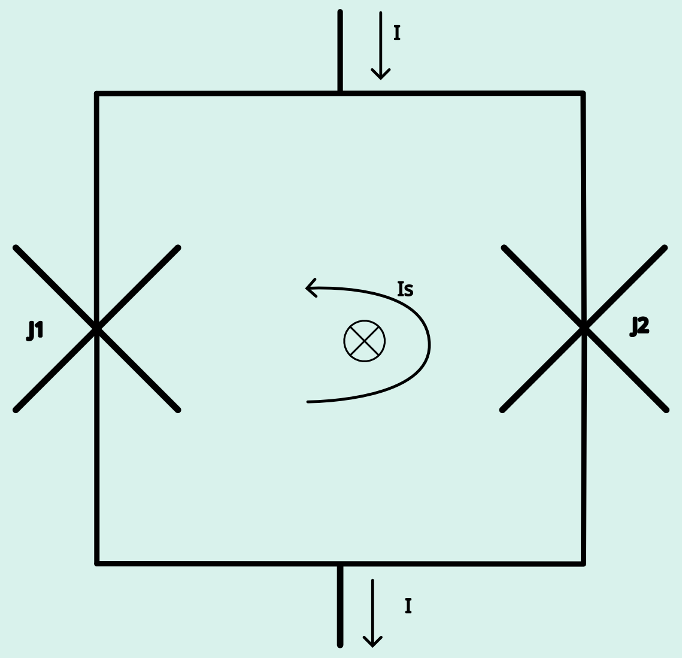
Development of a tool for the extraction of low frequencynoise power in SQUIDs

Paul Rossouw - 23572027

Background

What is a SQUID?

A DC superconducting quantum interference device (SQUID) consists of a superconducting loop interrupted by two Josephson junctions. When an external field is applied to the loop a screening current flows. This screening current modulates the crictical current of each junction. The voltage on accross the junctions is modulated in the same way through the josephson effect. The voltage is modulated by a signal that has a period of 1 flux quantum.



The importance of flux noise

In the field of superconducting electronics, the presence of excess low-frequency flux noise has puzzled many. The importance of designing for low-frequency flux noise cannot be understated. In applications such as biomagnetism and quantum computing the low-frequency flux noise has been identified as a limiting factor in the performance of the relevant superconducting electronics.

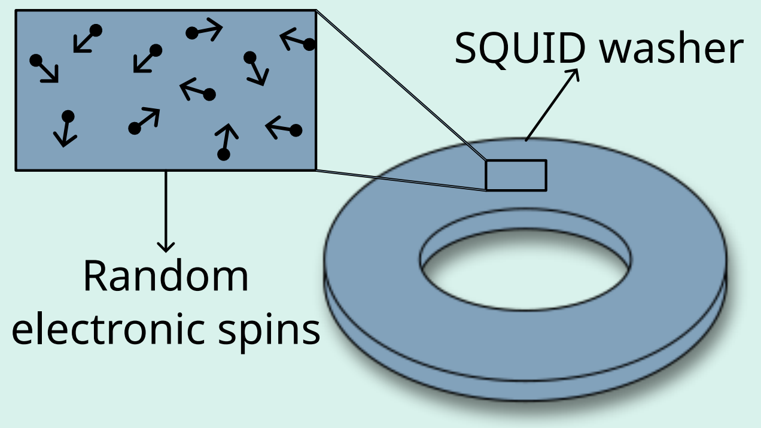
Project Scope

This project covered the design and implementation of a numerical framework for predicting mean square flux noise figures in SQUID sensors. The project will restrict its scope to the DC SQUID.

Theoretical Background

Problem:

How can the total effect of a number of randomly reversing magnetic moments on the flux in the loop be calculated?



Solution:

The large number of spins requires the use of the principle of reciprocity. The problem reduces to finding the B-field at the location of each spin.

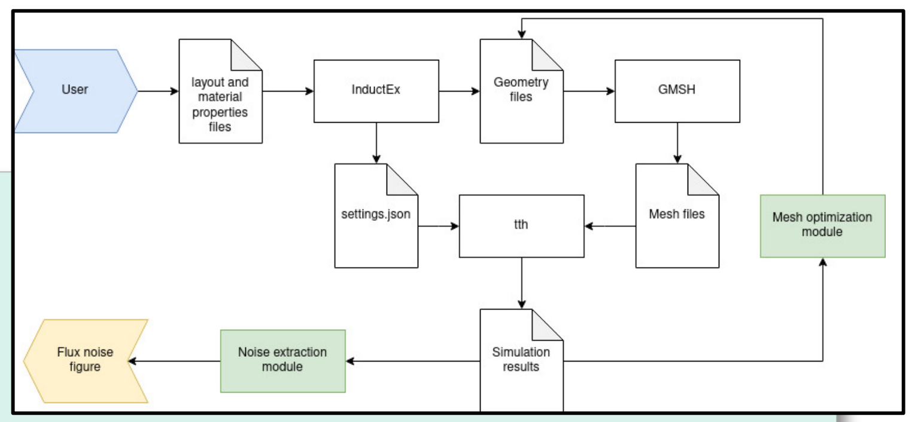
Flux due to a single spin:

$$\Phi = \vec{\mu_B} \cdot \frac{B(\vec{r})}{I}$$

Flux due to a collection of spins:

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

The noise extraction module



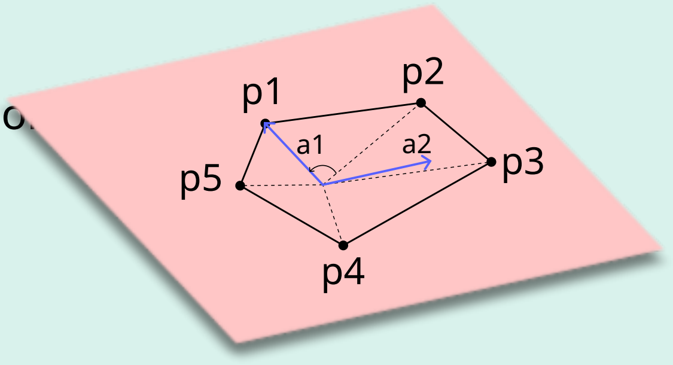
High level design

The figure above shows how the developed modules interact with InductEx and TetraHenry. TetraHenry is the electromagnetic solver used to compute the B-field. The noise extraction module is implemented as a command line tool that receives the VTK file that TetraHenry generates. It implements the equation below.

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

Point ordering

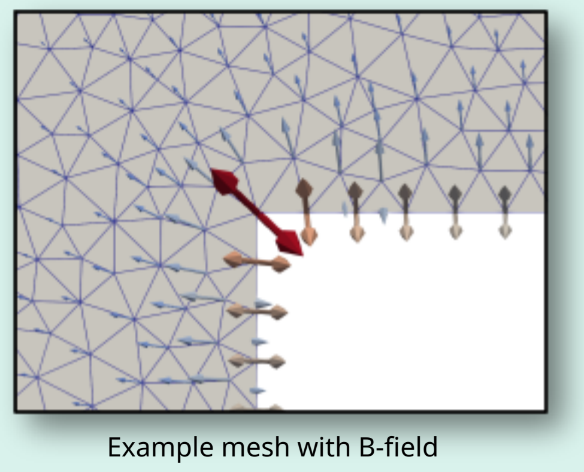
The area of each region is co shoelace algorithm. It assumes coplanar and ordered points. To order the points, an orthogonal basis is construct for each plane connected to a mesh node. The basis is used to compute an angle to sort the points by.



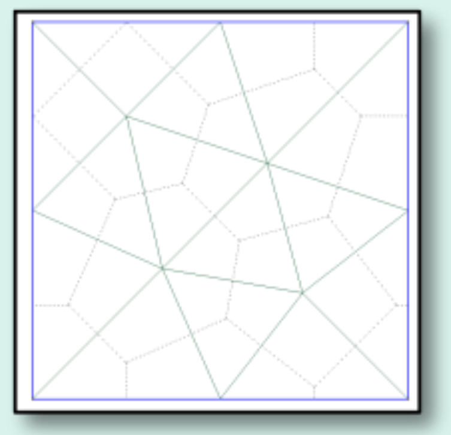
Selection of An

The adjacent figure shows an example mesh with vectors showing the B-field calculated at each mesh node. A suitable partitioning algorithm must be selected. The above equation assumes that the B-field is constant over the region. The chosen algorithm must minimize the error due to this assumption. The objective function is minimized when a Voronoi tesselation is selected.

$$J = \iint_{\vec{x} \in \vec{X}} \sum_{k}^{K} t_k(\vec{x}) ||\vec{x} - \vec{P_k}||^2 ds$$



Example mesh with B-field



Example Voronoi tesselation

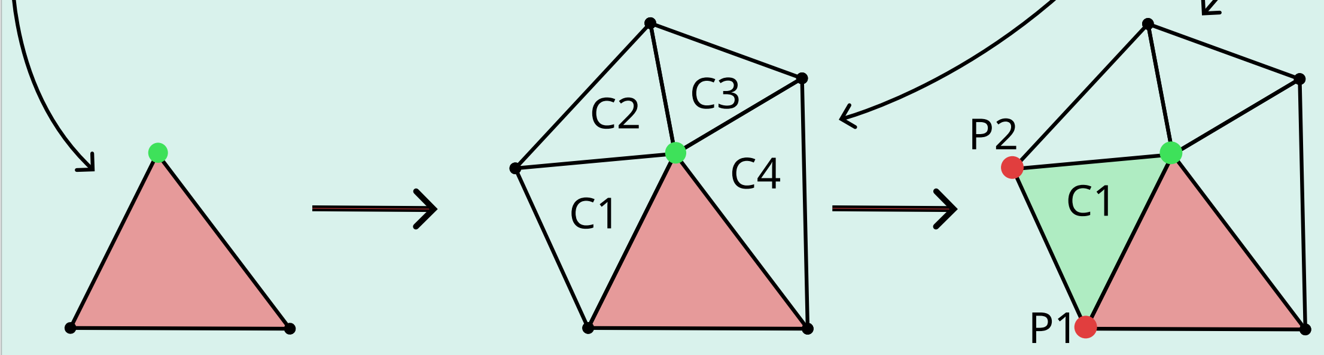
The mesh optimisation module

The need for mesh optimisation:

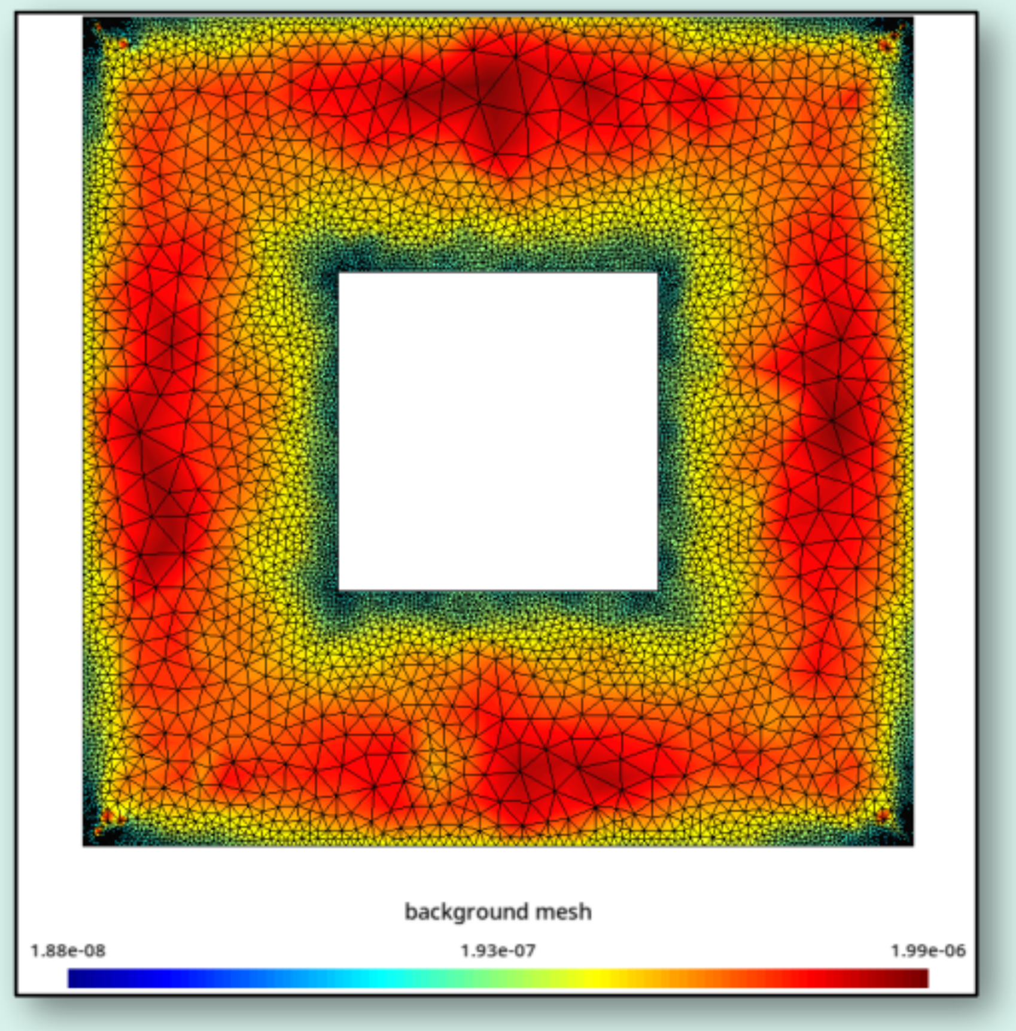
The MSFN figure is sensitive to error in the B-field. To reduce the error one must increase the number of elements in the mesh. Computational complexity scales roughly with N³ eliminating this option.

The mesh optimisation algorithm:

The mesh optimisation module receives a GMSH ".geo" file and a VTK file containing the current distribution generated by TetraHenry. Additional command line parameters specify user constraints such as the minimum relative change. The algorithm begins by looping through each cell in the mesh. Next it iterates through every point (p) in the cell. The algorithm loops through every cell connected to point p. For each connected cell, the relative change in the current density is calculated for each point in said cell. The process repeats for every cell in the mesh.

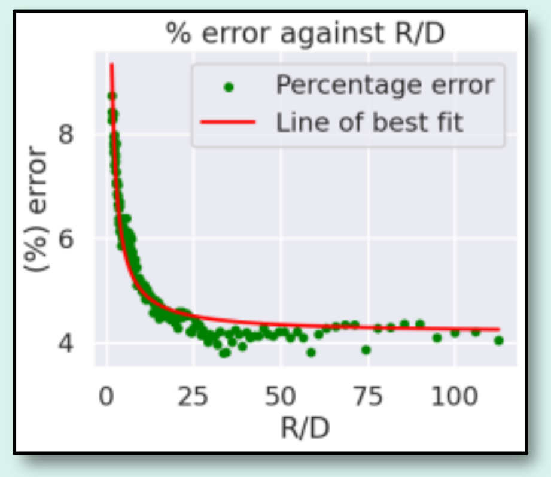


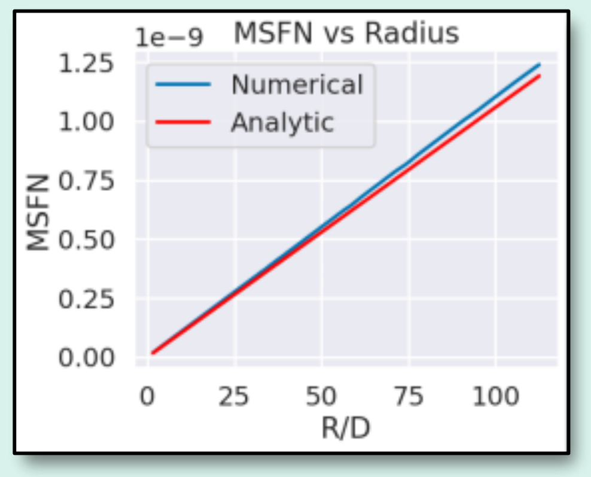
Example of an optimised mesh



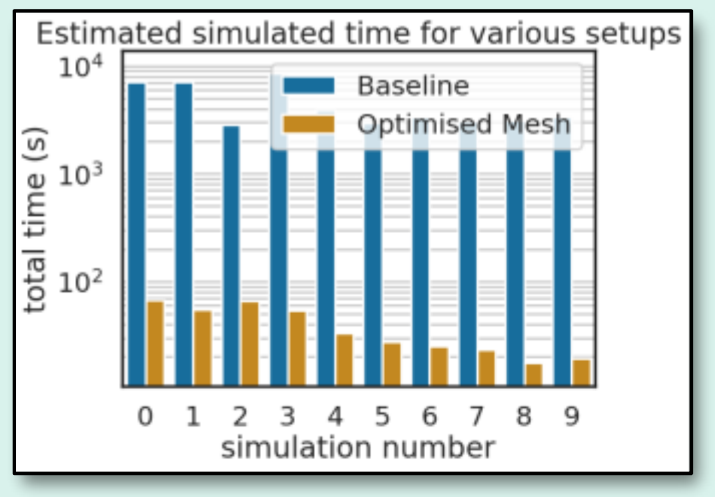
Results

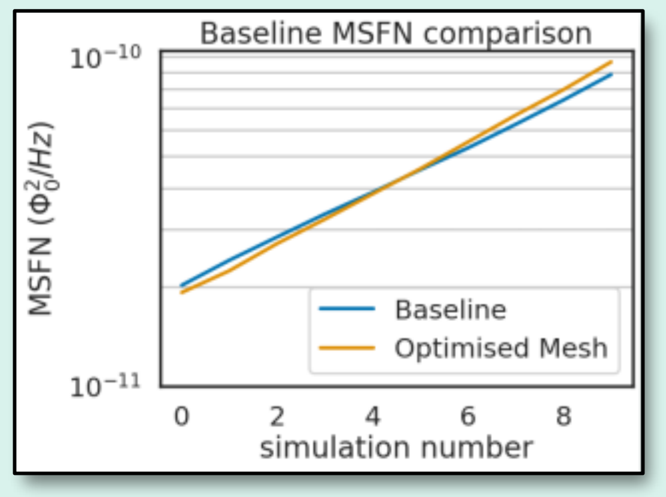
Noise extraction module results:





Mesh optimisation results:





Comparison with real SQUID designs:

SQUID Number	R	W	MSFN (measured) ($\frac{n\Phi_0^2}{Hz}$)	MSFN (numerical) ($\frac{n\Phi_0^2}{Hz}$)
I.1	12	0.5	4.5	111.924000
I.2	6	0.5	3	88.018600
I.3	3	0.5	1.2	0.121771
II.5	40	15	1	0.110498
II.1	265	240	2	0.098736
II.3	85	60	0.8	0.043155

Conclusion

Conclusion:

Future work: