the Death Star

Luke Skywalker
99652154

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Dr O. W. Kenobi

October 2099

Acknowledgements

I would like to thank my dog, Muffin. I also would like to thank the inventor of the incubator; without him/her, I would not be here. Finally, I would like to thank Dr Herman Kamper for this amazing report template.

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *Student number*	Handtekening / *Signature*
Voorletters en van / *Initials and surname*	Datum / *Date*

Abstract

English

The English abstract.

Afrikaans

Die Afrikaanse uittreksel.

Contents

List of Figures

List of Tables

Nomenclature

Variables and functions

$p(x)$	Probability density function with respect to variable x.
$P(A)$	Probability of event A occurring.
ε	The Bayes error.
ε_u	The Bhattacharyya bound.
B	The Bhattacharyya distance.
s	An HMM state. A subscript is used to refer to a particular state, e.g. s_i refers to the i^{th} state of an HMM.
$\mathbf{S}$	A set of HMM states.
$\mathbf{F}$	A set of frames.
$\mathbf{o}_f$	Observation (feature) vector associated with frame f.
$\gamma_s(\mathbf{o}_f)$	A posteriori probability of the observation vector $\mathbf{o}_f$ being generated by HMM state s.
μ	Statistical mean vector.
Σ	Statistical covariance matrix.
$L(\mathbf{S})$	Log likelihood of the set of HMM states $\mathbf{S}$ generating the training set observation vectors assigned to the states in that set.
$\mathcal{N}(\mathbf{x}\|\mu, \Sigma)$	Multivariate Gaussian PDF with mean μ and covariance matrix Σ.
a_{ij}	The probability of a transition from HMM state s_i to state s_j.
N	Total number of frames or number of tokens, depending on the context.
D	Number of deletion errors.
I	Number of insertion errors.
S	Number of substitution errors.

Acronyms and abbreviations

AE	Afrikaans English
AID	accent identification
ASR	automatic speech recognition
AST	African Speech Technology
CE	Cape Flats English
DCD	dialect-context-dependent
DNN	deep neural network
G2P	grapheme-to-phoneme
GMM	Gaussian mixture model
HMM	hidden Markov model
HTK	Hidden Markov Model Toolkit
IE	Indian South African English
IPA	International Phonetic Alphabet
LM	language model
LMS	language model scaling factor
MFCC	Mel-frequency cepstral coefficient
MLLR	maximum likelihood linear regression
OOV	out-of-vocabulary
PD	pronunciation dictionary
PDF	probability density function
SAE	South African English
SAMPA	Speech Assessment Methods Phonetic Alphabet

Chapter 1

Introduction

Chapter 2

Literature Review

2.1. Theory and applications of superconductivity

In order to understand and apply the methods described in [2] one must first understand the basic theory behind superconductivity as well as some examples of how this theory is used in practice.

2.1.1. Superconductivity

REVIEW NEEDED

Since the first discovery of superconductivity a couple of successfully theories have been put forth to explain the phenomenon. The London theory is a framework that describes the qualitative behaviour of superconductors and correctly describes perfect diamagnetism and zero resistance but fails to explain the effect on a microscopic level [4]. The London equations (equation 2.1 and equation 2.2) [5] is an addition to Maxwell's equations.

$$E = \frac{\partial}{\partial t}(\Lambda J_s) \tag{2.1}$$

$$h = -c\nabla \times (\Lambda J_s) \tag{2.2}$$

Here Λ is a phenomenological parameter determined through experimentation. The London equations allow us to calculate the current distribution in a superconductor which is very important to the objective of this project. BCS theory put forth a microscopic model of superconductors and explains the phenomenon as a quantum mechanical effect. The details are out of scope for this project but on a crude qualitative level BCS theory can be explained by the pairing of electrons in the crystal lattice of the superconductor allowing them to be considered one particle. These particles are known as cooper-pairs. At extremely low temperatures the formation of these cooper pairs are energetically favourable [6]. Electron pairs in this state can flow through the superconductor unimpeded. BCS theory is the most successful model of superconductivity discovered to date.

2.1.2. The Josephson junction

In superconducting electronics the active component is the Josephson junction [7]. The Josephson junction refers to a situation where two superconductors are connected through a thin non-conductive barrier. If this barrier is thin enough one can observe what is known as the Josephson effect. This phenomenon is can be explained by considering the effect of quantum tunnelling of cooper pairs through the non-conductive boundary. For a sufficiently large barrier one can express the ensemble average wave function in each superconductor independently [7]:

$$\Psi = |\Psi(\mathbf{r})| \exp\{i\theta(\mathbf{r})\} \tag{2.3}$$

This is due to the fact that it is energetically favourable for cooper pairs in proximity to one another to lock phases [7] allowing one to express a large collection of these cooper pairs as one ensemble wave function. The idea that it is energetically favourable for cooper-pairs in proximity to one another extends to the situation where the superconductors are separated by an insulating boundary. When the barrier is sufficiently small the energy of the system can be reduced by the coupling of wave functions in their respective superconductors [7]. This results in cooper-pairs being able to move across the boundary without energy loss. Following the derivation in [6] one can describe the system behaviour using Schrödinger's equation when a voltage is applied to the junction [6]:

$$i\hbar\frac{\partial\Psi_1}{\partial t} = U_1\Psi_1 + K\Psi_2 \tag{2.4}$$

$$i\hbar\frac{\partial\Psi_2}{\partial t} = U_2\Psi_2 + K\Psi_1 \tag{2.5}$$

Here $U_1 = qV/2$ and $U_2 = -qV/2$ [6] refer to the energy of the two wave functions and K refers to the coupling energy between each wave function. By taking 2.3 and setting $|\Psi(\mathbf{r})|$ equal to $\sqrt{\rho}e^{i\theta}$, where ρ refers to the cooper pair density in the super conductor. One can substitute the result into equations 2.4 and 2.5. From this substitution one can find that the rate of change of electron density on either side of the junction to be described by the following equation [6]:

$$\dot{\rho}_1 = \frac{2}{\hbar}K\sqrt{\rho_1\rho_2}sin(\delta) \tag{2.6}$$

Where ρ_1 is the electron density on one side of the junction. The rate of change of charge on the other side of the junction is simply $\dot{\rho}_2 = -\dot{\rho}_1$ [6]. The rate of change of charge is the current and therefore the current through the junction can be expressed as follows [8]:

$$I = I_0 sin(\theta_2 - \theta_1) \tag{2.7}$$

Equation 2.7 is known as the current-phase relation of the DC Josephson effect. The phases of the currents on each side of the boundary is described by equation 2.8 [6].

$$\dot{\theta}_2 - \dot{\theta}_1 = \frac{qV}{\hbar} = \dot{\delta} \tag{2.8}$$

Integrating on both sides yields equation 2.9:

$$\delta(t) = \delta_0 + \frac{q}{\hbar} \int V(t)dt \tag{2.9}$$

It is important to note that 2.7 is only valid for a limited number of cases and the actual current-phase relation is often more complicated [8]. The applications of equation 2.7 is limited to analysing analogue and digital devices based on Josephson junctions [8].

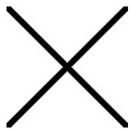

Figure 2.1: The circuit symbol for a Josephson junction

The RCSJ model

The math in this section needs review The phase-current and phase-voltage relation makes the assumption of a perfect Junction. This assumption is inaccurate and as such the RCSJ model is used to more accurately model the behaviour of a junction [5]. The resistively and capacitively shunted junction is a simple model used to describe the dynamics of a junction. As the name implies it consists of a resistor (R), capacitance (C) and a Josephson current (I_s). In a non-ideal junction a displacement current flows between the two superconductors. This displacement current is modelled by the capacitor. The resistive element models the effect of quasi-particle tunnelling across the boundary [7]. Figure Insert fig depicts the RCSJ model. To derive the current-voltage relation of an RCSJ model one can express the voltage across the parallel components as a function of the current flowing into the junction:

$$I = I_c sin(\delta) + C\frac{dV}{dt} + \frac{V}{R} \tag{2.10}$$

Equation 2.10 can be re-written using equation 2.9. The resulting equation is a second-order non-linear, inhomogeneous differential equation for which analytical solutions are not available. For the simple case when $C = 0$ a differential equation of the form of equation 2.11 can be obtained.

$$K = \frac{d\varphi}{d\tau} + sin(\varphi) \tag{2.11}$$

This equation can be integrated directly using separation of variables and a Weierstrass substitution [7]. A detailed derivation is listed in appendix List appendix. The resulting equation describes the dynamics of the phase shift for a constant input current. The frequency of the current through the junction is determined by the dynamics of the phase. The period of the phase-dynamics is the Josephson frequency. Using the Josephson frequency relation [7],

$$\omega = \frac{q}{\hbar}V \tag{2.12}$$

as well as the period of the equation for the phase, we find the time average voltage of the junction to be [7]:

$$V = \begin{cases} 0 & \text{for } I \leq I_c \\ I_c R[(\frac{I}{I_c})^2 - 1]^{\frac{1}{2}} & \text{for } I > I_c \end{cases} \tag{2.13}$$

Graphing equation 2.13 yields the figure below: Insert fig

2.1.3. SQUID's

An application of the Josephson effect is the superconducting quantum interference device (SQUID). In essence a SQUID refers to a superconducting ring that contains one or more Josephson junction. This interference effect is analogous to interference one might encounter in the field of optics [9]. The SQUID is a highly sensitive device that can in some cases measure fields as weak as $5 \times 10^{-14}T$ [10].

Flux quantisation

To understand the basic operation of a SQUID, one must first understand the concept of flux quantisation. To do so we consider a superconducting ring in the presence of a uniform magnetic field. The ring is superconducting, so it exhibits the Meissner effect and thus the current density inside the ring is zero. Recall that the flux through a ring is:

$$\Phi = \oint \mathbf{A} \cdot d\mathbf{s} \tag{2.14}$$

Now consider the equation for the current density in a superconductor [6]:

$$\mathbf{J} = \frac{\rho\hbar}{m}(\nabla\theta - \frac{q\mathbf{A}}{\hbar}) \tag{2.15}$$

The current density inside the ring in the superconducting state is zero so equation 2.15 becomes:

$$\nabla\theta = \frac{q}{\hbar}\mathbf{A} \tag{2.16}$$

Integrating on both sides around a curve deep inside the superconductor such that the assumption that the current density is zero holds we can express equation 2.16 as:

$$\oint \nabla\theta \cdot d\mathbf{s} = \frac{q\Phi}{\hbar} \tag{2.17}$$

Recognizing $\nabla\theta$ as vector field with potential function θ, we can simply write the left-hand side of equation 2.17 as $\theta(\mathbf{r_1}) - \theta(\mathbf{r_1})$. One might assume that the left-hand side of the equation 2.17 must be equal to zero. This is incorrect because the absolute phase, meaning the potential function θ cannot be determined and can only be determined up to some constant because the gradient of the scalar potential function will take any constant to zero. Therefore, the phase can only be determined relative to some point. According to [6] the only limitation we can place on the phase of the wave function is that the wave function must be singularly valued. This is intuitively understood as the fact that a particle cannot have two different amplitudes to be in a certain quantum state as that would imply that it has two probabilities associated with the exact same quantum state [6]. From equation 2.3 one can conclude that the phase can change by any integer multiple of 2π as this results in the exact same wave function. Equation 2.17 then becomes:

$$2\pi n = \frac{q\Phi}{\hbar} \tag{2.18}$$

Clearly equation 2.18 implies that the flux through the superconducting loop must be quantized as n can only take on integer values.

The DC SQUID

Spurred on by developments in reliable manufacturing processes of Josephson junctions the DC SQUID has become the dominant device in the field of niobium based sensors [10]. The initial discovery of the DC SQUID by [9] in 1964 came shortly after the discovery of the Josephson junction in 1962 [11]. The simplest realisation of the DC SQUID consists of two Josephson junctions connected in parallel as depicted in figure 2.2.

A relationship between the current and the flux through the loop of the SQUID would like to be derived. The current through each Josephson junction is denoted as I_1 and I_2 for junction 1 and 2 respectively. From figure 2.2 one can easily see that there are 2 possible paths through the SQUID. Following the derivation of [6] one can write the change in phase through each branch of the squid using equations 2.19 and 2.20 [6].

$$\Delta\theta_1 = \theta_1 + \frac{2q_e}{\hbar}\int_{\text{left branch}} \mathbf{A} \cdot d\mathbf{s} \tag{2.19}$$

$$\Delta\theta_2 = \theta_2 + \frac{2q_e}{\hbar}\int_{\text{right branch}} \mathbf{A} \cdot d\mathbf{s} \tag{2.20}$$

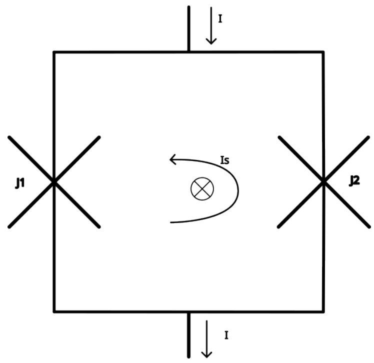

Figure 2.2: A circuit schematic depicting the configuration of a simple DC SQUID. The current I is the bias current and the current I_s is the screening current developed by an external magnetic field

Where θ_1 and θ_2 is the change in phase across their respective junctions. Similarly, $\Delta\theta_1$ and $\Delta\theta_2$ refer to the change in phase through the left and right branch respectively. As previously discussed, the integral of the phase gradient around a closed loop must be an integer multiple of 2π. As a consequence, subtracting equation 2.20 from 2.19 must yield $2\pi n$. Using equation 2.17 we derive equation 2.21.

$$\theta_2 = \theta_1 - 2\pi n + \frac{2q_e}{\hbar}\Phi \tag{2.21}$$

Using equation 2.7, 2.21 and a KCL at the top or bottom node, the relation between the current going into the loop and the flux through the loop is derived and equation is found. Here the $2\pi n$ term can be neglected as a $2\pi n$ shift in phase does not change the value of the function.

$$I = I_2 sin(\theta_2) + I_1 sin(\theta_2 - \frac{2q_e}{\hbar}\Phi) \tag{2.22}$$

Equation 2.22 describes the general behaviour of the DC SQUID. The current voltage characteristics are shown in the figure below Get figure

It is worth pointing out the mechanism through which flux is quantized in the SQUID loop. When an external field is applied to the washer a screening current begins to flow around the loop to oppose the applied field. The magnitude of the current is determined by the loop inductance. Assuming that the applied flux density is constant throughout the hole in the loop, the magnetic flux generated by the screening current is $\Phi_s = LI_s$. The total flux in the loop is $\Phi_{\text{ext}} + \Phi_s = n\Phi_0$. If applied flux is less than $\frac{\Phi}{2}$ the screening current flows to oppose the build up of flux in the loop by ensuring that $\Phi_s = -\Phi_{\text{ext}}$ and thus the total flux in the loop is 0. When the applied flux increases to $\frac{\Phi}{2}$ the flux state of the loop changes and the screening current changes direction [10] to ensure that the total flux in the loop is Φ_0. If the external flux is further increased the flux induced by the screening current decreases from $\frac{\Phi_0}{2}$ to 0 such that the total flux in the loop remains Φ_0. The trend continues as the external field strength is increased. Because of this mechanism the critical current and voltage-flux characteristics of the SQUID always has a period of Φ_0 [12]. As pointed out by [10], this allows the SQUID to be automatically calibrated as 1 period corresponds to 1 flux quantum in the loop.

2.1.4. Practical DC SQUIDS and design considerations

The resistively shunted junction

Practical realisation of a DC SQUID requires that the designer connect shunt resistors to each junction [10]. This is done such that the system is strongly over damped [12]. In the context of the discussion on the I-V characteristics of the Josephson junction the addition of the shunt resistors correspond to the effect of the junction capacitance becoming negligible. In the absence of a shunt resistor the I-V characteristics of a Josephson junction has hysteresis [10].

Effective Area and Inductance

Two important quantities for the design and analysis of SQUID sensors are the effective area and the SQUID inductance. The effective area of a SQUID washer is defined as [12]:

$$A_{\text{eff}} = \frac{\Phi_s}{B_a} \tag{2.23}$$

In equation 2.23 Φ_s refers to the flux coupled into the SQUID loop as a result of an applied magnetic field B_a. The effective area is typically higher than the geometric area of the washer hole due to the Meissner effect [10]. As will be shown in the next section there is a limit on the inductance of a SQUID loop and by extension a limit on how large one can make a SQUID. Critically, the field sensitivity is inversely proportional to the effective area. The magnetic field noise is inversely proportional to the geometric area of the hole in the SQUID loop. The design then comes down to creating a SQUID loop that maximises the effective area but minimizes the actual area of the SQUID loop to keep the SQUID loop inductance and thus the field noise to a minimum [12].

The uncoupled SQUID

Below is an example of an uncoupled DC SQUID structure. The uncoupled SQUID is the simplest of designs as it only consists of a single superconducting loop with 2 resistively shunted Junctions. The SQUID characteristics are determined by the loop inductance, critical current, shunt resistance and parasitic junction capacitance [10]. Design involves choosing the optimal set of these parameters to satisfy the design requirements. This is often not a trivial task due to the non-linear nature of the equations describing the behaviour of the Josephson junction and by extension the DC SQUID. Through extensive

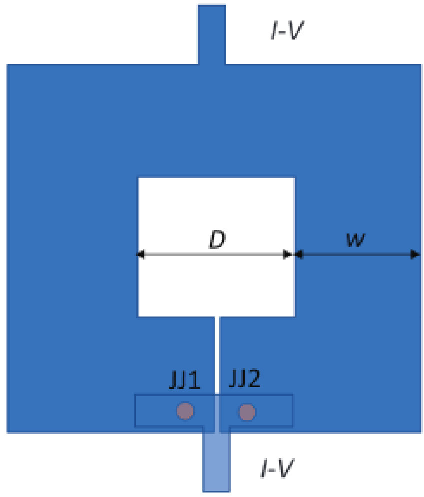

Figure 2.3: A diagram of a practical uncoupled DC SQUID structure [1]

simulation a number of key design parameters have been identified [10]. The first of which is described by equation 2.24 and is referred to as the noise parameter [12].

$$\Gamma = \frac{2\pi k_b T}{\Phi_0 I_c} \tag{2.24}$$

The noise parameter is the quotient of the thermal energy and the Josephson coupling energy [12]. It has the effect of an apparent reduction in the critical current of a junction at low voltages [10].

The second design consideration is the quality or "Q" factor. For currents the phase shift induced by the Josephson junction can be interpreted as an inductor [12]. Under this approximation it is clear to see how the combination of the shunt capacitance and junction create an RC resonator. As pointed out by [10] it is often desirable to keep the quality factor of resonating circuits in non-linear systems close to unity. This leads to the next design rule [12]:

$$\beta_c = Q_j^2 = 2\pi I_c R^2 \frac{C}{\Phi_0} \approx 1 \tag{2.25}$$

This parameter is largely controlled by designer by tuning the shunt resistance.

The next design rule sets a constraint on the choice of L (the inductance of the SQUID washer). The magnitude of the screening current in the loop is related to the applied flux through the inductance of the SQUID washer. Increasing the SQUID loop inductance results in a smaller screening current induced in the loop to keep the flux in the loop at an integer multiple of the flux quantum. This leads to a smaller modulation depth of the critical current and by extension the modulation depth of the voltage flux characteristic of the SQUID [10]. For this reason it is generally better to keep the inductance of the SQUID washer low. Simulations have shown that for very low inductances the SQUID noise increases, so a compromise is in order [10] leading to the next design rule:

$$\beta_L = \frac{2LI_c}{\Phi_0} \approx 1 \tag{2.26}$$

Coupled SQUIDS

See pg 177

SQUID readout

See pg 139 of SQUID hb

2.2. Noise in SQUIDS

The importance of understanding, modelling and predicting the noise energy of a SQUID design cannot be understated. A reduction in flux noise necessarily increases the sensitivity of a SQUID system. As reported by [13] a typical commercially available SQUID system can achieve magnetic field resolutions of 5fT/Hz. Experimental results suggest that the current consensus on the origin of white noise is in agreement with the theory, that is the origin of the noise is known. The theory explains the white noise as thermal noise and the statistical properties are usually well known [10] [12]. For frequencies greater than about $10 - 100$Hz the thermal noise dominates. Below $10 - 100$Hz the so called $1/f$ noise dominates [1]. Unlike thermal noise no consensus has been found on the origin of $1/f$ noise. According to [12], in the fields of biomagnetism and geophysics the low noise properties of SQUIDS must extend to frequencies below 1Hz. Furthermore, [14] reports the primary cause of decoherence in frequency tuneable quits is the so called $1/f$ flux noise, once again highlighting the important role $1/f$ flux noise has.

2.2.1. 1/f flux noise

Although there is no consensus on the exact origin of $1/f$ noise many agree that the origin of the excess noise at low frequencies is due to randomly reversing of magnetic moments distributed thinly on the surface of a super conductor [15]. As [15] reports the noise has a power spectral density of the form of equation 2.27.

$$S_\Phi(f) = \frac{A}{f^\alpha} \tag{2.27}$$

The parameter A is insensitive to the geometry and materials of the substrate [2]. The parameter α is the slope of the noise on a log-log plot. In [16] the authors propose that the origin of magnetic moments on the surface of the superconductor can be attributed to O_2

molecules absorbed on the surface. In [17] the authors provide further supporting evidence to the claims of [16] by investigating the effect of surface treatments on low frequency flux noise. They found that surface treatments as well as a better sample vacuum environment led to significant improvement in low frequency flux noise. Experimental results suggest the spin density on the surface of the superconductor to be $5 \times 10^{17} m^{-2}$. According to [2] various authors have characterized the noise due to surface spins by calculating the mean square flux noise ($\langle \Phi^2 \rangle$). The authors make use of a model where spins are independently and identically distributed with each spin having a magnetic moment equal to the Bohr magneton (μ_B). The results correlated with experimental evidence as it conformed to the properties of the power spectral density one expects from the low frequency flux noise [2]. Specifically, the models conformed to the property that the magnitude of the flux noise is a weak function of the geometry of the SQUID.

2.2.2. Techniques for predicting mean square flux noise

The importance of being able to predict the low frequency flux noise in SQUID's has led many to attempt to predict it using the mostly agreed upon model of magnetic defects on the surface of the super conductor.

Change title

Numerical method by Bialczak et al.

In the work done by Bialczak et al. [18], the authors describe how they calculate the mean square flux noise due to a random distribution of surface defects on the substrate of the SQUID. Critically, the authors do not make the assumption that defects are confined to the surface of the superconducting film. In this method a test loop is used to represent a surface defect. The test loop is dimensioned such that the product of the area of the loop and the current in the test loop is equal to the Bohr magneton. The authors used FastHenry to calculate the mutual inductance between the SQUID and the test loop. Once the mutual inductance is known, one can simply relate it to the flux coupled into the loop by a single magnetic moment by multiplying the test current by the mutual inductance [18].

$$\Phi_s = M(x, y)I \tag{2.28}$$

The method requires a new simulation to be run for each location and orientation of the test loop. This is computationally taxing and severely limits the resolution and accuracy of the calculation [2]. Despite the heavy computational cost of the method it is fairly general and can easily be extended to different geometries. In theory this method could be used to verify the validity of the theoretical model because, given a powerful enough computer, one could perform an exact simulation, neglecting the error due to meshing, where the test loop is dimensioned such that it is on the order of a single surface defect.

The authors found that the results they obtained from their numerical model match best when an areal spin density of $5 \times 10^{17}\mathrm{m}^{-2}$

The analytic method of Koch et al.

The approach taken by [19] attempts to derive analytic solutions for the mean square low frequency flux noise in a SQUID. In [19] the authors examine a theoretical model where the flux noise is a result of two-level state (TLS) defects in the oxides of the superconducting film [19]. A TLS defect is a microscopic defect that can be described by a two-state quantum system [19]. The defects can in principle be any plausible two state quantum system but in [20] the authors list the following as examples of such defects:

- Tunnelling atoms
- Dangling electronic bonds
- Surface impurities
- Trapped charges

To derive an analytic solution for the mean square flux noise figure one has to make assumptions about the nature of the TLS defects. In [19] the assumption that all defects have a magnetic moment equal to the Bohr magneton, the assumption that defects are uniformally distributed across the surface of the superconducting film and the assumption that the magnetic moments of the defects are randomly orientated is made.

We start by recognizing that the mutual inductance between any 2 arbitrarily shaped loops in space is equal. That is the inductance matrix is of the form:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} L_1 & M \\ M & L_2 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \tag{2.29}$$

The inductance is a function of the geometry of the problem and is therefore independent of the flux and current terms [6]. If one sets $I_2 = 0$ and $I_1 = I'$ it is easy to solve for the mutual inductance:

$$M = \frac{\lambda_2}{I'} \tag{2.30}$$

Doing the same but this time setting $I_1 = 0$ and $I_2 = I'$:

$$M = \frac{\lambda_1}{I'} \tag{2.31}$$

Because M is constant in both equations 2.31 and 2.30, λ_1 must equal λ_2. This demonstrates the principle of reciprocity. Clearly it does not matter which loop is excited by the current I' the flux remains the same in both cases. This is leveraged by [19] because unlike the method employed by Bialczak et al. [18], the test loop becomes the SQUID washer as the flux coupled into the washer due to a magnetic moment is the same as the flux coupled into the area of a magnetic moment (defined as $\boldsymbol{\mu} = \mathbf{S} \cdot I$ where S refers to the vector area) due to a test current in the SQUID washer.

For a thin wire of diameter D and loop radius R, the magnetic field on the surface and tangent to the circular cross-section of the wire is approximated for a large loop radius by applying the ampere circuital law. We choose a path of integration around the diameter of the wire such that the assumption that the magnetic flux density tangent to the cross-section of the loop is constant can be made. Under these assumptions' equation 2.32 is found.

$$B = \frac{\mu_0 I}{\pi D} \tag{2.32}$$

Following the law of reciprocity we can express the flux through the loop due to a surface defect randomly orientated on the ring at a position $\mathbf{r}$:

$$\Phi = \boldsymbol{\mu}_{\mathbf{B}} \cdot \frac{B(\mathbf{r})}{I} \tag{2.33}$$

The defects are uniformally distributed, so the number of magnetic moments is simply $\sigma 2\pi^2 RD$.

$$\langle \Phi^2 \rangle = \frac{\mu_B^2 \mu_0^2}{\pi^2 D^2} \cdot \langle (\mathbf{a}_{\mathbf{m}} \cdot \mathbf{a}_{\text{bfield}}(\mathbf{r}))^2 \rangle \cdot \text{Surface area of the loop} \cdot \sigma \tag{2.34}$$

Here $\mathbf{a}_{\mathbf{m}}$ is a random unit vector representing the direction the magnetic moment is pointing and $\mathbf{a}_{\mathbf{b}}(\mathbf{r})$ represents the direction of the b field as a unit vector. Trivially, the expectation of the dot product between a unit vector and a random unit vector in $\mathbb{R}^3$ (assuming the random vector has a uniform distribution) evaluates to $\frac{1}{3}$. Simplifying equation 2.34 and applying this result equation 2.35.

$$\langle \Phi^2 \rangle = \frac{2\mu_0^2 \mu_B^2 \sigma R}{3D} \tag{2.35}$$

This is the same result as found in [19]. Using the same technique [19] derives the equation for the mean square flux noise in a thin film circular loop of radius R and track width W to be:

$$\langle \Phi^2 \rangle \approx \frac{2\mu_0^2 \mu_B^2}{3} \sigma \frac{R}{W} [\frac{\ln(\frac{2bW}{\lambda^2})}{2\pi} + 0.27] \tag{2.36}$$

The authors concluded that the standard TLS model incorrectly describes the mechanism through which low frequency 1/f flux noise arises. Despite this conclusion, the analytical methods used by [19] is valuable as the use of the principle of reciprocity is the key to the next method used to calculate mean squared flux noise figures.

The numerical method of S.M. Anton et al.

This method follows on closely from the method employed by [19]. In [2] the same assumptions are made about the nature of the surface defects as in [19]. This technique takes the analytical method demonstrated by [19] and implements it numerically. To do this [2] makes use of a superconducting version of FastHenry to compute the current distribution due to a time varying voltage source applied to a test port on the SQUID washer as shown in figure 2.4. The author noted that the time complexity of FastHenry at

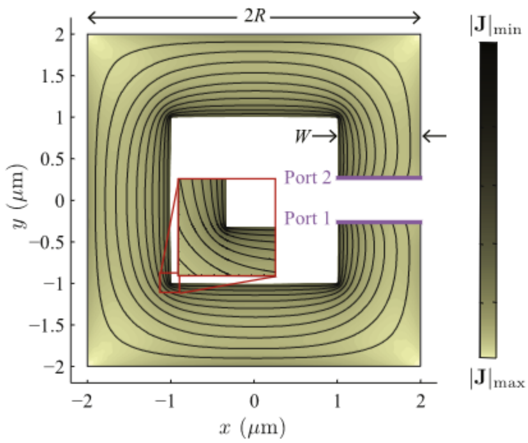

Figure 2.4: A figure produced by [2] showing the use of a test port on the SQUID washer

the time was roughly $\mathcal{O}(N_{\text{seg}})$, where N_{seg} refers to the number of segments in the mesh. FastHenry did not support a meshing scheme where segments could have varying segment widths [2]. The author concluded that in order to make the computational load reasonable while still capturing the rapidly varying nature of the current distribution localized to specific regions of the geometry, a new meshing scheme needs to be developed [2]. To solve this problem the author made use of the software package InductEx to create an optimized mesh where the mesh size is tuned to ensure that the change in current density from one mesh element to the next never exceeds a certain threshold. Once the current distribution is known, the magnetic field on the surface of the SQUID can be calculated using the law of Biot and Savart [2]:

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \sum_{\text{sll segments}} [J_n \times \int_{V_n} \frac{\mathbf{s} \cdot d\mathbf{V}}{|\mathbf{s}|^3}] \tag{2.37}$$

The current is assumed to be constant throughout each individual segment so J_n is taken out of the integral in equation 2.37. This allows the integral to be solved analytically as it only depends on the geometry of a segment and not the current distribution [2]. Once the magnetic flux density on the surface of the conductor is found, the principle of reciprocity can be used to calculate the mean square flux noise. In [2] equation 2.38 is listed.

$$\langle \Phi^2 \rangle = \frac{N\mu_B^2}{3I^2} \langle \mathbf{B}^2(\mathbf{r}) \rangle \tag{2.38}$$

In conclusion, the author reduces the problem of calculating the mean square flux noise to finding the average magnetic flux density on the surface of the conductor. A detail that is worth mentioning is that the number of spin densities N in equation 2.38 is considered large enough such that one can make a continuum approximation [2].

Chapter 3

Solution development

The goal of this project is to develop a tool for calculating the low frequency mean square flux noise figure given a SQUID's geometry with the use of the software package InductEx. In [2] the authors propose a framework for implementing such a tool but then only applied the method to a square washer. As pointed out in [2] the other numerical technique suffers from performance and resolution problems. This section will serve to communicate the design effort by systematically breaking down the problem and motivating design decisions along the way. I will start by first identifying the design requirements.

3.1. Design goals

In section 2 it was made clear that the origin of the low frequency noise is still unknown. The S.M. Anton et al. pointed concluded [2] by remarking on the flexibility of his numerical framework. It can readily be adapted, with only minor modifications to the framework, to account for a variety of different models. As such a major design requirement is to reflect the flexibility of the framework within the design of the tool. The second and perhaps most obvious design goal is that the tool must be fast. The end user of such a tool is a SQUID designer. As discussed in chapter 2 the design of SQUID systems relies heavily on computational methods. The non-linear nature of the Josephson junctions makes analytical solutions to design problems not feasible. In such cases the designer often has to rely on an iterative approach to design. The tool must be generalized to work on any geometry one might give it. The last design goal is to use TetraHenry for the calculation of the magnetic flux density.

3.2. High Level System Design

The project calls for the development of 3 independent modules. The first module must be able to take a certain geometry and extract the mesh of the surface of said geometry as this will be used for calculation of the surface magnetic flux density. The second module must implement mesh refinement around regions of rapidly varying currents across mesh nodes. The last module must implement the numerical framework as described by [2] using the simulation results from InductEx.

To understand the role the tool will play one must first understand the basic interactions a user might have with the software package. One must also understand what each entity requires as input and what its outputs are. Figure 3.1 summaries this and figure 3.2 shows how each module interacts with this process.

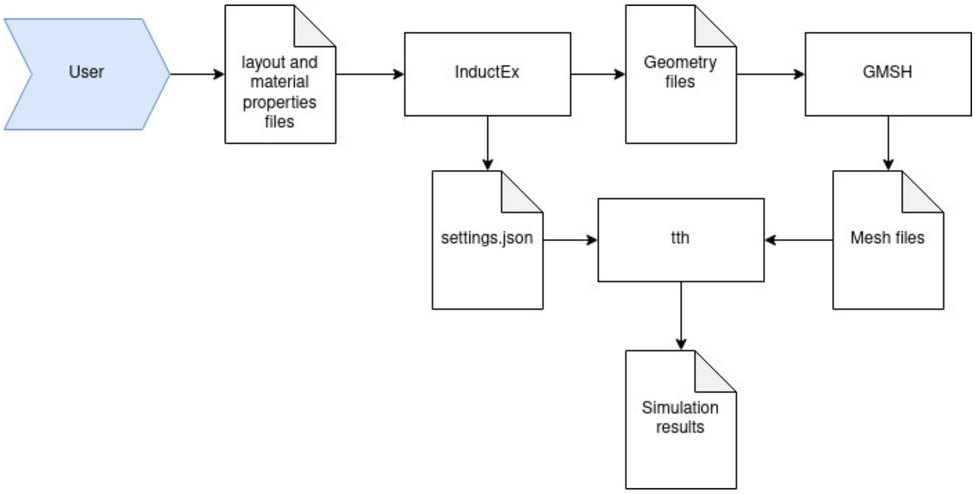

Figure 3.1: A process showing how the average user might interact with InductEx and TetraHenry. The figure also shows typical input and output files at each entity in the process.

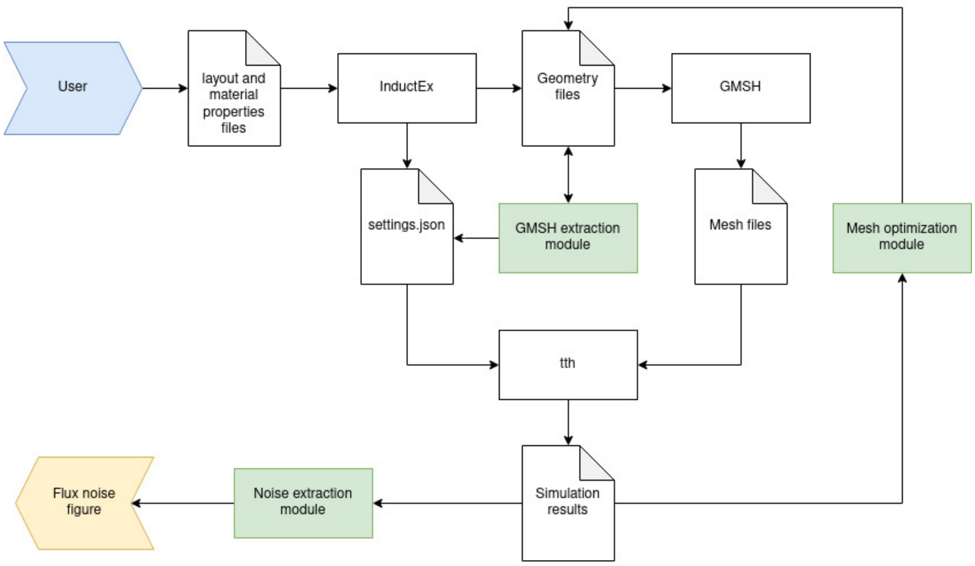

Figure 3.2: A process chart showing how the user interacts with InductEx as well as how each module will interact with this process.

3.2.1. InductEx

InductEx has gone through major changes since the publication of [2]. InductEx is a powerful tool that supports a large feature set. Most of these features are not relevant to this project. I only require magnetic field calculations for the surface of the conductor, so these features will mostly be ignored.

3.2.2. GMSH

GMSH is an open-source tool that is used to discretize geometry [3]. GMSH supports a scripting language to specify geometry before discretization. InductEx generates a file in this scripting language to be run through GMSH to generate a finite element mesh for use in TetraHenry. GMSH typically accepts a ".geo" file and outputs a ".msh" file. In GMSH you specify the geometry in a hierarchical fashion. You start by defining the lowest dimension elements and build up the geometry from there. In GMSH the lowest dimension entity is the point and is referred to as a "0 dimensional" entity. Higher dimensional entities are bounded by lower dimensional entities for example: A volume is bounded by a set of 2 dimensional surfaces. A surface is defined by a closed loop of 1 dimensional curves. A curve is defined by the 0 dimensional points. This fact will become useful in the design of the GMSH extraction module. GMSH entities are added to physical groups specified by the user. Physical groups are essentially just names for collections of mesh entities. The most common use of physical groups is to use them to define material properties. The implementation and use of physical groups depend on the context GMSH is used in. In TetraHenry physical groups are used to specify a number of things but most relevant to this project, it is used to specify the mesh entities to be used for field calculations.

3.2.3. TetraHenry (TTH)

TetraHenry is the numerical field solver that will be used to calculate the surface magnetic flux density. This project will interact directly with TetraHenry. Refer to appendix Insert sample json file in appendix for the template ".json" input file used. Depending on the setup of the simulation, TetraHenry will write the results to a folder titled output. When specified, TetraHenry will write the magnetic flux density to a "vtk" file as an unstructured grid. TetraHenry also supports a feature where a loop can be excited by specifying a "hole" port. This eliminates the problems [2] had where the current distribution was distorted around the port where the test current is injected.

3.3. Detailed Design

3.3.1. Choice of programming language

The visualisation tool kit (VTK) is an open-source modelling and visualisation tool kit. As TetraHenry reports simulation results in the VTK file format it is necessary that the language of choice supports the VTK library. Of the rich variety of programming and scripting languages available, few match the design requirements. The VTK library is only supported by python and C++. Python is a high level interpreted language and is often used for scientific computing. Python is very flexible but suffers from performance issues due to it being interpreted. C++ is a lower level compiled programming language that is generally regarded as a "fast" programming language. This is largely due to the fact that it does not have the overhead that comes with an interpreted language. In order to meet performance goals C++ will be used.

3.3.2. The GMSH extraction module

The first challenge to finding the mean square flux noise figure is calculating the magnetic flux density on the surface of the SQUID washer. To do this TetraHenry requires a 2D mesh of the surface. The flux density is then calculated at each node in the mesh. Telling TetraHenry which mesh to use to calculate the magnetic flux density requires specifying the physical group of the surface mesh as well as the ".msh" file in the settings file passed to TetraHenry. The input to this module is the geometry file for the SQUID washer and the output is a new geometry file describing the geometry of only the surface of the washer. When the output geometry file is meshed with GMSH it should generate a 2D mesh of the surface.

As previously discussed, in GMSH geometry is described by a hierarchical structure. We are ignoring the use of the openCASCADE kernal as this introduces top down constructive geometry which abstracts the hierarchical structure of the typical built-in kernal representation in GMSH. As such this module will not support geometry described by openCASCADE representation. With this limitation on the structure of the input geometry file we can assume that all volumes are bounded by a set of surfaces. With this assumption the design of this module reduces to parsing and editing the geometry file in the correct manner to produce the desired output.

A surface is created in GMSH using the following command "*Plane Surface(surface tag)* = { *curveTags* };". The surface tag is a unique numerical identifier assigned to each surface. Curve tags is the comma separated list of curve identifiers that define the boundary of the surface. A volume is defined in the same way with the only difference being the volume is defined by a list of surfaces that bound the volume. The algorithm begins by simply removing all volumes and physical groups from the model. It then finds and records the tags that correspond to a surface in the geometry file. For each surface discovered it creates a new physical group and adds the surface to it. The geometry file can be user created or generated by GMSH and must therefore be flexible to things like extra spaces or new line characters in the input. To account for this regular expressions are used in conjunction with the syntax expected from a geometry file without any errors. The regular expressions and their respective use is described in table 3.1.

Regular expression	Description
/Plane.*Surface.*(.*)/gmsU	Match the definition of a plane in the geometry file and store the tag in a capture group
/Physical/gmsU	Match any string that contains the word Physical. Run line by line to find physical groups and remove them.
/Volume.*.*.*;/gmsU	Find all volumes to delete
/Mesh*.([[:digit:]]);/gmsU	Find the mesh command to replace whatever is captured in the capture group by 2
/Save.*”(.*)”.*;/gmsU	Find the name of the output mesh for replacement

Table 3.1: A table with the regular expressions used in the GMSH extraction module along with their respective function.

With the use of these regular expressions one can follow the steps below to extract the surface mesh:

1. Delete all physical groups
2. Delete all volumes
3. Find all planes
4. For each plane find the tag and add that tag to a new physical group
5. Find the mesh command and ensure that it is set to make a 2D mesh
6. Find the save command and change the output file name
7. Run the file through GMSH to generate the surface mesh

3.3.3. The mesh optimisation module

The idea behind mesh optimisation in this context is to reduce the number of mesh elements in the resulting mesh without compromising on the accuracy of the field calculations. In the work done by Anton et al. [2] an older version of InductEx was used. This version of InductEx used filaments as mesh elements. In many optimisation problems the challenge is to find a solution to said problem under some constraint. The challenge in mesh optimisation is finding this constraint. The methods employed by the authors in [2] involved finding the spline interpolation of currents between adjacent nodes. The constraint was then set such that the change in current between nodes cannot exceed a user specified threshold. If adjacent mesh elements are found where this is the case the mesh elements are divided using the spline interpolation to adhere to this constraint. The new mesh is then used to solve for the current distribution and the same process is repeated until it reaches the iteration limit or the solution converges on a stable mesh (no subdivisions are made).

The version of InductEx this project is designed to work with supports tetrahedral meshing which allows for more freedom in the geometry one can mesh (is this true). As previously mentioned, all meshing is done with GMSH and as such mesh optimisation of this nature will require some form of integration with GMSH.

In GMSH the longest side of a mesh element (triangle) is referred to as the characteristic length. The characteristic length essentially defines how finely a geometric model is meshed. The characteristic length is therefore closely related to the number of mesh elements in the resulting mesh. Clearly this is how the mesh must be optimized. The characteristic length of each mesh element must be chosen such that a finer mesh is generated where the current varies rapidly across mesh elements. GMSH does not allow for the direct specification of mesh element characteristic lengths. The characteristic length of each element is instead specified through the following methods:

1. By specifying the characteristic lengths at each point in the geometric model. If the "*Mesh.MeshSizeExtendFromBoundary*" option is set to true, the characteristic length will be interpolated between defining points, curves and surfaces.
2. By specifying the number of mesh elements per rotation.
3. By specifying mesh size fields.
4. By specifying structured meshing constraints.

Option 3 was chosen because it provides the most control over mesh sizes and is the closest to being able to specify individual mesh element sizes.

A field in GMSH refers to a post-processing view. A post-processing view can be used as a mesh size field by setting it as a background mesh. The problem reduces to finding an appropriate background mesh. There are 2 quantities that can be used to generate the background mesh. The two options are using the magnetic flux density or using the current density. I opted for using the magnetic flux density as it is associated with a 2D mesh. The 2D background mesh generated can also be used to generate a 3D mesh where the mesh sizes are automatically interpolated by GMSH.

To convert the magnitude of the magnetic flux density to a characteristic length at each node a non-linear mapping is used. A linear approximation of a sigmoid function was chosen. The motivation for choosing a sigmoid function is it allows for the user to specify minimum and maximum mesh size while still interpolating between both boundaries. The approximation is used for performance reasons. Figure 3.3 shows the sigmoid and its linear approximation. The values that define the properties of the mapping are specified by the user. The user must specify the min cut-off, max cut-off, maximum magnetic flux

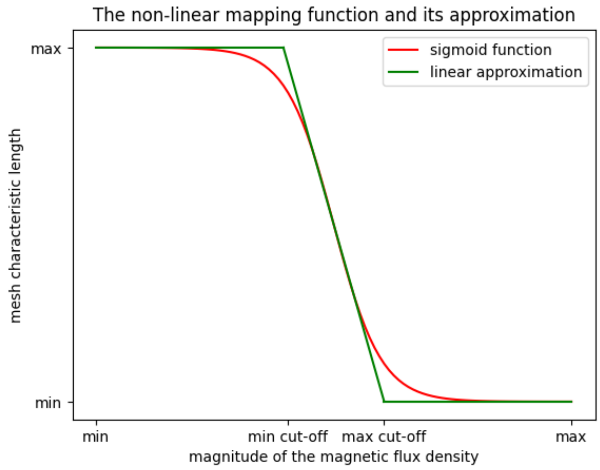

Figure 3.3: A graph showing the sigmoid function with its linear approximation

density, minimum magnetic flux density, maximum characteristic length and minimum characteristic length. The module then loops through each point in the mesh and calculates the desired characteristic length from the mapping function. The background mesh is specified as a collection of triangles. Each triangle consists of three points where the characteristic length is set for each point. The background mesh is appended to the geometry file and GMSH is run to mesh the structure.

3.3.4. The noise extraction module

This module is based on the work done by [2]. This module will be implemented as a command line tool. The module receives the path to a VTK file which defines an unstructured grid and the total current circulating in the SQUID washer as command line arguments. The unstructured grid specifies the individual mesh elements (triangles) that the surface of the SQUID washer in question consists of. Additionally, each node of each mesh element has a vector associated with it representing the magnetic flux density at the location of said node.

This module effectively implements equation 2.38 numerically. Expanding equation 2.38:

$$\langle \Phi^2 \rangle = \frac{N\mu_B^2}{3I^2} \frac{\iint \mathbf{B(r)} \cdot \mathbf{B(r)} ds}{\iint ds} \tag{3.1}$$

Noting that in equation 3.1 $N/\iint ds = \sigma$ and therefore the bottom integral can be eliminated [2]. Next we discretize the integral:

$$\langle \Phi^2 \rangle = \frac{\sigma\mu_B^2}{3I^2} \sum_{n}^{N_{\text{nodes}}} [\mathbf{B}(\mathbf{r_n}) \cdot \mathbf{B}(\mathbf{r_n})] A_n \tag{3.2}$$

In equation 3.2 A_n refers to the area associated with the n_{th} node. The method for determining A_n is discussed at a later stage in the design. Similarly, $\mathbf{r_n}$ refers to the vector position of the n_{th} node in the unstructured grid. The basic algorithm then follows as such:

Algorithm 3.1: The algorithm for evaluating the discretized integral

$N \leftarrow$ Number of points in grid
$pts \leftarrow$ list of all points
$A \leftarrow$ list of all areas
$i \leftarrow 0$
while $i < N$ **do**
  $A_n \leftarrow A[i]$
  $r_n \leftarrow pts[i]$
  $\Phi^2 \leftarrow \Phi^2 + \mathbf{B}(\mathbf{r_n}) \cdot \mathbf{B}(\mathbf{r_n}) A_n$
end while
$\Phi^2 \leftarrow \Phi^2 \cdot \frac{\sigma \mu_B^2}{3I^2}$

The Choice of $\mathbf{A_n}$

Equation 3.2 shows that the calculation requires the weighted summation of the flux density over the surface. The resulting unstructured grid generated by TetraHenry associates each flux density vector with a node in the surface mesh. Figure 3.4 shows and example of what this looks like in practice. The challenge then is finding a scheme for choosing A_n.

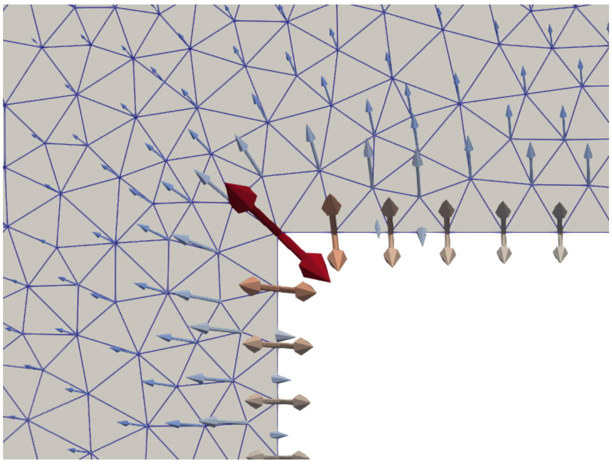

Figure 3.4: A close up of a square washer showing the field calculations at each node in the triangular mesh.

To get some idea of what A_n should look like we can investigate some of its properties. The first property of A_n is expressed in equation 3.3.

$$\text{Surface Area} = \sum_{n=0}^{N} A_n \tag{3.3}$$

The second property we would like from A_n is that it minimizes the error caused by discretization.

$$||\mathbf{B_n}A_n - \iint\limits_{r\in A_n} \mathbf{B}(\mathbf{r})ds|| = E \tag{3.4}$$

It is evident from equation 3.4 that the error is 0 when $\mathbf{B}(\mathbf{r}) = \mathbf{B_n}$. This is not usually the case. If we assume that the mesh is fine enough such that $\mathbf{B}(\mathbf{r})$ varies slowly across adjacent mesh elements, we can say that the error is roughly proportional to the distance of a point in the region defined by A_n to the mesh node to which A_n belongs to. With this in mind we construct a new optimisation problem.

Let $\mathbf{X}$ be a metric space representing the surface area of the structure. We define a set of points $\mathbf{P} \in \mathbf{X}$ such that $\mathbf{P_k}$ refers to the coordinates of the $k_{th} \in K$ node in the mesh. We would like to divide the metric space $\mathbf{X}$ into K regions such that each region is assigned the points closest to it thereby minimizing the error due to discretization. The k_{th} region is denoted as A_k. Based on these arguments we can write down an objective function for this problem. The objective function quantifies the total error due to our choice of region at point $\mathbf{x}$ defined by the function $t_k(\mathbf{x})$.

$$J = \iint\limits_{\mathbf{x}\in\mathbf{X}} \sum_{k}^{K} t_k(\mathbf{x})||\mathbf{x} - \mathbf{P_k}||^2 ds \tag{3.5}$$

check the def of k-means The problem reduces to finding the choice of $t_k(\mathbf{x})$ that minimizes J. The obvious choice for minimizing the sum is to set $t_k(\mathbf{x})$ to zero for all K regions except for the region where the distance to the point $\mathbf{P_k}$ is at a minimum. Mathematically this can be expressed by equation 3.6

$$t_k(\mathbf{x}) = \begin{cases} 1 & \text{If} \quad k = argmin_j\{||\mathbf{x} - \mathbf{P_j}||^2\} \\ 0 & \text{Everywhere else} \end{cases} \tag{3.6}$$

Recalling the definition of the Voronoi region defined in the metric space $\mathbf{X}$ with the distance function d and sites defined by the points $(\mathbf{P_k})_{k\in K}$:

$$\mathbf{A_k} = \{\mathbf{x} \in \mathbf{X} \quad | \quad d(\mathbf{x}, \mathbf{P_k}) \leq d(\mathbf{x}, \mathbf{P_j}) \forall j \neq k\} \tag{3.7}$$

If we choose the distance function in equation 3.7 to be the euclidean distance we see that the regions that minimize the objective function is the Voronoi regions.

Choosing A_n to be the Voronoi regions is a good choice not only because under the assumptions made, it minimizes the objective function but also because the triangular mesh is a Delaunay triangulation. The Delaunay triangulation is the geometric dual of the Voronoi tessellation meaning if you have the Delaunay triangulation, you have the Voronoi tessellation. This choice for A_n is therefore computationally efficient. This choice also adheres to the first property we expected of A_n as the Voronoi regions partition the metric space $\mathbf{X}$.

The mesh generated by GMSH can be used to directly compute the points that define the Voronoi regions. An example is shown in figure 3.5.

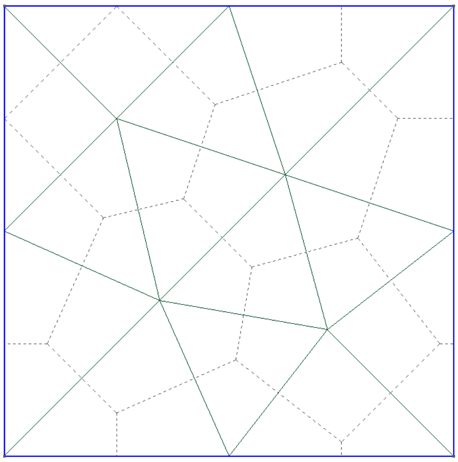

Figure 3.5: The Delaunay triangulation is shown. The triangular mesh is the solid lines. The Voronoi cells are constructed out of the dashed lines. This figure was generated with GMSH [3]

These points that define the vertices of the Voronoi cells are the circumcenters of the Delauny triangulation. The algorithm follows:

Algorithm 3.2: The algorithm for evaluating the discretized integral

```
CellList ←[ ]
for Node n in Nodes do
    C ← getPointCells(n)
    V_points ←[ ]
    for Cell c in C do
        P ← getCellPoints(c)
        cc ← computeCircumCenter(P)
        V_points.pushPoint(cc)
    end for
    CellList.pushNewCell(V_points)
end for
```

When the loop exits the *CellList* contains all the points that define the Voronoi cells. The VTK library is used to parse the simulation results. It supports optimized methods for retrieving all cells connected to a node as well as methods for fetching the points belonging to a cell. Given the position vectors P_1, P_2 and P_3 of the vertices of the triangle, the circumcenter (P_c) of a triangle in arbitrarily oriented in space can be calculated using equation 3.8.

$$P_c = \alpha P_1 + \beta P_2 + \gamma P_3 \tag{3.8}$$

where The perpendicular bisectors of the sides of a triangle intersect at the circumcenter,

$$\alpha = \frac{|P_2-P_3|^2(P_1-P_2)\cdot(P_1-P_3)}{2|(P_1-P_2)\times(P_2-P_3)|^2}$$

$$\beta = \frac{|P_1-P_3|^2(P_2-P_1)\cdot(P_2-P_3)}{2|(P_1-P_2)\times(P_2-P_3)|^2}$$

$$\gamma = \frac{|P_1-P_2|^2(P_3-P_1)\cdot(P_3-P_2)}{2|(P_1-P_2)\times(P_2-P_3)|^2}$$

so the midpoint of each side is added to the point list of a Voronoi cell. This does not influence the shape of the cell as adjacent triangles share a side and, by the definition of a perpendicular bisector, share a perpendicular bisector. The added point is therefore collinear with the line joining adjacent circumcenters. This is done to satisfy boundary conditions. The effect of this will be discussed in more detail when the ordering of points in 3 dimensions is discussed.

The shoelace algorithm

Now that an algorithm for choosing A_n has been selected the next step is to calculate the area. If we consider an ordered set of coplanar points that define a convex N-sided polygon, the surface area of this polygon can easily be calculated using the shoelace algorithm. One property of this specific Voronoi tessellation is that the regions are always defined by convex polygons making this assumption very reasonable. Equation 3.9 calculates the area given an ordered list of coplanar points $v_0, v_1, \ldots, v_N$.

$$A = \frac{1}{2}|\sum_{i=1}^{N}(v_i - v_0) \times (v_{(i+1)\%N} - v_0)| \tag{3.9}$$

The assumptions that all points in the Voronoi cell are coplanar and that all points are ordered does not hold. On curved surfaces or on surface edges there is more than 1 plane that the points in the cell belong to. Focusing on the assumption that all points in the Voronoi cell are co-planar, the algorithm must be able to group the points into sub cells that are coplanar. The assumption that the points are ordered will be addressed in the next section.

The points are read in using the VTK library triangle by triangle. Each triangular cell is read in through the ”getPointCells” method. This method returns the cells that all share a point specified by an argument passed to the method. The shared point is used as the reference point. The reference point is then subtracted from every other point in the cell. The cross product of the resulting vectors is the normal to the plane that the cell is in. Due to the non-commutative nature of the cross product the order of the vectors matter. To account for this the implementation will consider both a_n and $-a_n$ as the same plane. Where a_n refers to a unit vector normal to the plane. The last modification that must be made is the addition of the reference point to the geometry of each sub cell. The implementation will add the reference point to each sub cell if the number of unique planes that the reference point belongs to is greater than 1. Figure 3.6 demonstrates the need for this.

(a) The Voronoi tessellation with the inclusion of the reference point

(b) The Voronoi tessellation without the inclusion of the reference point

Figure 3.6: The figure shows two examples of a cube. The surfaces have been meshed into triangles. The red lines indicate the boundaries of the Voronoi cell and the shaded region shows the Voronoi cell region. The yellow dot indicates the current reference point.

The shoelace algorithm can then be performed on each individual sub cell to calculate the total area associated with a node. Figure 3.6 also shows why it is necessary to add the midpoint of each side to the Voronoi cell.

Point ordering

The points available through the ”getPointCells” method in the VTK library does not have any connectivity information. That is the points are unordered. As such an algorithm needs to be developed that can generate the connectivity information after the vertex data is retrieved. The algorithm starts by defining an orthogonal basis in the plane associated with the Voronoi cell. To do this we start by taking the mean of the points and subtracting it from every point in the list. This is done so that the plane of the points runs through the origin. The next step takes the first point in the list and normalises it. This is the first vector in the orthogonal basis. The second vector is created by taking the cross product of the first vector and the normal vector of the plane. The new vector is once again normalised. By the definition of a plane the new vector has to lie in the plane.

Now that an orthogonal basis is established we can begin to order the points. The basic idea behind the algorithm is to take the zero mean list of points and calculate the angle between the position vectors of each point and one of the vectors in the basis set. Points are then sorted in ascending order with points with the smallest angles appearing first in the sorted list. It should be noted that this process assumes the points are co-planar, so it must be performed on a per sub cell basis.

$$\theta_i = cos^{-1}(\frac{v_i \cdot a_1}{|v_i|}) \tag{3.10}$$

Equation 3.10 calculates the angle between the i_{th} point and first basis vector a_1. Because reflections of vectors across the axis of a_1 will give the same angle, there is ambiguity introduced. The sign of the dot product between each point and the second basis vector is used to resolve the ambiguity. With the help of an example the convention used is shown below in figure 3.7 and table 3.2.

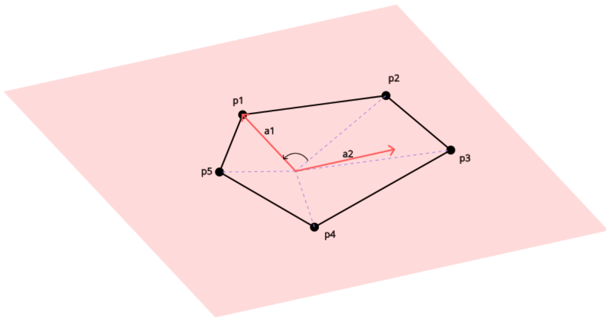

Figure 3.7: A figure showing an example Voronoi cell. Each point is labelled, and the position vectors are shown.

Point	Angle	sign($a_2 \cdot P_i$)	Final angle
p1	0	+	0
p2	60	+	60
p3	95	+	95
p4	175	-	185
p5	80	-	280

Table 3.2: The first column refers to the point label and is a reference to figure 3.7. The second column is the result of applying equation 3.10. The third column shows the sign of the projection of vector P_i onto the second basis vector. The fourth column shows the angle after adjustment. The fourth column is calculated as: $360 - Angle$.

Chapter 4

Results

Chapter 5

Summary and Conclusion

Bibliography

[1] I. Sochnikov, D. Davino, and B. Kalisky, “dc squid design with femtotesla sensitivity for quantum-ready readouts,” *Physical review applied*, vol. 14, no. 1, 2020.

[2] S. M. Anton, I. A. Sognnaes, J. S. Birenbaum, S. R. O’Kelley, C. J. Fourie, and J. Clarke, “Mean square flux noise in squids and qubits: Numerical calculations,” *Superconductor Science and Technology*, vol. 26, no. 7, p. 075022, 2013.

[3] C. Geuzaine and J.-F. Remacle, “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities,” *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[4] A. A. Golubov, *The Evolution of Superconducting theories.* Institute of Physics Pub., 1998, pp. 3–37.

[5] M. Tinkham, *Historical Overview*, 2nd ed. Dover Publications, 2015, pp. 4–5.

[6] R. P. Feynman, R. B. Leighton, and M. L. Sands, *The Schrödinger Equation in a Classical Context: A Seminar on Superconductivity.* Pearson India Education Services Pvt Ltd, 2013, vol. 3.

[7] T. V. Duzer, *Chapeter 4: Josephsons Junctions.* Prentice Hall, 1999, pp. 158–218.

[8] A. A. Golubov, M. Y. Kupriyanov, and E. Il’ichev, “The current-phase relation in josephson junctions,” *Reviews of Modern Physics*, vol. 76, no. 2, pp. 411–469, 2004.

[9] R. C. Jaklevic, J. Lambe, A. H. Silver, and J. E. Mercereau, “Quantum interference effects in josephson tunneling,” *Phys. Rev. Lett.*, vol. 12, pp. 159–160, Feb 1964. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.12.159

[10] D. Drung, C. Abmann, J. Beyer, A. Kirste, M. Peters, F. Ruede, and T. Schurig, “Highly sensitive and easy-to-use squid sensors,” *IEEE Transactions on Applied Superconductivity*, vol. 17, no. 2, pp. 699–704, 2007.

[11] B. Josephson, “Possible new effects in superconductive tunnelling,” *Physics Letters*, vol. 1, no. 7, pp. 251–253, 1962. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0031916362913690

[12] J. Clarke and A. I. A. I. Braginski, *The SQUID handbook.* Weinheim, [Cambridge: Wiley-VCH, 2004 - 2006.

[13] M. Faley, U. Poppe, K. Urban, D. Paulson, T. Starr, and R. Fagaly, “Low noise hts dc-squid flip-chip magnetometers and gradiometers,” *IEEE Transactions on Applied Superconductivity*, vol. 11, no. 1, pp. 1383–1386, 2001.

[14] J. Braumüller, L. Ding, A. P. Vepsäläinen, Y. Sung, M. Kjaergaard, T. Menke, R. Winik, D. Kim, B. M. Niedzielski, A. Melville, J. L. Yoder, C. F. Hirjibehedin, T. P. Orlando, S. Gustavsson, and W. D. Oliver, “Characterizing and optimizing qubit coherence based on squid geometry,” *Physical review applied*, vol. 13, no. 5, 2020.

[15] F. C. Wellstood, C. Urbina, and J. Clarke, “Role of geometry on the color of flux noise in dc squids,” *IEEE transactions on applied superconductivity*, vol. 21, no. 3, pp. 856–859, 2011.

[16] H. Wang, C. Shi, J. Hu, S. Han, C. C. Yu, and R. Wu, “Candidate source of flux noise in squids: Adsorbed oxygen molecules,” *Physical review letters*, vol. 115, no. 7, pp. 077 002–077 002, 2015.

[17] P. Kumar, S. Sendelbach, M. Beck, J. Freeland, Z. Wang, H. Wang, C. C. Yu, R. Wu, D. Pappas, and R. McDermott, “Origin and reduction of 1 / f magnetic flux noise in superconducting devices,” *Physical review applied*, vol. 6, no. 4, 2016.

[18] R. C. Bialczak, R. McDermott, M. Ansmann, M. Hofheinz, N. Katz, E. Lucero, M. Neeley, A. D. O’Connell, H. Wang, A. N. Cleland, and J. M. Martinis, “1/f flux noise in josephson phase qubits,” *Physical review letters*, vol. 99, no. 18, pp. 187 006–187 006, 2007.

[19] R. H. Koch, D. P. Divincenzo, and J. Clarke, “Model for 1/f flux noise in squids and qubits,” *Physical review letters*, vol. 98, no. 26, pp. 267 003–267 003, 2007.

[20] L. V. Abdurakhimov, I. Mahboob, H. Toida, K. Kakuyanagi, Y. Matsuzaki, and S. Saito, “Identification of different types of high-frequency defects in superconducting qubits,” *PRX quantum*, vol. 3, no. 4, pp. 040 332–, 2022.

Appendix A

Project Planning Schedule

This is an appendix.

Appendix B

Outcomes Compliance

This is another appendix.