

Semesterarbeit

Wilke Grosche
ETH Zurich

Supervisor: Solange Emmenegger

July 2020

Abstract

The controlled creation of magnetic fields is of relevance especially in the shielding of magnetic fields. This can be useful in experiments that require a highly homogeneous magnetic field for the purposes of heightened precision. The motivation for this work is the field stabilisation for the neutron electric dipole measurement being conducted at Paul Scherrer Institute (PSI). A novel approach to coil design allows for a controlled creation of a magnetic field within a large ($125m^3$) area to a field homogeneity of $\pm 1\%$. The goal is to use this system as an active magnetic shield to reduce and homogenise fields from outside the system. In this work the focus is on optimising the positions of the magnetic sensors. These are the points at which the field is measured before these values are fed back into a feedback algorithm so that they may be used to compensate the background field. This optimisation is done via a gradient descent using the simultaneous perturbation stochastic approximation gradient descent. Through this methodology viable fluxgate positions were found for both the proposed coil system at PSI and a prototype.

Contents

1	Introduction	3
2	Theory	4
2.1	nEDM Measurement	4
2.2	Magnetic Shielding	4
2.3	Feedback Algorithm	5
3	Method	6
3.1	Field Generation:	7
3.2	COMSOL Modelling	7
3.3	Optimisation	8
3.4	Comparison of Optimisation Methods	8
3.4.1	Approximate Gradient Methods	9
3.4.2	Simultaneous Perturbation Stochastic Approximation (SPSA)	11
3.5	Random Search	11
4	Results and Discussion	13
4.1	Convergence of the Algorithms	13
4.2	Number of Sensors	14
4.3	Refinements on Previously Calculated Positions	15
4.4	Refinements on Corner Starting Positions	16
4.5	Random Structure Search	17
4.6	Scaling	17
4.7	n2EDM Measurement	18
4.8	Validity of the Final Positions	19
4.9	Stability of the Final Positions	20
4.10	Improvements	21
5	Conclusion	21
A	Prototype Results	24
B	n2EDM Results	26

1 Introduction

The standard model is still unable to explain the matter antimatter asymmetry in the universe. A metric used for this disparity is $\eta = \frac{n_b - n_{\bar{b}}}{n_\gamma}$, here $n_b - n_{\bar{b}}$ is the difference in the Baryon and Anti-Baryon densities and n_γ is the photon number density. Whilst the established CP violation mechanisms are capable of explaining the disparity to some extent, we can only explain an asymmetry up to $\eta = 10^{-18}$, however, in reality we observe $\eta = 10^{-10}$ ([3]). This means that we require further sources of CP violation in order to explain the observed physics. One possible source of such violation is the neutron electric dipole moment (nEDM).

The Hamiltonian of a non-relativistic particle in an electromagnetic field is given by:

$$H = -\mu \vec{B} \cdot \frac{\vec{S}}{S} - d \vec{E} \cdot \frac{\vec{S}}{S} = -2(\mu \vec{B} + d \vec{E}) \cdot \vec{S} \quad (1)$$

Where μ and d are the magnetic moment and electric moment respectively. The electric field is \vec{E} and the magnetic field \vec{B} . The spin of the particle in question is given by \vec{S} . The second equality holds for spin $S = 1/2$ particles. A non-zero electric dipole moment d causes the Hamiltonian to be both P and T violating. This makes the nEDM a good probe for physics beyond the standard model. Here we will focus on the experimental setup of the nEDM measurement at Paul Scherrer Institute (PSI). The work will build upon the active magnetic shielding developed by Beatrice Franke[7] and Michal Rawlik et al. ([9]). nEDM measurements lose their applicability if the uncertainty of the experiment is greater than the absolute value of the dipole moment being measured. As can be seen in Fig. 1 the uncertainty in the measurement is not yet low enough to exclude physics beyond the standard model.

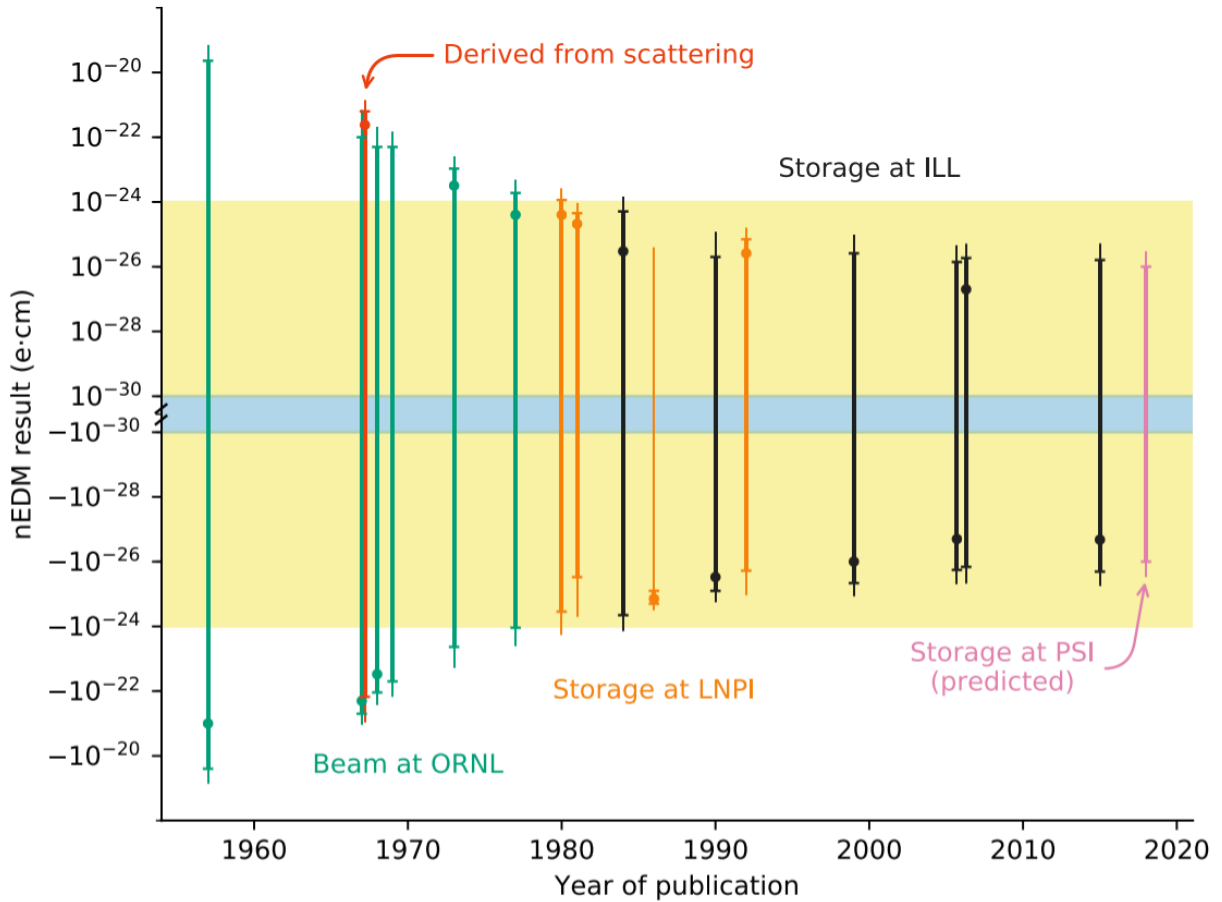


Figure 1: The uncertainty in the nEDM for different measurements over time. Depicted are three standard deviations of the measurements against the year of publication. The blue zone represents the predicted values for the nEDM in the standard model whilst the yellow zone is for extensions to the standard model.[9]

The standard model predicts values for the nEDM of the order $10^{-32}e$ cm. Measurements of the nEDM with sufficient precision that point to values above this threshold would suggest physics beyond the standard model. Such as those predicted by Supersymmetry. As shown in Fig. 1 the upper bound for the nEDM has been decreasing but is still far away from securing the nEDM to be of the order predicted in the standard model. The aim of the n2EDM experiment at PSI is to push this boundary further by increasing the precision of the measurement by an order of magnitude.

2 Theory

In order to increase the precision of the n2EDM experiment at PSI we must first understand what is limiting the precision of the measurement. In order to do so we will examine the method being used.

2.1 nEDM Measurement

The n2EDM experiment at PSI relies on the Ramsey method. This works by using the differences in the aligned and unaligned Larmor precession frequencies. Looking at Eq. 1 we can see that the energy of the particle depends on the alignments of the \vec{E} and \vec{B} fields relative to the spin state. Larmor precession occurs when the magnetic field is not aligned with the particle spin. The frequency of this precession is given by the energy difference between the spin up/down states. From Eq. 1 we find that the energies are given by:

$$h\nu_{\uparrow\uparrow} = -2(\mu B_{\uparrow\uparrow} + d_n E_{\uparrow\uparrow}) \quad (2)$$

For parallel spins. And:

$$h\nu_{\uparrow\downarrow} = -2(\mu B_{\uparrow\downarrow} - d_n E_{\uparrow\downarrow}) \quad (3)$$

For anti-parallel spins. The energy difference between the two states is what matters for the dipole measurement. Subtracting Eq. 3 from 2 and solving for d_n gives the electric dipole moment in terms of the Larmor frequencies (Eq. 4).

$$d_n = \frac{h(\nu_{\uparrow\downarrow} - \nu_{\uparrow\uparrow}) - 2\mu(B_{\uparrow\uparrow} - B_{\uparrow\downarrow})}{2(E_{\uparrow\uparrow} + E_{\uparrow\downarrow})} \quad (4)$$

This measurement of the nEDM is often limited in its precision by the term $(B_{\uparrow\uparrow} - B_{\uparrow\downarrow})$. This is because this term is susceptible to fluctuations in the magnetic field. If the field changes between two measurements this can be misunderstood as an electric dipole moment. As such it is important to eliminate these effects. These fluctuations are harder to monitor and compensate for than those in the electric field.

2.2 Magnetic Shielding

The established way to shield magnetic fields is through the use of passive magnetic shielding in the form of materials with a high magnetic permeability. For this purpose mu-metals were developed. These are nickel-iron soft ferromagnetic alloys with a permeability of order 10^4 . The high permeability allows the magnetic fields to follow a path within the material instead of penetrating into the enclosed area. However, the mu-metal shield struggles at shielding slowly oscillating fields and for large field changes the shield can become magnetised. For this reason active magnetic shielding was proposed as a method to damp the external field and slow fluctuations in the background field. The idea is to create a coil system capable of generating a desired field within an expanded region. The coil system, developed by the ETH group and detailed in Michal Rawlik's PhD thesis ([9]), is able to create a stable field in an expanded region ($125m^3$ with a field homogeneity of $\pm 1\%$). The coil system being developed at PSI is shown in Fig. 2. The coil system is constructed in 8 coils as described in chapter 4 ("Coil Design") of [9]. Fig. 3 shows the field for the homogeneous y coil based on the complexity of the current decomposition in the prototype active magnetic shield (AMS). From the figure we can see that this decomposition is good at generating a uniform field at $50\mu T$ over the entire y position.

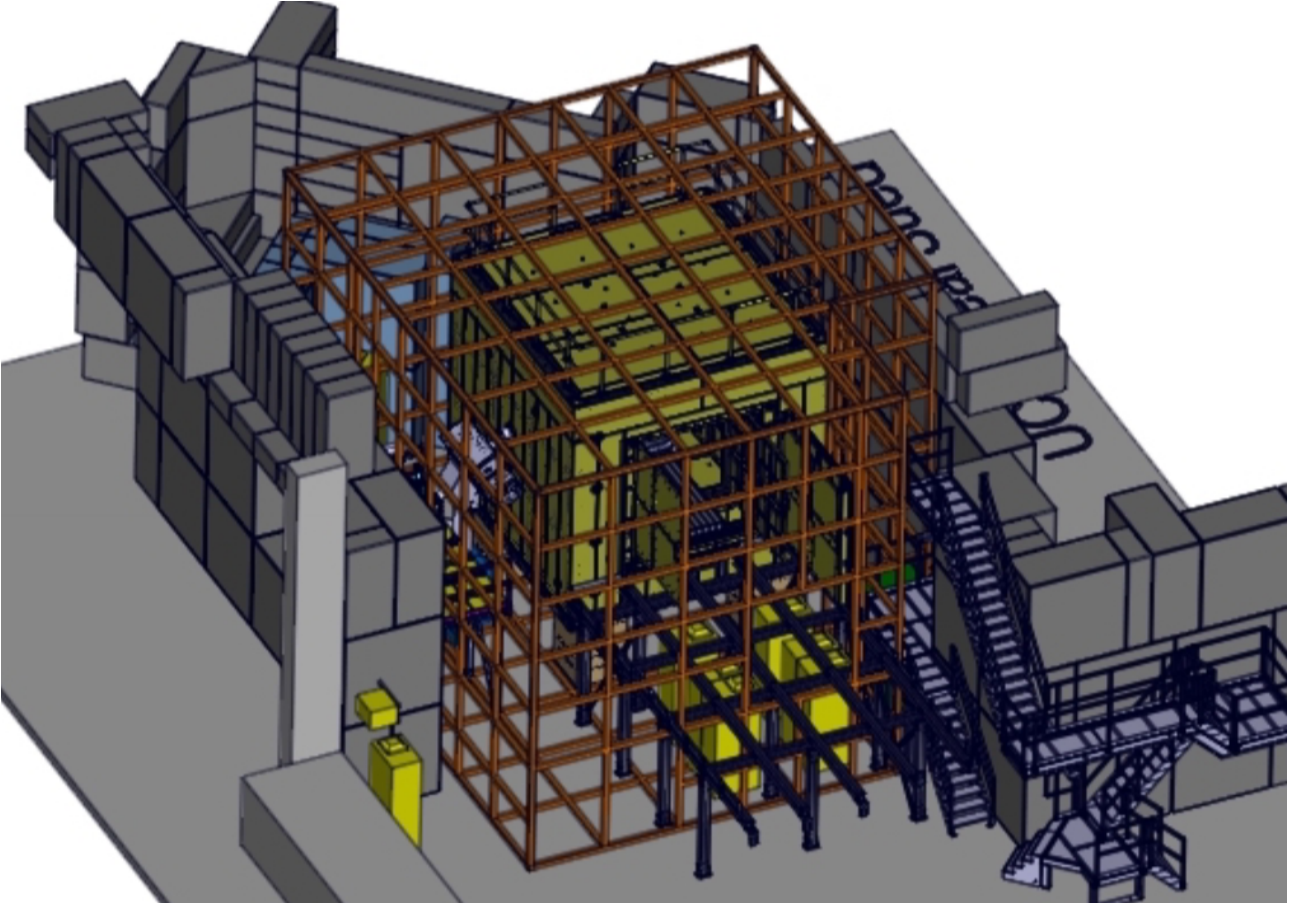


Figure 2: A CAD rendering of the coil system being developed at PSI for the purposes of active field compensation. The coil system is in orange. The passive mu-metal shield is depicted in yellow. [9]

The idea is to use this coil system in a manner similar to noise cancelling headphones, to generate a field that exactly compensates the background field at each point in the measurement zone. In order to have this happen in real time a feedback algorithm was developed [7].

2.3 Feedback Algorithm

In order to facilitate the real time field compensation, the field has to be measured at fixed points. Since the field generated by the coil system is given by the Biot Savart law, a static correlation matrix \mathbb{M} can be calculated which links the field at each measurement point \vec{B}_{tot} to the current (\vec{I}) required to generate a compensating field, $\vec{B}_{tot} = \mathbb{M}\vec{I} + \vec{B}_{offset}$. Here \vec{B}_{offset} is the background field that is to be compensated.

The matrix is calculated by taking the field at different currents at the different measurement points and calculating the gradient. Since the relationship between the currents and the field is linear for each coil, the components are simply given by the slope of $\vec{B}(\vec{I}_i)$ ($i \in \{x, y, z\}$).

The field generated by the shield is given by $\vec{B} = \mathbb{M}\vec{I}$. Here the currents through the generating coils are gathered into a vector in $\mathbb{R}^{n_{coils}}$ where n_{coils} is the number of coils. The correlation matrix \mathbb{M} is in $\mathbb{R}^{(n_{coils} \times 3n_{sensors})}$.

$$M_{i,j} = \begin{pmatrix} P1_{x,1} & P2_{x,1} & \cdots & P8_{x,1} \\ P1_{y,1} & P2_{y,1} & \cdots & P8_{y,1} \\ \vdots & \vdots & \ddots & \vdots \\ P1_{z,n} & P2_{z,n} & \cdots & P8_{z,n} \end{pmatrix} \quad (5)$$

This feedback matrix is static in the external fields as it only depends on the positions of the measurement points at which it is calculated. At each measurement point we will place a fluxgate.

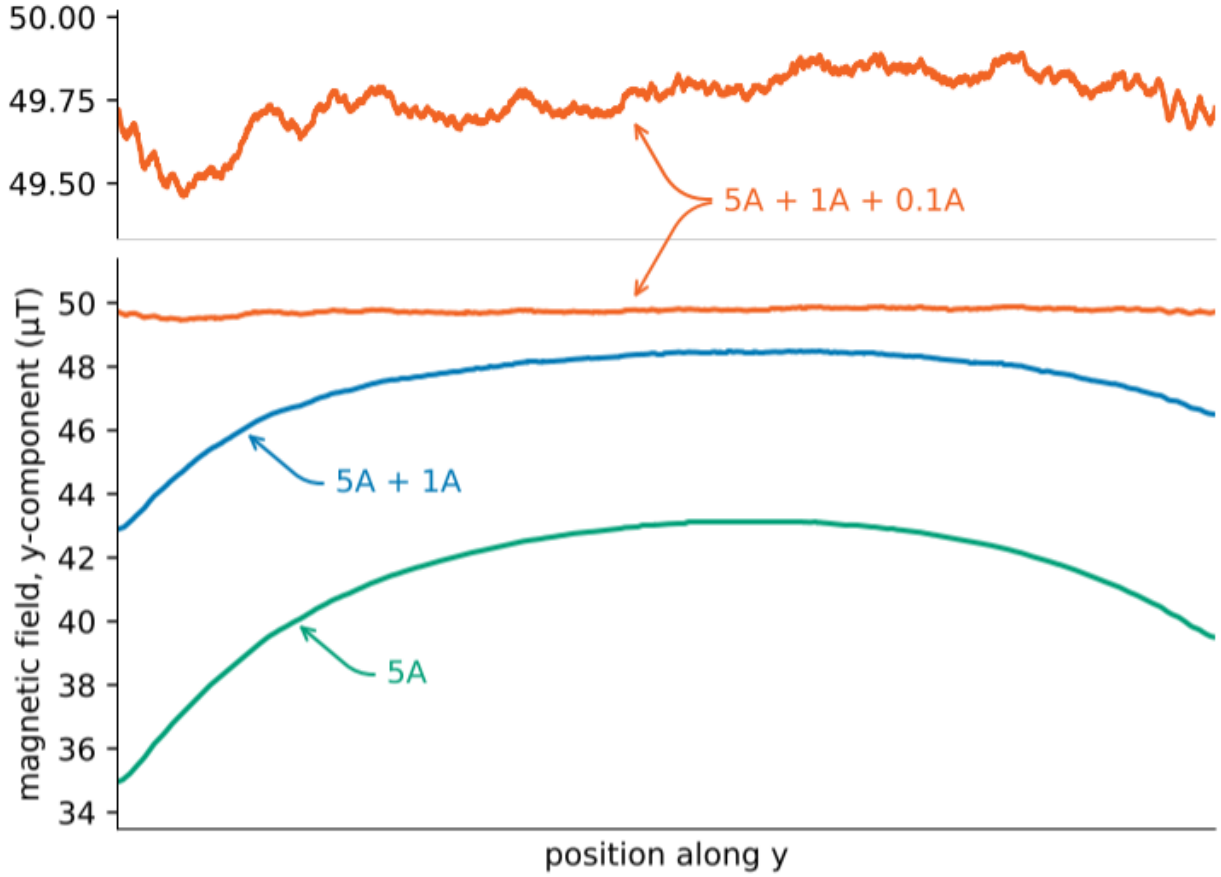


Figure 3: Linear map of the P2 coil (homogeneous y coil). Shown is the field through the middle of the x-z plane. As a function of the y position the green curve shows the field due to the 5A loop. In blue the combination of the 5A and 1A loops is shown and the orange curve shows all 3 current loops (5A+1A+0.1A). A zoomed in version of the full coil field is shown above the plot. [9]

This is a magnetic field sensor with which the amplitude and direction of magnetic field at that point can be measured. The danger with this matrix is that, with certain fluxgate positions, the matrix can become ill conditioned. This means that the matrix has a very high condition number. The condition number is a measure for how sensitive ([4]) the output of a function is to small changes in the input. In the case of a matrix equation the condition number of a matrix is given by:

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} \quad (6)$$

Where $\sigma_{max}(A)$ and $\sigma_{min}(A)$ are the maximal and minimal singular values of A respectively. A high condition number in the feedback matrix corresponds to unequal coupling of the fields generated by the different coils at one or more of the points at which sensors are placed. This is a problem because it could result in the generated fields being dominated by the contribution due to one over-sensitive measurement. This would lead to high instability in the generated field and an unreliable field compensation. As a result the condition number of the feedback matrix is to be minimised.

3 Method

In order to determine the best positions for the sensors the system is simulated and a gradient descent on the condition number is performed. For this to work the fields generated by the coil system have to be calculated.

3.1 Field Generation:

The fields generated by the active magnetic shielding (AMS) system can be modelled computationally via the use of the Biot Savart law:

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \int_C \frac{I d\vec{l} \times \vec{r}'}{|\vec{r}'|^3} \quad (7)$$

For this purpose Michal Rawlik developed the Coils Package for Julia ([10]). This allows the generation of a system comprising line elements which carry a current. The coil systems can be read in and the currents required to generate a desired field are calculated. Since we need an orthogonal basis for \mathbb{M} , the nEDM coil system is composed of 8 different coils each generating a Cartesian harmonic in the polynomial decomposition of the magnetic field:

$$\vec{B}(\vec{r}) = \sum_n H_n \vec{P}_n(\vec{r}) \approx \sum_{n=1}^8 H_n \vec{P}_n \quad (8)$$

In order to achieve the desired homogeneity in the field of $\pm 1\%$ in the measurement volume we only require fields up to those of linear order ([9]).

$$P_1(\vec{r}) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, P_2(\vec{r}) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, P_3(\vec{r}) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (9)$$

$$P_4(\vec{r}) = \begin{pmatrix} x \\ 0 \\ -z \end{pmatrix}, P_5(\vec{r}) = \begin{pmatrix} y \\ x \\ 0 \end{pmatrix}, P_6(\vec{r}) = \begin{pmatrix} 0 \\ y \\ -z \end{pmatrix}, P_7(\vec{r}) = \begin{pmatrix} z \\ 0 \\ x \end{pmatrix}, P_8(\vec{r}) = \begin{pmatrix} 0 \\ z \\ y \end{pmatrix} \quad (10)$$

Each field (P_1 to P_8) is generated independently and the matrix 5 can be determined.

3.2 COMSOL Modelling

The fields generated in Julia are for a system without a mu-metal cube. For this reason a further simulation step is required. The fields generated in Julia are read into COMSOL and linearly interpolated. The fields are set to 0 outside of the modelled domain. This can be done because the fields outside of the AMS have no bearing on the fields inside and thus do not change the optimal fluxgate positions.

The system is set to target a goal field of $50 \mu\text{T}$ inside the measurement zone. As can be seen in Fig. 4 the field being read in is at around $50 \mu\text{T}$ in the measurement zone.

The mu-metal shield is modelled in the form of a hollow cube surrounded by a spherical region filled with air. The boundary conditions are enforced via the use of an infinite element domain which sets the field as 0 at infinity. This can be seen in Fig. 5.

The field read into COMSOL is divided by the vacuum permeability and set as the reduced background field for the "magnetic fields, no currents" module. The resulting field with the mu-metal cube is calculated and shown in Fig. 6. From the image it becomes clear that the mu-metal cube has an effect outside of the cube itself. The P_1 field, which runs in the positive x-direction, is pulled into the high permeability material and routed through to the other side. This results in the field at the top and bottom sides of the cube being lower than at the left and right hand side, as seen in Fig. 6.

The calculated field is read out at the Lagrange points determined by COMSOL. The resulting field configuration is read into Julia. In order for the field to be continuous the matrix is linearly interpolated using scipy's NDinterpolator in a PyCall wrapper. The optimisation is applied to the continuous field in order to find the fluxgate positions with minimal condition number in the coupling matrix.

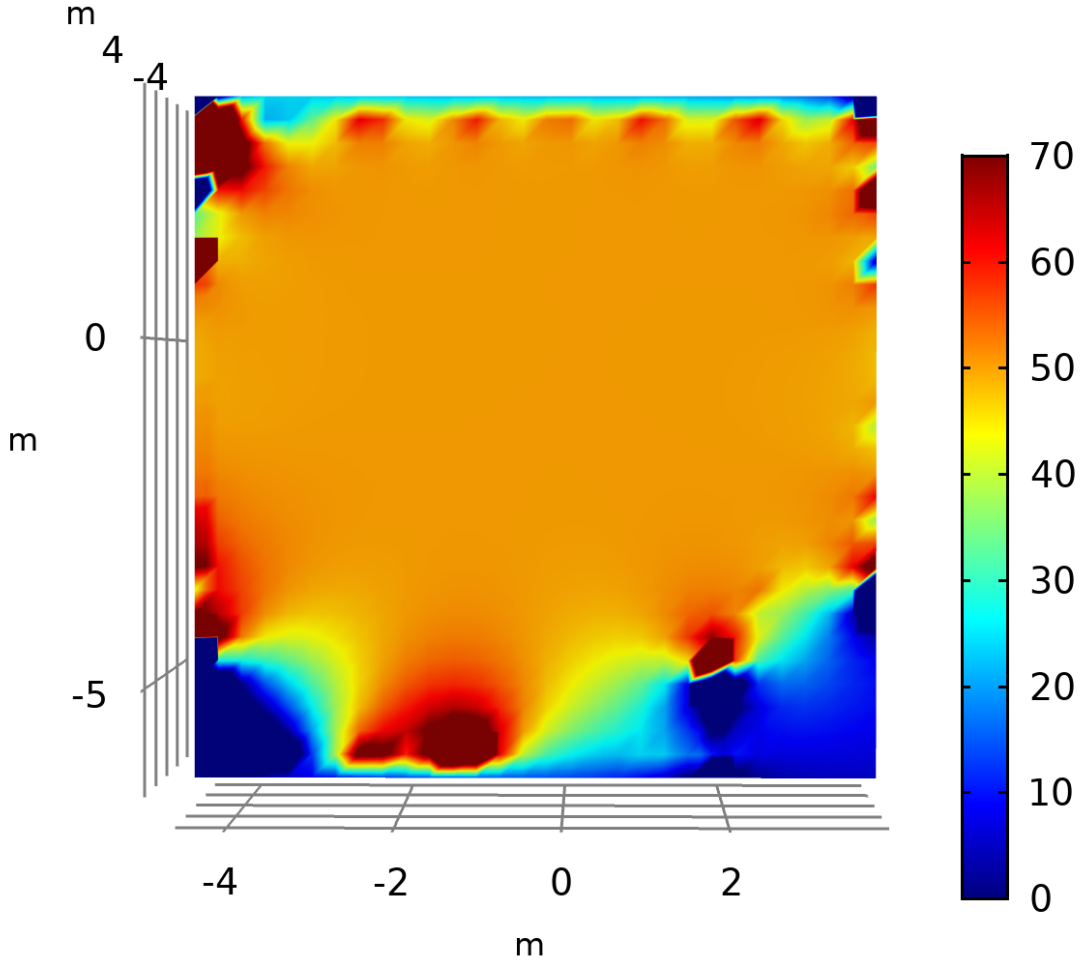


Figure 4: The x-component of the P1 Field for the n2EDM experiment. Shown is the x-y plane for an offset of $z=0$. The field at the centre is at $50 \mu\text{T}$.

3.3 Optimisation

The feedback algorithm requires the magnetic field to be measured at given points in the region of interest in order to work. The danger with placing these measurement points randomly is that this may lead to a high condition number of the feedback matrix. The optimisation minimises the condition number of the feedback matrix.

This optimisation takes place in a $24 = 3 \times n_{sensors}$ dimensional space as each sensor takes field components for all three spatial directions. Since the calculated fields are read out on discrete points and the condition number itself is not easily differentiable a simple gradient descent is not possible. This lead to the consideration of the following techniques:

- Genetic algorithm
- Nelder Mead
- SPSA
- Approximate gradient methods

3.4 Comparison of Optimisation Methods

Both the Nelder Mead optimisation method and the Genetic algorithm were discarded as options for this optimisation. The Nelder Mead algorithm was discarded due to its poor convergence in

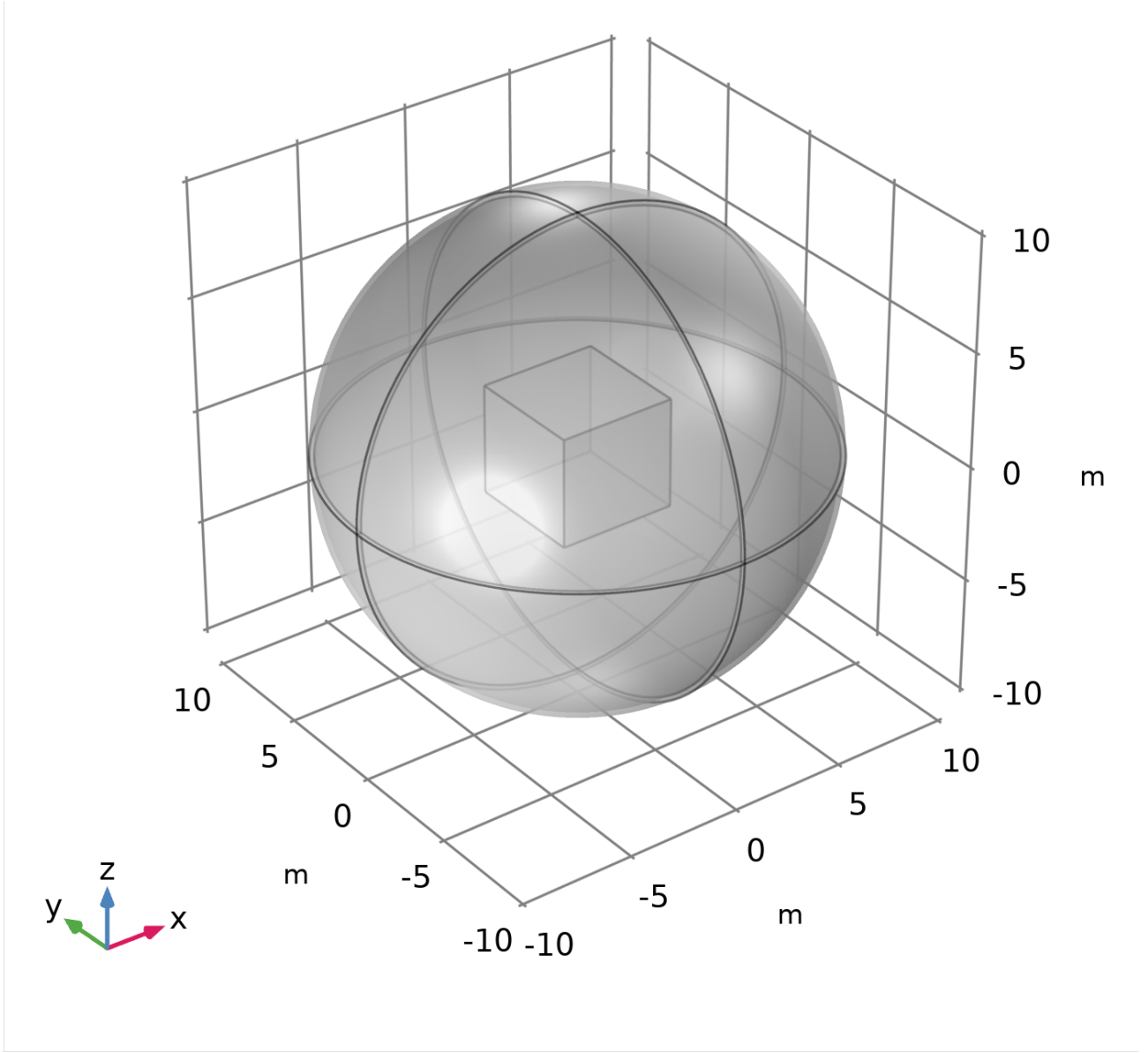


Figure 5: The modelled components in COMSOL. The hollow mu-metal cube is surrounded by a spherical region of air. The sphere’s outer layer constitutes an infinite element domain. The cube thickness is set to 1cm.

higher dimensions ($d > 10$)[5]. The genetic algorithm was discarded because it requires a ”genetic representation” of the solution domain in order for the evolution of the algorithm to take place. What is required is a method by which we can generate the second generation of fluxgate positions as a function of the first, whilst considering crossover and mutations. Crossover requires interplay of multiple configurations. The difficulty lay in translating these two concepts to our application.

3.4.1 Approximate Gradient Methods

The approximate gradient methods are of use in this situation because the loss function of the problem is not easily differentiable. This means that, in order to use gradient descent, the gradient must be estimated at each point. There are many established ways of doing this. The initial implementation of the gradient descent was with a static finite central differences approximation for the gradient. This was the most intuitive approach for estimating the gradient.

$$\nabla_i(\vec{x}) = \frac{f(|\vec{x}|(\hat{x} + \hat{x}_i\epsilon) - f(|\vec{x}|(\hat{x} - \hat{x}_i\epsilon))}{2|\vec{x}|\sqrt{\epsilon}} \quad (11)$$

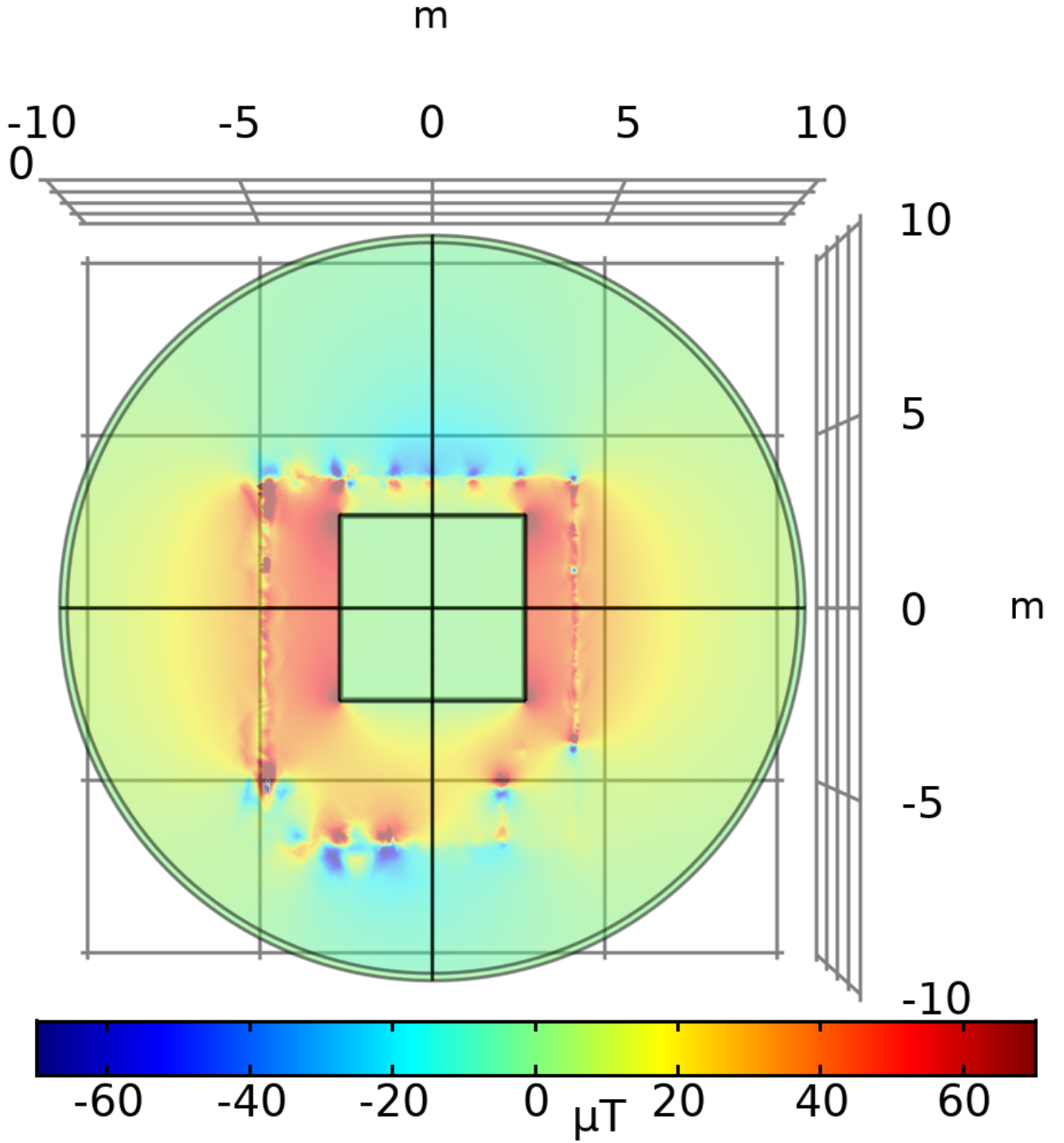


Figure 6: The x-component of the P1 Field for the n2EDM experiment with the mu-metal included. Shown is the x-y plane for an offset of $z=0$. The field inside the cube is at $1\mu\text{T}$. The field outside the cube is at $50\mu\text{T}$.

Where \hat{x}_i is the unit vector in direction i and ϵ is the machine ϵ supplied by the system. This method is more accurate when it comes to approximating the gradient than forward or backwards differences gradient approximations [8].

This method is guaranteed to converge to a local minimum when coupled with the step size given by the Barzilai Borwein method (BBM) [1]:

$$\gamma_n^{BB} = \frac{|(\vec{x}_n - \vec{x}_{n-1})^T [\nabla F(\vec{x}_n) - \nabla F(\vec{x}_{n-1})]|}{\|\nabla F(\vec{x}_n) - \nabla F(\vec{x}_{n-1})\|^2} \quad (12)$$

This quasi Newton method for calculating the step size offers guaranteed convergence without the

	Gradient SPSA	Gradient BB	Step Size SPSA	Step size BB	SPSA Steps	BB Steps
Min Time	976.301 μ s	21.302 ms	140.349 ns	46.077 ms	1.451 ms	43.740 ms
Median Time	1.297 ms	26.591 ms	148.694 ns	49.123 ms	1.770 ms	50.147 ms
Mean Time	1.623 ms	31.850 ms	171.692 ns	55.272 ms	2.529 ms	57.252 ms
Max Time	63.520 ms	133.480 ms	2.318 μ s	108.894 ms	63.135 ms	131.147 ms
Memory Alloc.	76.81 KiB	1.77 MiB	48 B	3.54 MiB	131.17 KiB	3.54 MiB

Table 1: It should be known that this benchmarking is tied to computational performance and only the relative times are relevant. This data suggests that the SPSA descent is of the order 20 times faster than the BB method.

increased computational requirement for small constant step sizes.

However, this approach has two major problems. Firstly, the algorithm is computationally expensive as it requires $2 \times d$ ($d = 24 = n_{sens} \times 3$) function evaluations to approximate the gradient, where d is the dimension of the problem. The second problem is the fact that, whilst this algorithm is guaranteed to converge, convergence is only guaranteed to a local minimum. It is impossible for the algorithm to escape local minima.

3.4.2 Simultaneous Perturbation Stochastic Approximation (SPSA)

A more computationally efficient method of approximate gradient descent is the simultaneous perturbation stochastic approximation (SPSA) algorithm[2].

In high dimensional optimisation problems the simultaneous perturbation stochastic approximation (SPSA) method can lead to large decreases in the computational cost of the descent. This is because, instead of calculating the gradient in each direction individually as in Eq. 11 the gradient is calculated in a random direction:

$$\nabla_{k,i}(\vec{x}_k) = \frac{f(\vec{x} + c_k \vec{\Delta}_k) - f(\vec{x} - c_k \vec{\Delta}_k)}{2c_k \Delta_{k,i}} \quad (13)$$

Where Δ_k is drawn from $(Bernoulli(\frac{1}{2}))^{3 \times num_{sens}}$. This leads to 2 instead of $2 \times d$ function calls for the gradient calculation. In our case this results in a decrease in the computation time of factor ≈ 20 . This was measured by benchmarking with the Julia package BenchmarkTools [12].

The reason for the significant reduction in calculation cost for the step size is that for the SPSA algorithm convergence is achieved reliably with a decreasing step size given by:

$$a_k = \frac{a}{(k + A)^\alpha} \quad (14)$$

SPSA has an additional benefit over the BBM. It is able to leave local minima with the correct choice of the perturbation coefficient c_k . Due to the randomised nature of the direction and the step length being uncoupled to the gradient the SPSA algorithm is able to leave local minima.

Another problem with the optimisation is the question of what to do with runaway descents. This is where the fluxgate positions leave the measurement zone, the final configuration can be a minimum if all fields couple equally weakly. In this case the result may minimise the condition number whilst having no actual information about the fields inside the measurement zone. To fix this problem the search domain was constrained to a region within a fixed distance from the mu-metal shield. Since the field near the mu-metal shield is closely related to the background field inside the measurement zone without the shield and the fluxgates have to be attached to the mu-metal shield this is a good choice. The constraint was applied by setting wandering coordinates to the edge of this domain.

3.5 Random Search

As mentioned in section 3.4.2 this method guarantees convergence to a local minimum. Unfortunately, from the results it becomes clear that the local minima are not guaranteed to come close to the global minimum. To achieve a good approximation of the global minimum, an adapted method of the ab initio

random structure search (AIRSS)[6], used to find the minimum energy solution of solid state systems, is used. The optimisation algorithm is randomly initialised and the descent is performed separately for each optimisation. The lowest result is taken as an approximation of the global minimum.

4 Results and Discussion

The plots are titled to describe the setup used. The list in the titles can be understood as describing the setup, the method by which the fields are created, the starting configuration and the fields supplied. For the field scaling the first number describes the goal field for the homogeneous fields and the second number the goal for the linear fields. As discussed in 4.6 the resulting coupling matrix has to be scaled accordingly in order to compensate. The methods by which the fields are generated are a direct field calculation in Julia or the addition of a COMSOL step to include the mu-metal shield.

4.1 Convergence of the Algorithms

Due to computational limitations the number of iterations was limited to 1000 for the Julia (Generated) fields. As shown in Fig. 7 the prototype algorithm converges to a minimum within the algorithm's runtime.

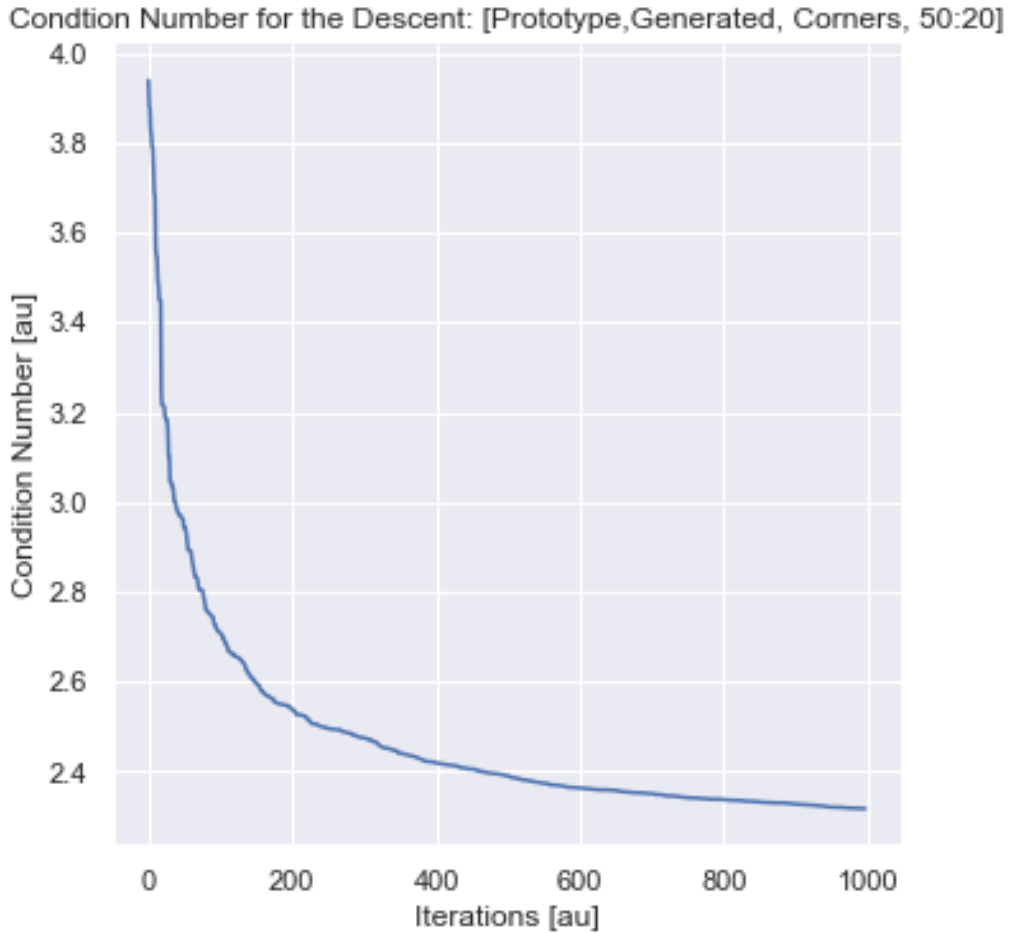


Figure 7: The condition number (arbitrary units) as a function of the iterations (arbitrary units) for the corner starting positions in the prototype configuration. The condition number reached levels off at around 300 iterations at 2.5 but continues improving to 2.31 at the end of the optimisation. The fields used were generated directly in Julia.

As shown in Fig. 8 convergence is slower for the n2EDM configuration. It does not occur within the runtime. Instead the calculations have to be extended to around 10000 iterations.

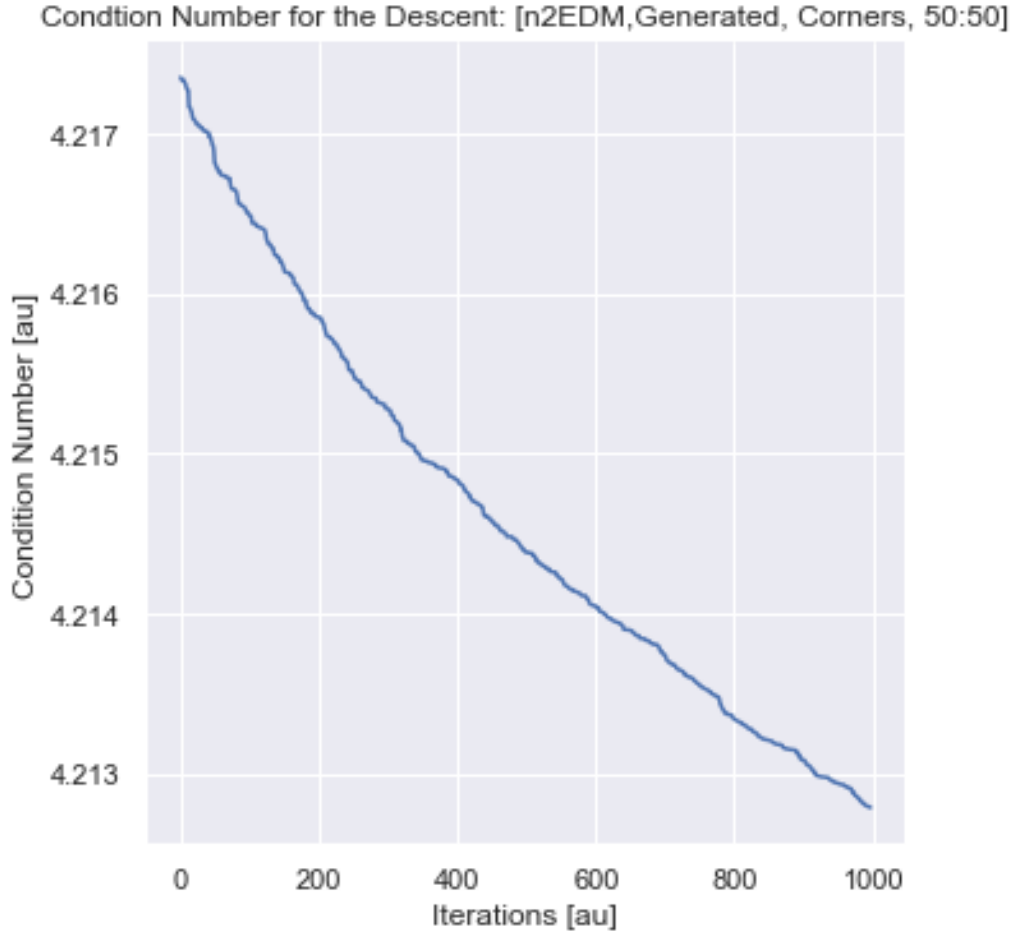


Figure 8: The condition number(arbitrary units) as a function of the iterations(arbitrary units) for the corner starting positions in the n2EDM configuration. At 1000 iterations convergence is not yet reached.

4.2 Number of Sensors

In his work, another semester student determined the ideal number of sensors for the optimisation to be 8 [11]. This is verified by the optimisation run on the prototype simulation. As shown in Tab. 2 the lowest condition numbers were achieved for 8 sensors in both the purely Julia simulated setup and that generated in COMSOL. These values were obtained by running the SPSA descent over 1000 iterations with 10 random initialisations and choosing the lowest result achieved.

prototype	Lowest Condition	Achieved (10 Runs)
Number of Sensors	Julia	COMSOL
4	2.206	2.564
5	2.543	2.125
6	2.116	2.15
7	2.038	2.183
8	1.821	2.117
9	1.966	2.185
10	2.258	2.419
11	2.148	2.593
12	1.944	2.871
13	1.85	2.186
14	1.966	2.697
15	2.022	2.765
16	1.962	3.116
17	1.982	3.562
18	1.927	3.266
19	1.908	3.51
20	1.943	3.443

Table 2: The lowest condition numbers achieved over 10 runs of the SPSA algorithm for varying numbers of fluxgates in the prototype configuration. In both the Julia and the COMSOL variant the lowest condition number was achieved for 8 fluxgates. In both configurations the goals for the linear and homogeneous fields were set to $50 \mu\text{T}$.

4.3 Refinements on Previously Calculated Positions

A descent was initialised using the positions calculated by Tim Roethlisberger [11] to be the optimal configuration. With the fields used here these resulted in a condition number of 2.65 for the pure Julia setup and 2.74 for the COMSOL variant. When optimised by the algorithm these were reduced to 2.51 for the pure Julia setup and 2.54 for the COMSOL variant after 10000 iterations. This is a mild improvement on the condition number achieved by the initial points. As shown in Fig. 9 the algorithm converges for both variants but in both cases the improvement over the initial condition number is marginal. Further, it is worse than the descent initialised with Corners as the starting points, this is shown in Section. 4.4.

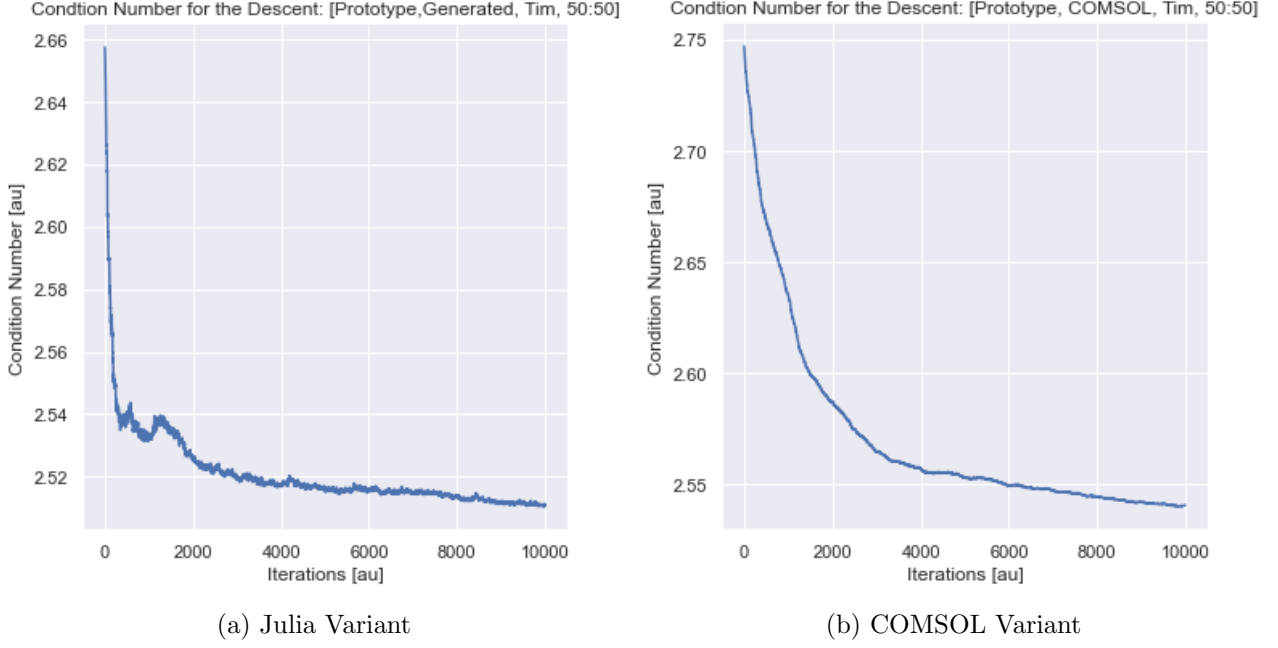


Figure 9: The condition number as a function of the SPSA iterations for the previously calculated starting positions in the prototype configuration with $m = 50, m_g = 50\mu T$. The lowest condition number reached is 2.51 for the Julia variant and 2.54 with the COMSOL fields.

4.4 Refinements on Corner Starting Positions

Another option for initialisation is to start with points located at the corners of the search domain. This should represent the symmetries inherent in the setup. With this initialisation the condition number achieved is 1.71 for the prototype descent with $m = 50\mu T$ and $m_g = 50\mu T$. For $m = 50, m_g = 5\mu T$ the results achieved were significantly worse at 2.01. Shown in Fig. 10 is the convergence for the two corner configurations after 10000 steps.

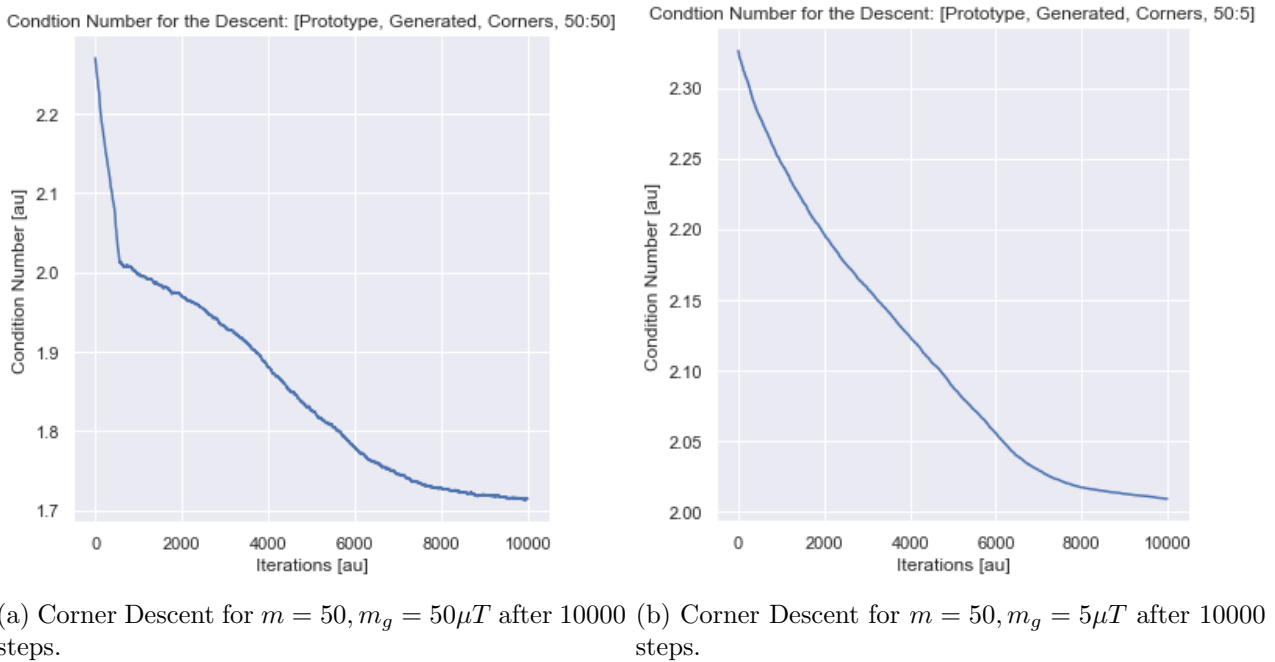


Figure 10: The condition number as a function of the SPSA iterations for corner starting positions in the prototype configuration. The lowest condition number reached is 1.71 for $m = 50, m_g = 50\mu T$ and 2.01 for the $m = 50, m_g = 5\mu T$.

4.5 Random Structure Search

Theoretically the lowest results for the condition number should be obtained using the random search. Shown in Fig. 11 is the convergence to the minimum for the results of the random structure search followed by a fine tuning for the lowest condition number found. This 'fine tuning' was done to reduce the computational cost of the optimisation.

Condition Number for the Descent: [Prototype, Generated, Fine Tuned, 50:50]

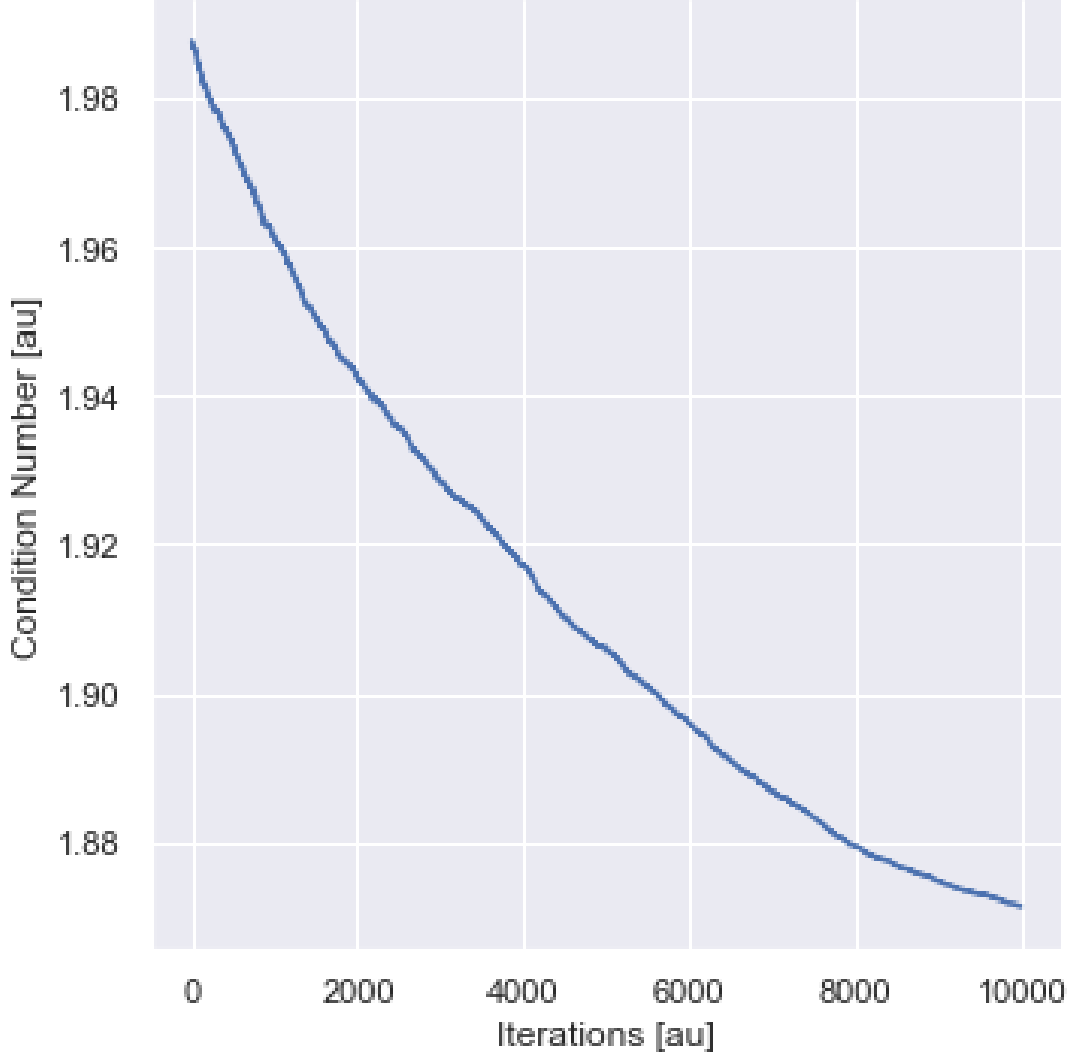


Figure 11: The condition number as a function of the iterations for the fine tuned random structure search. The condition number reached is 1.89. Initial seeding for the descent was done by initialising 600 random descents to a depth of 200 iterations and choosing the lowest result as the starting point.

This method led to a condition number of 1.89, comparable to the result obtained with the corner descent. This is likely due the limited number of initialisations and also the completely random manner of the initialisation.

4.6 Scaling

One problem with the method arises when different goal field coupling constants are used for the homogeneous and gradient fields respectively. This unequal coupling results in large condition numbers, of around 20, as the gradient fields couple far weaker (a factor 10) than the homogeneous fields. In order to combat this, an attempt was made at scaling the relevant matrix components. Here the attempted fix was to scale the gradient components up by a factor of $\frac{m}{m_g}$. The results for the generated fields for

Condition Numbers:	Scaling: (m_g)	5:	20:	50:
Configuration:	Field Type:			
Prototype:	Julia:	2.54	2.32	2.50
	COMSOL:	13.72		2.04
PSI:	Julia:	5.39	1.68	4.21
	COMSOL:	16.53	2.21	

Table 3: The condition numbers achieved by the corner descent for various configurations of the SPSA algorithm. The results for the Julia descent are reached after 1000 iterations. COMSOL results are achieved after 10000 iterations.

the prototype were good, as shown in Tab. 3. The achieved values compared to those achieved with the 50:50 fields. This changed when applied to the COMSOL fields, likely due to the influence of the mu-metal shield being non-linear in components not included in the goal fields 9. The results for the COMSOL fields were only in the expected region for the fields without additional rescaling.

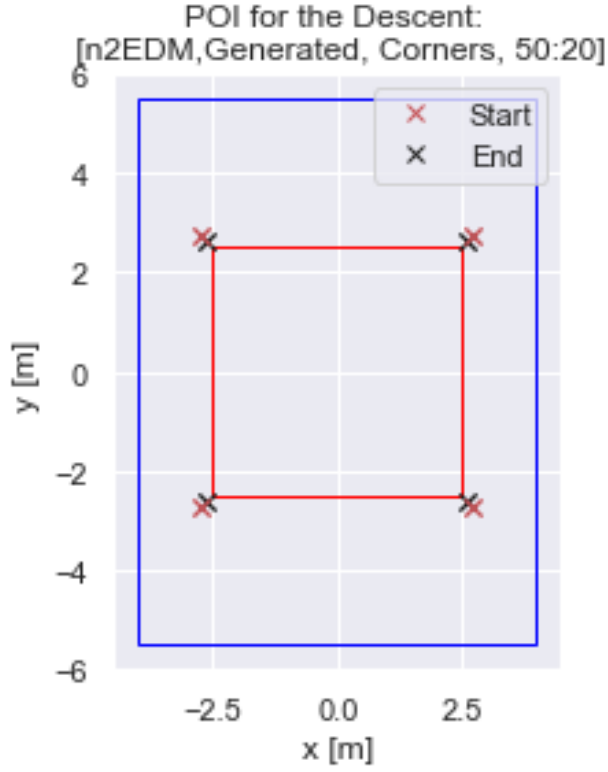
The results for the n2EDM configuration are poor for both the COMSOL and the generated fields. Strangely the results for $m_g = 20\mu T$ are very low despite this not being the actual strength used in the generation of the fields. It is unclear as to why the scaling does not work for the generated fields as it does for the prototype. It is likely due to the lack of symmetry in the system. This merits further investigation. Possible avenues for an improvement of the scaling are an examination of the effect the scaling has on the end positions or the use of less simplistic scaling of the components. Whilst scaling only the relevant components in the matrix yielded poorer results than the brute scaling of all the gradient components for the generated fields in the prototype configuration it could be better at capturing non-linearities due to the mu-metal shield and also the asymmetrical setup for the n2EDM measurement.

For the n2EDM setup it should be noted that the scaling was simply conducted afterwards on the components of the correlation matrix. Unlike the prototype the fields were only calculated with goal fields at $m = 50\mu T, m_g = 5\mu T$. This means that the underlying fields were the same in all cases and then scaled up to different levels afterwards. This differs from the prototype approach where the fields were first calculated for different values of m_g and then scaled to the same level.

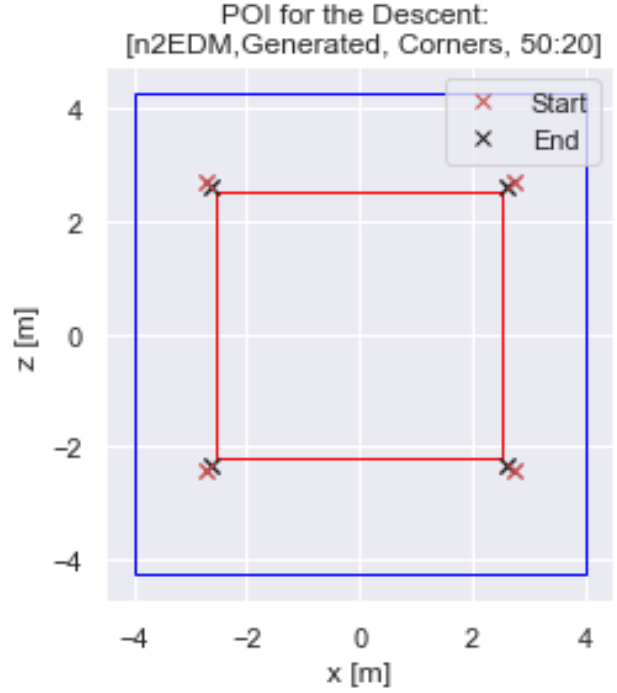
4.7 n2EDM Measurement

The numbers resulting from the optimisation on the n2EDM setup are around 5 for the fields generated directly in Julia and 15 for the COMSOL fields. This increase over what is expected is likely due to the fact that the currents supplied to the optimisation are for a system with $m = 50\mu T$ and $m_g = 5\mu T$. Since the scaling attempt detailed in section 4.6 did not solve the problem, other approaches are required. One possibility is to rerun the calculation of the currents for the n2EDM setup with the equal values for m and m_g and apply the standard descent. Another possibility is to find a method of scaling the components that results in the proper field values. The best condition numbers were achieved when the scaling was set for a field configuration with $m_g = 20\mu T$.

Depicted in Fig. 12 are the final positions for the corner descent in various configurations. The coordinates and condition number for this configuration are given in B. From the figure we can see that the optimisation does not change the positions dramatically and the changes are symmetric around the mu-metal shield.



(a) Corner Positions x-y view



(b) Corner Positions x-z view.

Figure 12: Finalised positions of the SPSA descent after 10000 iterations for the corner configuration of the n2EDM. Depicted in red are the starting positions. Black are the final positions determined by the algorithm.

4.8 Validity of the Final Positions

Whether the final positions hold up in the real world must be verified experimentally. As can be seen in Fig. 13 the final positions for the prototype descent with corner starting positions do not change significantly from their initial positions. Despite this fact, an improvement in the condition number of has been achieved. The coordinates of the points are given in Tab. 4.

Fluxgate	x /m	y /m	z /m
1	-0.544	-0.184	-0.472
2	0.547	-0.183	-0.474
3	-0.585	0.845	-0.478
4	0.585	0.845	-0.482
5	-0.548	-0.181	0.476
6	0.541	-0.184	0.472
7	-0.585	0.841	0.475
8	0.585	0.844	0.479

Table 4: Final fluxgate positions for the corner descent in the prototype 50:50 scaling configuration.

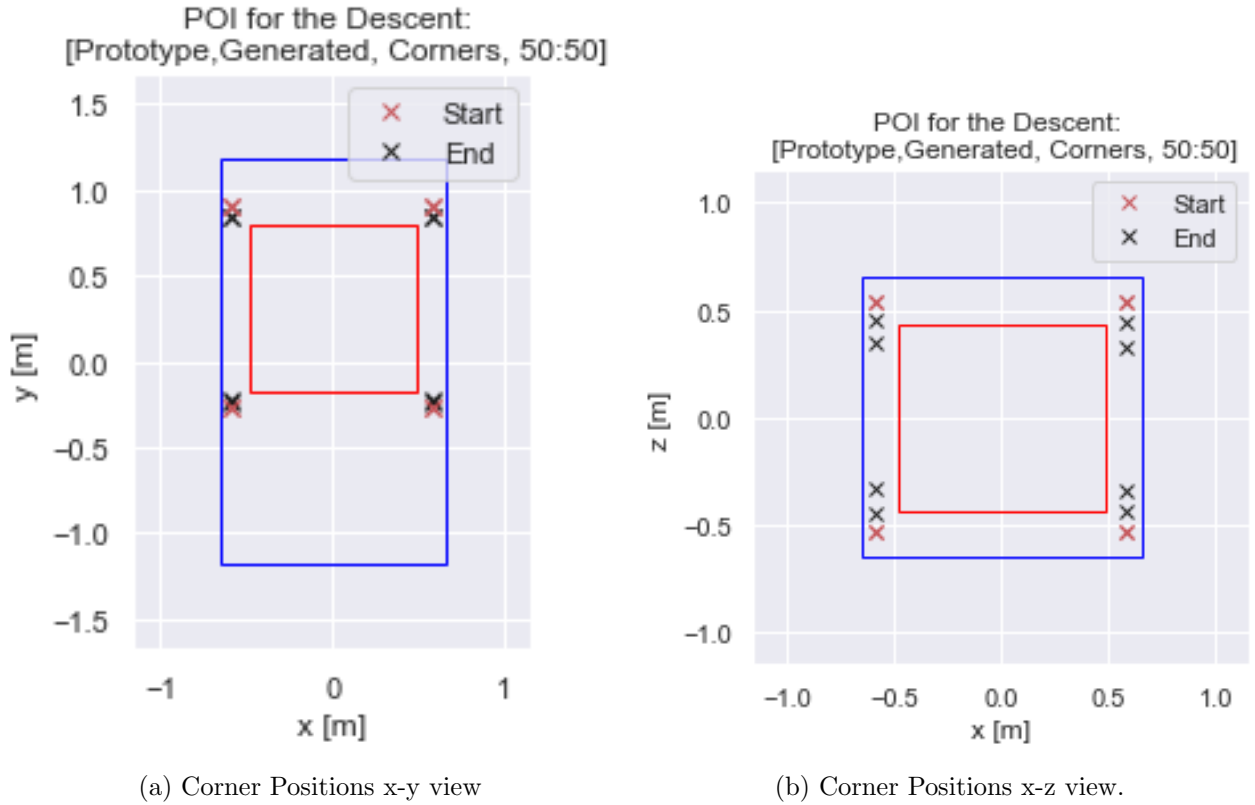


Figure 13: Finalised positions of the SPSA descent after 10000 iterations for the corner configuration. Depicted in red are the starting positions. Black are the final positions determined by the algorithm.

This calls into question the stability of the configuration. As a result the change in the condition number when the fluxgates are perturbed should be examined.

4.9 Stability of the Final Positions

For the results to be of use in the nEDM experiment the final positions for the fluxgates have to be stable under positional shifts due to human error. Further, this investigation allows us to visualise whether or not we are actually in a minimum. As can be seen in Fig. 14 the change in the condition number due to the shifting of a single fluxgate is of the order 10^{-3} for a shift of $\pm 10\text{cm}$. This suggests that the position of a single fluxgate is fairly stable under spatial perturbation.

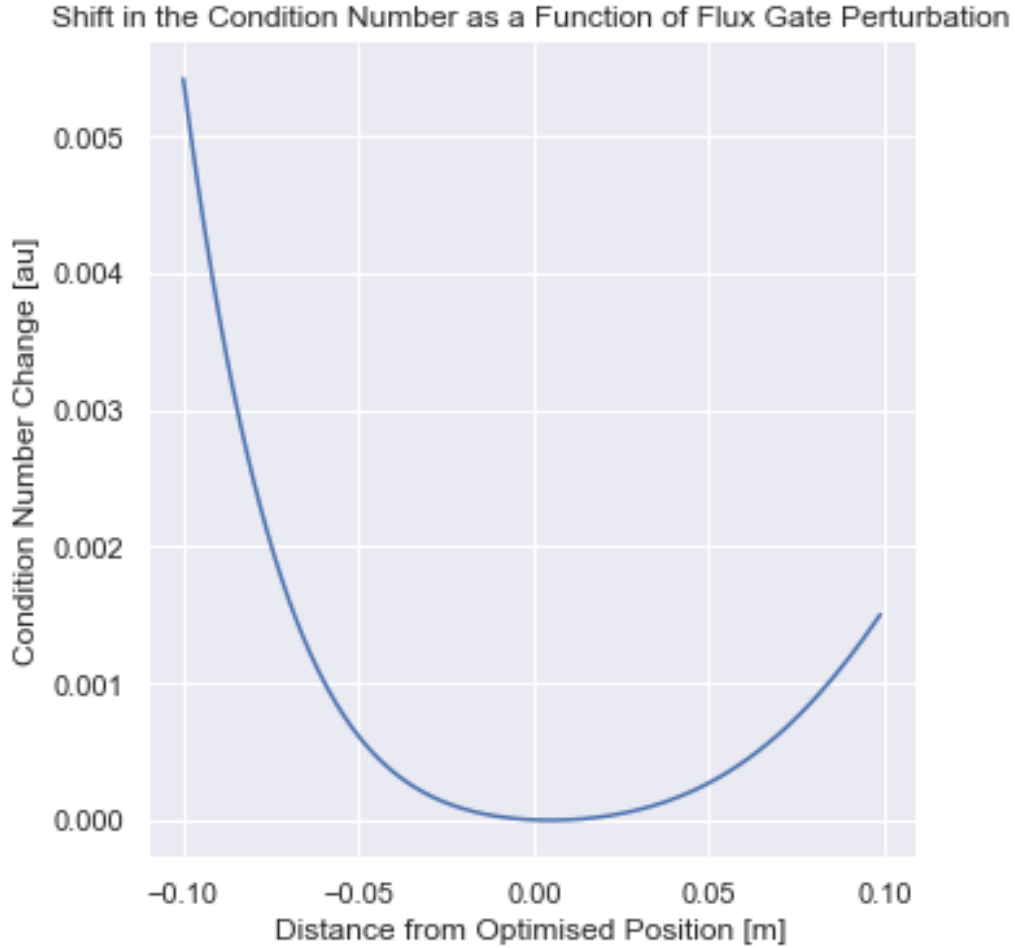


Figure 14: The shift in the condition number of the correlation matrix for a shift in the z coordinate for the first fluxgate of the corner descent. Depicted are shifts of up to 10cm in either direction. The condition number shift is of the order 10^{-3} . The calculation is performed on the fine tuned descent after 10000 iterations.

The fact that we are in a minimum also suggests that the condition number has been minimised in this instance.

4.10 Improvements

The values for the n2EDM experiment suggest that the coupling here is not ideal. This is most likely due to the asymmetry in the coil system. Whether this requires amending will be determined when the system is tested. The algorithm itself has a few avenues for improvement. Currently it relies heavily on vectorisation and initialises unnecessary arrays. The algorithm could be optimised by reducing the number of array assignments. Further, as Julia is a compiled language, iteration is more expensive than vectorisation. Taking this into account can reduce the computational costs of the correlation matrix calculation. This could reduce computation time for the optimisation significantly.

5 Conclusion

The optimisation shows promise, with the results for the prototype consistently being between 1.7-2 for the fine tuned descent. This is a proof of concept. The good performance of the corner descent also suggests that we can make use of the inherent symmetries of the system as was to be expected. Unfortunately, the results for the n2EDM experiment are not in the range of values that were aspired to. With values falling around 5 it is unclear whether this is a problem for the usefulness of the final results. This has to be verified experimentally. With greater computational bandwidth a more

comprehensive random search may yield better results than those achieved here but the problems with the n2EDM AMS's values seem to be more fundamental. The final positions of the descent are stable under perturbation which suggests they may be of use when implemented in the real world.

References

- ¹J. B. Jonathan Barzilai, “Two-point step size gradient methods”, *IMA Journal of Numerical Analysis* **8**, 141–148 (1988).
- ²J. C. Spall, “Implementation of the simultaneous perturbation algorithm for stochastic optimization”, *IEEE Transactions on Aerospace and Electronic Systems* **34**, 817–823 (1988).
- ³M. T. Antonio Riotto, “Recent progress in baryogenesis”, *Annual Review of Nuclear and Particle Science* **49:35-75** (1999).
- ⁴L. N. Trefethen and D. Bau, *Numerical Linear Algebra* (SIAM, 2000).
- ⁵L. HAN*† and M. NEUMANN‡, “Effect of dimensionality on the nelder–mead simplex method”, *Optimization Methods and Software* (2004).
- ⁶C. J. Pickard and R. J. Needs, “Ab initio random structure searching”, *Journal of Physics: Condensed Matter* **23** (2011).
- ⁷B. Franke, “Investigations of the internal and external magnetic fields of the neutron electric dipole moment experiment at the paul scherrer institute”, PhD thesis (ETH Zurich, 2013).
- ⁸J. Blazek, *Computational fluid dynamics: principles and applications* (Elsevier, 2015).
- ⁹M. Rawlik, “Active magnetic shielding and axion dark matter search”, PhD thesis (ETH Zurich, 2018).
- ¹⁰M. Rawlik, *Coils Package for Julia*, <https://github.com/rawlik/Coils.jl>, 2018.
- ¹¹T. Röthlisberger, *Optimization of the fluxgate positioning for the n2EDM Surrounding Field Compensation prototype*, June 2019.
- ¹²*Julia Package BenchmarkTools*, <https://github.com/JuliaCI/BenchmarkTools.jl>.

A Prototype Results

Corners/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	-0.534	-0.189	-0.483	2.040
	2	0.556	-0.192	-0.474	
	3	-0.577	0.843	-0.474	
	4	0.574	0.840	-0.473	
	5	-0.549	-0.182	0.478	
	6	0.538	-0.190	0.476	
	7	-0.571	0.844	0.481	
	8	0.585	0.846	0.478	
5 [μT]	1	-0.585	0.419	-0.535	13.720
	2	0.435	0.523	-0.535	
	3	-0.583	0.847	-0.508	
	4	0.585	0.880	-0.533	
	5	-0.585	0.450	0.535	
	6	0.204	-0.030	0.490	
	7	-0.585	0.869	0.534	
	8	0.585	0.875	0.530	

Table 5: Results for the prototype COMSOL Corner Optimisation

Fine Tuned/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	-0.506	-0.178	-0.535	1.963
	2	0.585	0.900	-0.476	
	3	-0.582	-0.190	0.505	
	4	0.585	-0.210	0.495	
	5	-0.531	-0.169	-0.535	
	6	0.532	-0.177	-0.527	
	7	-0.555	0.889	-0.535	
	8	0.497	-0.188	-0.534	
5 [μT]	1	0.571	0.061	-0.535	44.101
	2	0.580	0.689	-0.534	
	3	-0.585	0.104	0.534	
	4	-0.579	-0.127	0.535	
	5	0.583	-0.010	-0.534	
	6	-0.138	-0.248	-0.435	
	7	-0.585	0.894	-0.441	
	8	0.584	0.891	-0.531	

Table 6: Results for the prototype COMSOL Fine Tuned Optimisation

Previous/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	0.461	-0.270	0.502	2.541
	2	0.470	-0.270	-0.490	
	3	-0.452	-0.270	0.503	
	4	-0.470	-0.270	-0.482	
	5	-0.002	0.900	0.489	
	6	0.005	0.900	-0.493	
	7	0.001	0.316	0.480	
	8	-0.003	0.314	-0.479	
5 [μT]	1	-0.585	0.434	-0.535	11.765
	2	-0.564	0.866	-0.530	
	3	-0.564	0.861	-0.526	
	4	0.585	0.848	-0.503	
	5	-0.574	0.900	0.535	
	6	-0.584	0.876	0.535	
	7	-0.073	-0.130	0.535	
	8	0.585	0.844	0.489	

Table 7: Results for the prototype COMSOL Optimisation of Previous POI

Corners/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	-0.585	-0.231	-0.338	1.715
	2	0.585	-0.230	-0.347	
	3	-0.585	0.839	-0.447	
	4	0.584	0.838	-0.444	
	5	-0.585	-0.230	0.344	
	6	0.585	-0.232	0.329	
	7	-0.585	0.840	0.448	
	8	0.584	0.837	0.447	
5 [μT]	1	-0.544	-0.184	-0.472	2.009
	2	0.547	-0.183	-0.474	
	3	-0.585	0.845	-0.478	
	4	0.585	0.845	-0.482	
	5	-0.548	-0.181	0.476	
	6	0.541	-0.184	0.472	
	7	-0.585	0.843	0.475	
	8	0.585	0.844	0.479	

Table 8: Results for the prototype Julia Corner Optimisation

Fine Tuned/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	0.483	-0.178	-0.434	1.871
	2	-0.585	0.855	-0.445	
	3	0.585	-0.206	-0.054	
	4	0.585	0.894	0.151	
	5	-0.523	-0.178	0.429	
	6	0.585	-0.227	0.249	
	7	0.480	-0.183	-0.443	
	8	-0.518	-0.177	0.420	

Table 9: Results for the prototype Julia Fine Tuned Optimisation

Previous/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	0.458	-0.270	0.478	2.511
	2	0.442	-0.270	-0.459	
	3	-0.451	-0.270	0.471	
	4	-0.438	-0.270	-0.460	
	5	0.010	0.900	0.457	
	6	0.017	0.900	-0.464	
	7	-0.027	0.253	0.464	
	8	-0.053	0.288	-0.465	
5 [μT]	1	0.466	-0.270	0.493	2.763
	2	0.469	-0.270	-0.497	
	3	-0.468	-0.270	0.494	
	4	-0.469	-0.270	-0.498	
	5	-0.003	0.900	0.485	
	6	0.001	0.900	-0.485	
	7	0.002	0.312	0.478	
	8	0.005	0.317	-0.477	

Table 10: Results for the prototype Julia Optimisation of Previous Poi

B n2EDM Results

Corners/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
5 [μT]	1	-0.544	-0.184	-0.472	16.429
	2	0.547	-0.183	-0.474	
	3	-0.585	0.845	-0.478	
	4	0.585	0.845	-0.482	
	5	-0.548	-0.181	0.476	
	6	0.541	-0.184	0.472	
	7	-0.585	0.843	0.475	
	8	0.585	0.844	0.479	
Fine Tuned/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	-2.720	-2.065	-2.287	12.358
	2	2.688	2.721	2.429	
	3	2.707	2.722	2.454	
	4	-2.247	2.606	2.650	
	5	2.650	-2.677	-2.230	
	6	-2.722	2.722	-2.414	
	7	-1.996	-2.721	-2.389	
	8	2.722	-2.722	2.702	

Table 11: Results for the n2EDM COMSOL Optimisation

Corners/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	-2.628	-2.628	-2.313	3.884
	2	2.628	-2.626	-2.315	
	3	-2.627	2.625	-2.314	
	4	2.624	2.621	-2.312	
	5	-2.628	-2.629	2.601	
	6	2.628	-2.628	2.599	
	7	-2.627	2.628	2.602	
	8	2.625	2.625	2.601	
20 [μT]	1	-2.624	-2.624	-2.312	1.679
	2	2.624	-2.624	-2.312	
	3	-2.624	2.623	-2.313	
	4	2.623	2.623	-2.312	
	5	-2.624	-2.624	2.597	
	6	2.624	-2.624	2.597	
	7	-2.623	2.624	2.597	
	8	2.623	2.622	2.597	
5 [μT]	1	-2.619	-2.620	-2.307	4.396
	2	2.619	-2.619	-2.307	
	3	-2.620	2.619	-2.309	
	4	2.614	2.615	-2.309	
	5	-2.619	-2.620	2.591	
	6	2.620	-2.620	2.592	
	7	-2.618	2.622	2.592	
	8	2.617	2.618	2.591	

Table 12: Results for the n2EDM Julia Corner Optimisation

Fine Tuned/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	2.526	-2.525	-2.214	4.351
	2	-2.525	2.526	-2.215	
	3	2.524	2.525	2.501	
	4	-2.527	-2.526	2.501	
	5	2.526	-2.526	-2.215	
	6	-2.526	2.526	2.501	
	7	-2.525	2.526	-2.215	
	8	2.525	-2.526	2.500	
20 [μT]	1	2.522	2.522	2.502	1.800
	2	1.833	-2.722	1.055	
	3	-2.525	-2.524	-2.213	
	4	2.522	-2.525	2.501	
	5	-1.614	-2.613	-1.055	
	6	2.523	2.522	-2.214	
	7	2.523	2.522	-2.215	
	8	-2.526	2.524	2.501	
5 [μT]	1	0.387	-2.719	0.278	3.907
	2	-2.522	-2.521	-2.211	
	3	2.522	-2.522	2.502	
	4	2.522	-2.522	-2.214	
	5	2.522	2.522	-2.214	
	6	-2.610	-1.054	0.509	
	7	2.522	2.522	-2.214	
	8	-2.520	2.524	2.497	

Table 13: Results for the n2EDM Julia Fine Tuned Optimisation

Previous/ Scaling	Fluxgate	x [m]	y [m]	z [m]	Condition Number
50 [μT]	1	2.528	-2.524	2.501	4.351
	2	2.529	-2.525	-2.214	
	3	-2.524	-2.525	2.502	
	4	-2.524	-2.525	-2.214	
	5	-2.524	2.525	2.502	
	6	-2.524	2.525	-2.215	
	7	-2.524	2.525	2.501	
	8	-2.524	2.524	-2.215	
20 [μT]	1	2.525	2.430	2.522	2.122
	2	-2.523	-2.522	-2.286	
	3	2.430	-2.524	-2.522	
	4	2.525	-2.286	2.522	
	5	-2.523	-2.522	2.430	
	6	-2.286	2.522	-2.522	
	7	-2.522	2.430	2.522	
	8	-2.524	-2.522	-2.286	
5 [μT]	1	2.527	-2.514	2.493	5.045
	2	2.527	-2.513	-2.210	
	3	-2.518	-2.523	2.495	
	4	-2.522	-2.522	-2.213	
	5	-2.522	2.522	2.502	
	6	-2.522	2.522	-2.214	
	7	-2.522	2.522	2.502	
	8	-2.522	2.522	-2.214	

Table 14: Results for the n2EDM Julia Optimisation of Previous POI