University of Kentucky

UKnowledge

Theses and Dissertations--Physics and Astronomy

Physics and Astronomy

2024

Magnetic Spin Transport Coils for the n2EDM Experiment

David C. Bowles

University of Kentucky, dbowles333@gmail.com

Author ORCID Identifier:

https://orcid.org/0009-0006-9552-8482

Digital Object Identifier: https://doi.org/10.13023/etd.2024.55

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Bowles, David C., "Magnetic Spin Transport Coils for the n2EDM Experiment" (2024). *Theses and Dissertations--Physics and Astronomy*. 120.
https://uknowledge.uky.edu/physastron_etds/120

This Doctoral Dissertation is brought to you for free and open access by the Physics and Astronomy at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Physics and Astronomy by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student’s advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student’s thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

David C. Bowles, Student

Dr. Chris Crawford, Major Professor

Dr. Chris Crawford, Director of Graduate Studies

Magnetic Spin Transport Coils for the n2EDM Experiment

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Arts and Sciences
at the University of Kentucky
By
David C. Bowles
Lexington, Kentucky

Director: Christopher B. Crawford, Professor of Physics
Lexington, Kentucky
2024

Copyright© David C. Bowles 2024

ABSTRACT OF DISSERTATION

Magnetic Spin Transport Coils for the n2EDM Experiment

Magnetic Spin Transport Coils for the n2EDM Experiment

The n2EDM experiment is being construct at Paul Scherrer Institute to measure the electric dipole moment of the neutron (nEDM), in a search for new physics which could help explain the baryon asymmetry of the universe. To reach the experimental sensitivity goal of $d_n \sim 10^{-27}$, over an order of magnitude improvement from the current world limit, a spin polarization transfer efficiency of 99% for ultracold neutrons entering or exiting the precession chamber is required, placing a stringent requirement on the adiabaticity of the magnetic field taper in the neutron guide system. The spin transport fields transition from 5 T (longitudinal) in the superconducting polarizer to 30 μT (vertical) at the entrance of the magnetic shield room, further tapering to 1 μT in the precession chambers. Utilizing the magnetic scalar potential, the design process was separated into two phases: determination of the ideal magnetic field profile, and calculation of surface current coils within the geometrical constraints of the apparatus to realize this field. New techniques were developed to model the magnetic field and optimize adiabaticity of the spin transport, and to design a coil to compensate for magnetic fringes on the inner wall of the shield. The resulting hand-wound spin transport coils were installed and tested at PSI, and exceeded the spin transport requirements.

KEYWORDS: neutrons, magnetic fields, magnetic scalar potential, guide coils, dipoles, CP-violation, nEDM, COMSOL, 3D printing

David C. Bowles

April 24, 2024

Magnetic Spin Transport Coils for the n2EDM Experiment

By

David C. Bowles

Christopher B. Crawford

Director of Dissertation

Christopher B. Crawford

Director of Graduate Studies

April 24, 2024

Date

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Chris Crawford, for his support, mentorship, and guidance throughout my graduate and undergraduate career. Dr. Crawford has had a hand in most of my physics education, without his countless hours of help I would not have made it as far as I have.

Thank you to my committee members for there guidance in research and challenging me to strive for the best: Dr. Brad Plaster, Dr. Renee Fatem, and Dr. Peter Hislop.

A special thanks to all my collaborators for PSI-n2nEDM experiment, especially Dr. Geza Zsigmond and Dr. Efrain Patrick. Both have been instrumental in the development of the guide coils.

Thank you to my colleagues and research group members; their experience, insights, and support have been invaluable. Thanks to Joseph Bates, David Matthews, Stephanie Thomas, Austin Nelson, Jacob Hemple, Amber McKey, Hannah Smith, and Michelle Gervais. They have been my closest friends and support network, without whom I would not have finished.

Finally, thank you to my family for their endless support and encouragement. Without them, I would not have been able to attempt this challenge. A special thanks to Gerry Adair for his help editing my dissertation and giving me the confidence to finish. A very special thanks to my fiancée Colleen, your patients, encouragement, and support has been a rock for me. I am so fortunate to have such an amazing partner.

TABLE OF CONTENTS

Acknowledgments . . . iii

Table of Contents . . . iv

List of Figures . . . vii

List of Tables . . . x

Chapter 1 Introduction . . . 1

- 1.1 The Neutron's EDM . . . 1
  - 1.1.1 Motivation . . . 1
  - 1.1.2 Symmetries . . . 3
  - 1.1.3 $\mathcal{CP}$ Violation and The Sakharov Conditions . . . 6
  - 1.1.4 Measuring the nEDM . . . 7
  - 1.1.5 n2EDM Design . . . 9
- 1.2 Magnetic Field Adiabaticity . . . 11
  - 1.2.1 Field rotation . . . 11
  - 1.2.2 Larmor Precession . . . 13
- 1.3 Magnetic Scalar Potential and Coil Construction . . . 14

Chapter 2 Target Magnetic Field Design . . . 21

- 2.1 Background Magnetic Field And The MSR . . . 21
  - 2.1.1 The MSR . . . 22
  - 2.1.2 COMSOL Modeling of Background Magnetic Fields . . . 26
  - 2.1.3 Total Background Magnetic Field In The Target Region . . . 30
- 2.2 Finding the Target Field for the MSP . . . 33

- 2.2.1 Simplifying Adiabaticity for Conjugate Gradient 34
- 2.2.2 Designing Magnetic Flux Basis Functions 37
- 2.2.3 Modeling Basis Functions In COMSOL 40
- 2.2.4 Conjugate Gradient and Optimal Coefficients 42

Chapter 3 Coil Design and Modeling 48
- 3.1 Finding Isocontours 48
- 3.2 Extracting the Coil Geometries 52
- 3.3 Refining Wire Exports 56
  - 3.3.1 Sorting Contours into individual Wires 57
  - 3.3.2 Sorting Wires into Paths 57
- 3.4 Modeling Coils In COMSOL from Paths 65
  - 3.4.1 Importing Wires Into COMSOL and Rerouting 66
  - 3.4.2 Meshing The Coil Paths 69
  - 3.4.3 Field Exports and Adiabaticity 69

Chapter 4 Building the Coils & Results 73
- 4.1 Images of Wire Paths For Hand Wound Coils 73
- 4.2 3D Printing Coil Shells 76
  - 4.2.1 Trenching Wires On Surface 77
  - 4.2.2 Clip Generation and Placement 79
- 4.3 PCB Coils 81
- 4.4 Results 82

Appendices : 86

Chapter A Technical Specifications of COMSOL Models 87
- A.1 The MSR 87
- A.2 The SCM 87

A.3 The $\mathbf{B}_0$ Coil 89
A.4 The STC 90

Chapter B Components of a COMSOL Model 92

Chapter C OBJ File Structure 94

Chapter D Additional Code 97
D.1 Input and Output Functions 97
D.2 OBJ Manipulation Functions 97

Glossary 104

Bibliography 105

Vita 109

LIST OF FIGURES

1.1 Discrete Vs. Continuous Symmetry 3

1.2 Spin Precession Chambers 9

1.3 Changing Perpendicular Field 12

1.4 Regions for MSP Coil Construction 16

1.5 Amperian Loop For Scalar Field 18

2.1 MSR Modeled in COMSOL 22

2.2 MSR Edge Meshing 23

2.3 MSR STC Center Meshing Line 23

2.4 MSR Meshing Distributions Around STC Circular Edges 25

2.5 SCM Naming Scheme 26

2.6 SCM Cylinder Face Meshing 27

2.7 SCM Sweep Meshing 28

2.8 SCM Mesh Element Conversion 28

2.9 Plot of SCM Magnetic Field Components Along The Center Line 29

2.10 $\boldsymbol{B}_0$ Coil Geometry 30

2.11 $\boldsymbol{B}_0$ Coil's Magnetic Field Components Along STC Center Lines 31

2.12 SCM Magnetic Field Components Along STC Center Lines 31

2.13 Total Background Magnetic Field Components Along STC Center Lines 32

2.14 Total Background Magnetic Field Adiabaticity Along STC Center Lines 32

2.15 Coil Region Names 38

2.16 Basis Function Model Geometry 41

2.17 Basis Function Model Flux Boundary Conditions 41

2.18 STC Basis Functions 43

2.19 Annulus Basis Functions 43

2.20 Total Target Magnetic Field . . . 46
2.21 Adiabaticity of Target Field . . . 47

3.1 Regions Of MSR For Scalar Potential Model . . . 49
3.2 Flux Boundary Condition Surfaces For Portion Of The STC Inside The MSR . . . 50
3.3 Guide Coil Naming Scheme . . . 53
3.4 Inner and $\boldsymbol{B}_0$ Coil's Geometries . . . 53
3.5 CB Inner Face Coil Geometry . . . 54
3.6 CB, CB Outer Face and Middle Coil's Geometries . . . 54
3.7 Outer and Annulus Coil's Geometries . . . 55
3.8 Total Wire Geometries for Guide Coil . . . 56
3.9 Distance-Based Sorting For Unequal Distant Distribution of Points . . . 59
3.10 Complications Of Generalized Wire Sorting . . . 60
3.11 Annulus Coil Sorting Criteria . . . 62
3.12 Starting Points of Guide Coils Visualization In Blender . . . 64
3.13 Total Guide Coil Visualization In Blender . . . 64
3.14 All Wires In COMSOL . . . 68
3.15 Rerouting For The Middle and Outer STC Edge . . . 69
3.16 Guide Coil Meshing Shell . . . 70
3.17 Guide Coil Wires Distribution Mesh . . . 70
3.18 Guide Coil Field on Center Lines of STC . . . 71
3.19 Guide Coil Field on Center Lines of STC With Background Field Included 71
3.20 Adiabaticity of Total Fields on Center Lines of STC . . . 72
3.21 Adiabaticity Comparison of Total Fields With Target Field . . . 72

4.1 Coil Images On Cylinder's Axis . . . 73
4.2 Coils Images In xy Plane . . . 74

4.3 Wound Coils On Plastic Mesh 75
4.4 Installed Annulus and $\boldsymbol{B}_0$ Coils at PSI 75
4.5 CB And CB Face Coil Around CB Support Bar 76
4.6 Installed Coil Measured And Interpolated 76
4.7 Rectangular Trenching On Cylindrical Surface 78
4.8 Vertex Normal Vector Ordering For 3D OBJ Wire 79
4.9 Wire 3D Object For CB Face and Resulting 3D Printable Coil Shell . . . 80
4.10 Basic Wire Clip Face Profile 80
4.11 PCB Coil Design 81
4.12 Ramsey Cycle From Efrain 82
4.13 Geza's Prototype Coil Neutron Polarization 83
4.14 Updated Coils Field in Top STC Region 83
4.15 Updated Coils Field in Bottom STC Region 84
4.16 Geza's Updated Coil Neutron Polarization 84

A.1 COMSOL MSR Geometry 88
A.2 Naming and Orientation of SCM Cylinders 88
A.3 Guide Coil Naming Scheme 91

C.1 OBJ File Structure 94
C.2 Normal Components Connection to Point Ordering In OBJ File 95

LIST OF TABLES

1.1 Physical quantities under $\mathcal{C}$, $\mathcal{P}$, and $\mathcal{T}$ transformations [14] 4

2.1 Export Volumes for All COMSOL Models 24
2.2 Minimization Parameters For Target Field 46

3.1 Equations 3.1, 3.2, 3.3, and 3.4 parameters 51

A.1 μ-metal layer dimensions: Dimensions are measured from outside edge to outside edge and, therefore, include the thickness of the μ-metal in the listed dimension. 89
A.2 μ-metal thicknesses. 89
A.3 SCM Parameters 90

Chapter 1 Introduction

1.1 The Neutron's EDM

1.1.1 Motivation

Electric dipole moments are sensitive probes of time-reversal $\mathcal{T}$ invariance violation and, therefore, $\mathcal{CP}$ violation. According to the Sakharov conditions [25], $\mathcal{CP}$-violation is necessary to explain the magnitude of the Baryon Asymmetry of the observable Universe. The neutron Electric Dipole Moment (EDM) is ideal for a search for $\mathcal{CP}$-violation because it is an elementary particle with zero charge, has a simple spin structure with a nonzero magnetic moment, and has a relatively long lifetime of about ≈ 15 min. Observation of a nonzero nEDM at the presently obtainable experimental uncertainty would be evidence of new physics since the predicted contribution of $\mathcal{CP}$-violation from the CKM matrix [10] in the Standard Model is 10^{-32}–$10^{-31} e\,\mathrm{cm}$, several orders of magnitude below current sensitivity limit of $|d_n| < 1.8 \times 10^{-26} e\,\mathrm{cm}$ (90% CL) [1]. Discovery of new physics would address the strong $\mathcal{CP}$ problem within the Standard Model or discover physics beyond the Standard Model [13, 20, 11]. The strong $\mathcal{CP}$ problem refers to the $\mathcal{CP}$-violating term in the QCD Lagrangian, $|\theta_{QCD}| < 10^{-10}$, which is already strongly suppressed below the natural value of order unity by the current limit on the nEDM.

Since the first proposal of $\mathcal{T}$ violation with the constraint (from scattering experiments) $d_n < 3 \times 10^{-18} e\,\mathrm{cm}$ [22], and subsequent measurement at ORNL [23] the limit has improved over 8 orders of magnitude to the current sensitivity. Several experiments are being developed worldwide to improve the limit to the level of $10^{-27} e\,\mathrm{cm}$: The LANL EDM experiment in the US, the PanEDM experiment at ILL, the TUCAN experiment at TRIUMF, and the n2EDM experiment at PSI. The n2EDM and LANL experiments use existing solid deuterium Ultra Cold Neutron (UCN) production facilities. The n2EDM experiment uses a completely redesigned magnetic shield and UCN precession apparatus based on experience with the Sussex experiment moved from ILL to PSI, which produced the present limit. The LANL experiment will use a first-generation apparatus being developed to take advantage of their greatly improved UCN source.

The quest for 10× improvement in all four nEDM experiments requires higher flux UCN sources and larger precession chambers to increase the statistics, and unprecedented control of the magnetic field uniformity. Instead of multilayer magnetic shields wrapped directly around the experimental chamber in previous experiments, all four new experiments employ magnetic shielding rooms (MSRs) [26] modeled after the PTB BMSR-2 facility [7]: first the TUM MSR [4] (by Imedco) being partially used in PanEDM, followed by the 6-layer PSI MSR [6] (by VAC), and finally two similar shields (by MSL) for the LANL and TUCAN experiments. The new shields have carefully designed B_0, gradient, trim, and spin transport coils with extensive field mapping campaigns to establish field (and thus spin precession) uniformity. They also rely on increasing numbers of optical magnetometers placed around the precession chamber to monitor gradients and time-dependence of the magnetic field: all but the PanEDM experiment plan to use Hg-199 as a comagnetometer.

The experiment is being carried out by the EDM collaboration of institutions from Belgium, England, France, Germany, Poland, Serbia, Switzerland, and the United States. The design is very mature, rooting back 20 years with the proposal to take data with the modified ILL apparatus on the PSI UCN source before subsequent upgrades to the apparatus. The final Conceptual Design Report (CDR) was published in Jan 2017, and the Technical Design Report (TDR) in Jan 2019. In spite of delays due to COVID, almost all major components of the experiment have been constructed and tested, culminating in the detection of polarized UCNs transported

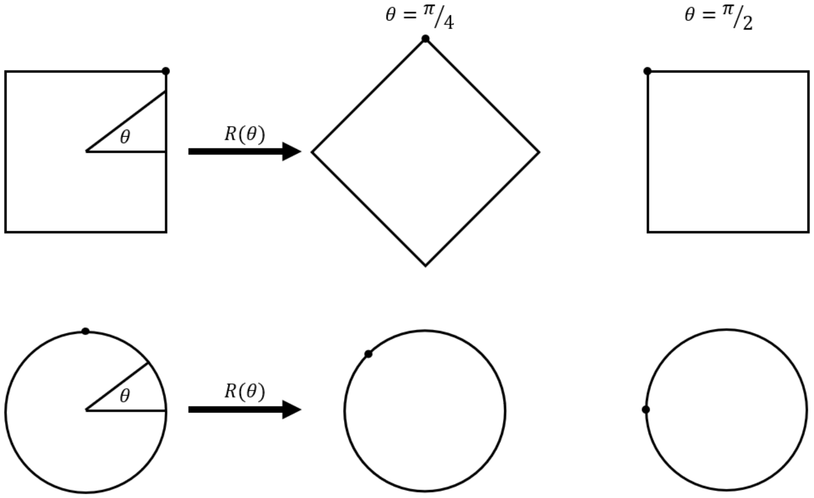


Figure 1.1: Preserving a square under rotation means that the vertices of the square map to the same locations after the rotation is performed. In the case of a $\pi/4$ rotation, the vertices are not the same as before the rotation and are, therefore, not symmetric under this rotation. However, rotation by $\pi/2$ has the same vertex locations and is therefore symmetric. Because there are only a finite number of angles that will preserve the squares' vertices under a rotation, that set of rotations is a discrete set. All points on a circle will remain on the circle after any arbitrary rotation. In that sense, rotations are a continuous symmetry for circles.

and stored in the B_0 field of the precession chamber. The experiment is expected to collect physics data during the 2024 beam cycle.

1.1.2 Symmetries

Symmetry is described as an action performed on an object that leaves the object unchanged. Symmetries are object-dependent, meaning an action performed on one set of objects could be symmetric while the same action performed on another set is not. This is seen by comparing squares and circles. Any rotation angle applied to a circle leaves the circle unchanged. However, preserving a square under rotation depends on the angle (See Figure 1.1). Symmetry, in this context, refers to the behavior of groups, specifically the relation between the group action and its elements. The example of the square vs the circle objects demonstrates two different types of symmetry: continuous and discrete. Continuous symmetries are actions characterized by a continuous variable (Which can be characterized by a Lie Group). Discrete symmetries are actions that can be indexed by counting numbers (Called Finite Groups)[9].

Symmetries play a fundamental role in studying and understanding physics. To make the case, one need only look at Noether's theorem, which states that for every continuous symmetry, there exists an associated conserved quantity [16, 19, 15]. Noether's theorem shows the connection between the invariance of the Lagrangian under spatial, temporal, and reference angle translations and the associated conserved quantities of momentum, energy, and angular momentum, respectively[15]. For discrete symmetries, Noether's theorem does not hold. However, discrete symmetries are of great importance to the Standard Model (SM). Specifically, Parity $\mathcal{P}$, Charge Conjugation $\mathcal{C}$, and Time Reversal $\mathcal{T}$ characterize the nature of fundamental forces and allowed particle reactions, and can indicate their role (for example, of the weak force) in effective interactions (for example, the hadronic interaction).

	$\mathcal{P}$	$\mathcal{T}$	$\mathcal{C}$
Coordinate ($\vec{\boldsymbol{x}}$)	$\vec{\boldsymbol{x}} \to -\vec{\boldsymbol{x}}$	$\vec{\boldsymbol{x}} \to \vec{\boldsymbol{x}}$	$\vec{\boldsymbol{x}} \to \vec{\boldsymbol{x}}$
Time (t)	$t \to t$	$t \to -t$	$t \to t$
Momentum ($\vec{\boldsymbol{p}}$)	$\vec{\boldsymbol{p}} \to -\vec{\boldsymbol{p}}$	$\vec{\boldsymbol{p}} \to -\vec{\boldsymbol{p}}$	$\vec{\boldsymbol{p}} \to \vec{\boldsymbol{p}}$
Energy (E)	$E \to E$	$E \to E$	$E \to E$
Angular Momentum ($\vec{\boldsymbol{J}}$)	$\vec{\boldsymbol{J}} \to \vec{\boldsymbol{J}}$	$\vec{\boldsymbol{J}} \to -\vec{\boldsymbol{J}}$	$\vec{\boldsymbol{J}} \to \vec{\boldsymbol{J}}$
Spin ($\vec{\boldsymbol{S}}$)	$\vec{\boldsymbol{S}} \to \vec{\boldsymbol{S}}$	$\vec{\boldsymbol{S}} \to -\vec{\boldsymbol{S}}$	$\vec{\boldsymbol{S}} \to \vec{\boldsymbol{S}}$
Charge (Q)	$Q \to Q$	$Q \to Q$	$Q \to -Q$
Electric Field ($\vec{\boldsymbol{E}}$)	$\vec{\boldsymbol{E}} \to -\vec{\boldsymbol{E}}$	$\vec{\boldsymbol{E}} \to \vec{\boldsymbol{E}}$	$\vec{\boldsymbol{E}} \to -\vec{\boldsymbol{E}}$
Magnetic Field ($\vec{\boldsymbol{B}}$)	$\vec{\boldsymbol{B}} \to \vec{\boldsymbol{B}}$	$\vec{\boldsymbol{B}} \to -\vec{\boldsymbol{B}}$	$\vec{\boldsymbol{B}} \to -\vec{\boldsymbol{B}}$

Table 1.1: Physical quantities under $\mathcal{C}$, $\mathcal{P}$, and $\mathcal{T}$ transformations [14]

Charge Conjugation $\mathcal{C}$

Charge conjugation is the operator that takes particles p to their corresponding anti-particles $\bar{p}$ or vise-versa.

$$\mathcal{C}p = \bar{p} \tag{1.1}$$

In this sense, $\mathcal{C}$ acting on a particle changes the sign of all internal quantum numbers (baryon number, lepton number, strangeness, etc...)[16]. From Equation 1.1, it can be seen that the application of $\mathcal{C}$ twice results in an unchanged system, so that the eigenvalues are ±1. Eigenstates of $\mathcal{C}$ are particles that are their own anti-particle (like photons). $\mathcal{C}$ symmetry is not directly tested in the nEDM experiment; however, the combined symmetry $\mathcal{CP}$ is tested. $\mathcal{C}$ symmetry is particularly important in studying baryon asymmetry (See the Sakharov Conditions in Section 1.1.3).

Parity $\mathcal{P}$

Parity, also called space inversion, is the operator that reflects all points through the origin.

$$\mathcal{P}\boldsymbol{x} = -\boldsymbol{x} \tag{1.2}$$

Like C, $\mathcal{P}$ returns to the original state when acted on a state twice. This results in the eigenvalues of ±1. The eigenstates of $\mathcal{P}$ are less restrictive than those of C; most particles are eigenstates of $\mathcal{P}$[16]. $\mathcal{P}$ is violated by the nEDM, as seen in the Hamiltonian ($\mathcal{H}$) for the interaction between the dipole moment ($\vec{\boldsymbol{d}}_n \propto \vec{\boldsymbol{S}}$) and the electric field ($\vec{\boldsymbol{E}}$)

$$\mathcal{H} = \vec{\boldsymbol{d}}_n \cdot \vec{\boldsymbol{E}} \tag{1.3}$$

This is violated because $\mathcal{P}\vec{\boldsymbol{E}} = -\vec{\boldsymbol{E}}$ and $\mathcal{P}\vec{\boldsymbol{S}} = \vec{\boldsymbol{S}}$ (See Table 1.1), therefore $\mathcal{PH} = -\mathcal{H}$[8]. Thus, a non-zero value for the nEDM results in parity violation.

Time Reversal $\mathcal{T}$

Time inversion describes the process of taking $t \to -t$. This definition is incomplete, as explicitly seen in quantum mechanical momentum and energy. The momentum operator is defined as $\hat{\boldsymbol{p}} = -\boldsymbol{i}\hbar\partial_{\boldsymbol{x}}$ and classical momentum is defined as $\vec{\boldsymbol{p}} = m\partial_t\vec{\boldsymbol{x}}$; if $\mathcal{T} = t \to -t$ then the momentum operator $\hat{\boldsymbol{p}}$ does not change signs while classical momentum $\vec{\boldsymbol{p}}$ does. Additionally, the Schrodinger equation, $\boldsymbol{i}\hbar\partial_t\psi = \hat{\mathcal{H}}\psi$, with the same definition of time inversion $\mathcal{T}$ applied results in energy changing signs. Table 1.1 shows that energy is invariant under time inversion[1] meaning a change is needed in the definition of time inversion that leaves the classical equations unchanged and modifies their quantum counterparts. This can be done by introducing an antilinear component, the complex conjugate[2]. Defining $\mathcal{T}$ to do both $t \rightarrow -t$ and $\boldsymbol{i} \rightarrow -\boldsymbol{i}$ fixes the sign issue for both the time-reversed momentum operator and Schrodinger's equation. It also switches initial $|i\rangle$ and final $\langle f|$ states in transition matrix elements, since $\langle f|M|i\rangle^* = \langle i|M^\dagger|f\rangle$.

For the nEDM, time reversal asymmetry is inherent to the Hamiltonian in Equation 1.3. Table 1.1 shows $\mathcal{T}\vec{\boldsymbol{S}} = -\vec{\boldsymbol{S}}$ and $\mathcal{T}\vec{\boldsymbol{E}} = \vec{\boldsymbol{E}}$, thus a non-zero dipole moment d_n for the neutron would result in time-reversal symmetry to be violated.

$\mathcal{CPT}$

Currently, $\mathcal{CPT}$ symmetry is held to be a fundamental symmetry of the Standard Model[24]. That is, the application of each operation $\mathcal{C}$, $\mathcal{P}$, and $\mathcal{T}$ in any order should leave the Hamiltonian unchanged. This would mean that if $\mathcal{CP}$ is violated, then $\mathcal{T}$ must also be violated to preserve the overall symmetry of the Hamiltonian[8]. CPT-theorem and Lorentz symmetry (continuous symmetry under translations, rotations, and boosts) are general features of all quantum field theories.

1.1.3 $\mathcal{CP}$ Violation and The Sakharov Conditions

$\mathcal{CP}$ violation is a built-in feature of the SM; this is characteristically seen in electroweak interactions to the extent that it is ubiquitous in electroweak reactions via the complex phase in an off-diagonal element of the Cabibbo–Kobayashi–Maskawa (CKM) matrix. There is potential for $\mathcal{CP}$ violation to occur in Quantum Chromody-

---

[1] this can also be seen in the classical definition of the Hamiltonian: $\mathcal{H} = \frac{\vec{\boldsymbol{p}}^2}{2m} - V(x)$. Where $V(x)$ does not change sign under time reversal and $\vec{\boldsymbol{p}}$ does, but its square does not, leaving $\mathcal{H}$ invariant.

namics (QCD)[2], the quantum field theory of the strong force. $\mathcal{CP}$ violation can occur in strong interactions if the coefficient θ is nonzero in the strong field Lagrangian of the form

$$\mathcal{L}_{QCD}^{eff} = \mathcal{L}_{QCD} + \theta \frac{g^2}{32\pi^2} \epsilon^{\mu\nu\alpha\beta} G^a_{\mu\nu} \tilde{G}^a_{\alpha\beta} \tag{1.4}$$

where $G^a_{\mu\nu}$ is the gluon-field strength tensor[17, 8, 3]. The consequence of such a term would be an EDM for the neutron. This would result in an upper bound of $d_n \approx 10^{-16} e \cdot cm$, taking θ to be of the order unity[12].

An observation of $\mathcal{CP}$ violation would have greater impact than mere discovery of either *theta* or physics beyond the SM. Finding a non-zero nEDM and subsequent $\mathcal{CP}$ violation would provide evidence of one of the Sakharov Conditions, which describe the necessary circumstances for baryon asymmetry to evolve in the early universe[3]. The conditions are as follows:

- Violation of baryon number conservation
- Violation of $\mathcal{C}$ and $\mathcal{CP}$
- Departure from equilibrium

These conditions state the need for a process for creating baryons, particles to be treated differently from anti-particles, and a future that is different from its past[25].

1.1.4 Measuring the nEDM

Measurement of the nEDM relies on precise estimation of the precession frequency (f_n) of polarized Ultra Cold Neutron (UCN)s stored in a weak magnetic field $\boldsymbol{B}_0$ and a strong electric field $\boldsymbol{E}$. The nEDM is obtained by comparing the precession frequencies for parallel(↑↑) and anti-parallel(↑↓) configurations of the magnetic and

---

[2] As of this writing, no known $\mathcal{CP}$ violation exists in quantum chromodynamics, which is called the strong $\mathcal{CP}$ problem.

[3] Baryon Asymmetry refers to the imbalance of matter and antimatter in the universe.

electric fields[5]

$$d_n = \frac{\pi\hbar}{2|\boldsymbol{E}|}(f_{n,\uparrow\downarrow} - f_{n,\uparrow\uparrow}), \tag{1.5}$$

where f_n can be expressed in terms of the magnetic field $\vec{\boldsymbol{B}}_0$, the electric field $\vec{\boldsymbol{E}}$, the gyromagnetic ratio of the neutron γ_n, and the neutron dipole moment d_n:

$$f_n = \left|\frac{\gamma_n}{2\pi}\vec{\boldsymbol{B}}_0\right| \mp \frac{d_n}{\pi\hbar}|\vec{\boldsymbol{E}}| \tag{1.6}$$

The "+" sign is for the anti-parallel configuration and the "-" sign is for the parallel configuration. Measurement of f_n is done in a multi-step process.

1. UCNs are produced by the full PSI proton bean. Protons are fired at the UCN source for 8 s every 5 min.

2. The free neutrons are guided to the measurement chamber by the UCN transport system. The neutrons are polarized along the path by a 5 T superconducting magnetic to near 100% polarization.

3. Once the chambers are filled with polarized neutrons, the shutters close, and Ramsey's method of separated rotations fields is performed: an RF pulse rotates the spin vectors by an angle of $\pi/2$ into the plane perpendicular to B_0, after which the neutrons begin precessing about B_0. After a fixed amount of time, the spin is rotated back along the B_0 field by a second $\pi/2$ RF pulse. The optimal Ramsey cycle is about 184 s long[5], which is a compromise between the accumulation of extra phase, and less signal due to neutron decay.

4. The UCNs are released from the precession chamber by opening the shutters and are guided to the spin analyzer. Each chamber is connected to a dedicated spin-sensitive detector. The devices simultaneously and independently count the number of neutrons in the two spin states. The polarization of the neutrons is sensitive to the phase of precession at the time of the 2nd RF pulse, which is

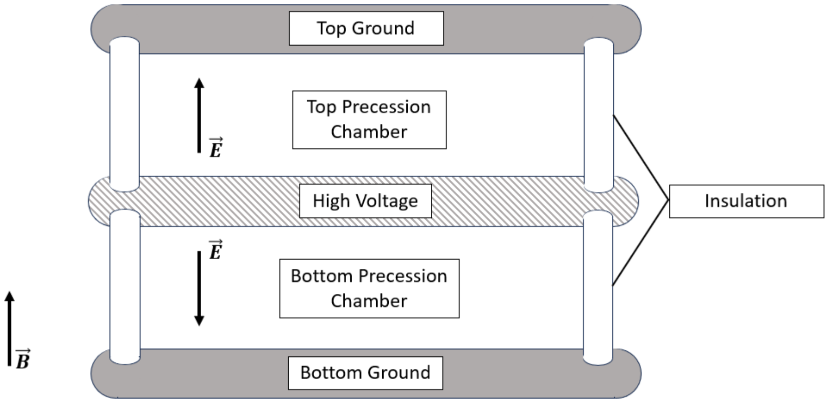


Figure 1.2: Spin Precession Chambers with field orientation

a measure of the electric and magnetic dipole moments. If spins were oriented along the rotation axis of the $\pi/2$ pulse, the polarization would be zero; if they were perpendicular to this axis, the polarization would be maximal.

1.1.5 n2EDM Design

The design of the second generation of nEDM experiments, called n2EDM, was focused on maximizing the counting statistics while keeping the systematic effects minimal. This led to improvements in the UCN storage, the UCN transport system, and the magnetic field environment in the chamber. To increase the counting statistics, the volume of the UCN storage cells was increased to accommodate more neutrons per cycle. The single precession chamber was replaced by two chambers with opposite electric fields. The spin transport system was optimized to yield a high spin polarization post-transit. The magnetic field in the chamber is controlled with a high degree of precision via a main coil and an additional 70 trim coils, each powered by stable power supplies.

The basic apparatus in the vacuum chamber, shown in Figure 1.2, consists of two stacked cylindrical storage chambers where the Ramsey spin-precession is to be measured. Central to the two chambers is a high-voltage electrode with a corresponding ground electrode at the top and bottom of the cylinders. An insulating ring confines the cylinder's radial edges. This arrangement produces an electric field in opposite directions for the top and bottom chambers as required by Equation 1.5. This allows for the chambers to measure simultaneously the parallel and anti-parallel configurations of fields at the same time, and the double chamber can accommodate twice as many neutrons per cycle. The cylinder is placed inside the central chamber with a uniform upward-pointing magnetic field. The magnetic field is monitored by a mercury comagnetometer in the precession cells and an array cesium magnetometers around the cells. The magnetic field is the second component of Equation 1.5 in orienting the parallel and anti-parallel directions and in generating the precession frequencies.

The portion of the design dealt with in this dissertation is the spin transport system. As such, the focus will be on maximizing spin polarization as the neutrons pass through the experiment from the polarizer to the precession chambers. This is accomplished by the design of guide coils that maximize the spin transport adiabaticity, which is discussed in Section 1.2. The statistical sensitivity of the measured nEDM is

$$\sigma(d_n) = \frac{\hbar}{2\alpha E T \sqrt{N}} \tag{1.7}$$

, where the visibility α of the Ramsey Cycle depends on the UCN polarization, the analyzing power, and the T_2 time (field uniformity); T is the precession time; and N is the total number of neutrons present in both the top and bottom components of the precession chamber. Thus maximizing spin polarization in the chambers leads to a linear increase in sensitivity versus, for example, $\sqrt{N}$ by increasing the number of neutrons.

1.2 Magnetic Field Adiabaticity

Adiabaticity is a measure of the smoothness of changes in the magnetic field as it relates to maintaining spin polarization. To ensure the spin follows the direction of the magnetic field, the rate at which the field changes direction (ω_B, discussed in Section 1.2.1) needs to be much smaller than the rate at which the spin processes around the field (the Larmor frequency ω_L, discussed in Section 1.2.2). Adiabaticity is the ratio of the two frequencies:

$$\alpha^{-1} = \frac{\omega_B}{\omega_L} = \frac{\left|\left(\boldsymbol{v} \cdot \overrightarrow{\nabla}\right) \boldsymbol{B} \times \boldsymbol{B}\right|}{\gamma B^3}, \tag{1.8}$$

where v is the particle's speed (neutron in this case) and γ is the gyromagnetic ratio (again for the neutron in this case). The goal then becomes to maximize the ratio ω_L/ω_B to preserve the highest level of spin polarization.

1.2.1 Field rotation

To quantify how fast the magnetic field is changing compared to the rate at which the particle is precessing, the magnetic field is broken into components, $\vec{\boldsymbol{B}}_\perp$ being perpendicular to the axis of spin, and $\vec{\boldsymbol{B}}_\parallel$ being parallel to the axis of spin. Varying $\vec{\boldsymbol{B}}_\parallel$ only affects the precession rate; it does not change the axis the spin is oriented about. To change the spin axis, $\vec{\boldsymbol{B}}_\perp$ must vary. In Figure 1.3, the change of $\vec{\boldsymbol{B}}_\perp$ is associated with the rate of change of $\frac{\Delta\theta}{\Delta T}$. $\boldsymbol{B}_\perp$ can be expressed as

$$\Delta \boldsymbol{B}_\perp = \frac{|\Delta \vec{\boldsymbol{B}} \times \vec{\boldsymbol{B}}|}{B}$$

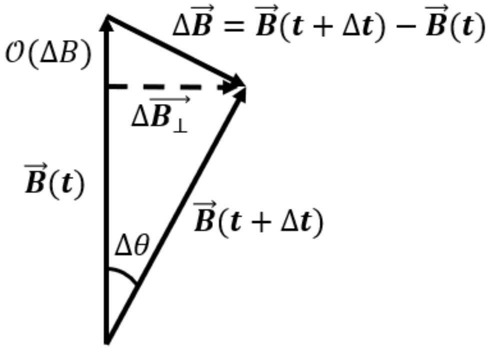


Figure 1.3: Changing perpendicular field is characterized by angle θ. From this $\tan(\Delta\theta) = \Delta\vec{B}_\perp/(B - \mathcal{O}(\Delta B))$, taking $\Delta\theta$to be small this simplifies to $\Delta\theta = \Delta\vec{B}_\perp/B$.

This yields the magnitude of the perpendicular component; the direction is irrelevant.

Combining this expression and the relation for $\Delta\theta$ it is seen that

$$\Delta\theta = \frac{\Delta B_\perp}{B} = \frac{|\Delta\vec{B} \times \vec{B}|}{B^2}$$

This can be divided by some small time interval Δt and take the limit as $\Delta t \to 0$

$$\lim_{\Delta t \to 0}\left(\frac{\Delta\theta}{\Delta t} = \frac{\left|\frac{\Delta\vec{B}}{\Delta t} \times \vec{B}\right|}{B^2}\right)$$

Looking at each side separately

$$\lim_{\Delta t \to 0}\frac{\Delta\theta}{\Delta t} = \omega_B$$

where ω_B is the angular frequency of the magnetic field change and

$$\lim_{\Delta t \to 0}\frac{\Delta\vec{B}}{\Delta t} = \frac{d\vec{B}}{dt} = \partial_t\vec{B} + \left(\frac{d\vec{x}}{dt}\cdot\vec{\nabla}\right)\vec{B}$$

This results in

$$\omega_B = \frac{\left|\left(\partial_t\vec{\boldsymbol{B}} + (v \cdot \vec{\nabla})\vec{\boldsymbol{B}}\right) \times \vec{\boldsymbol{B}}\right|}{B^2}$$

Because the neutrons are moving through a static field (no time-varying component), the term $\partial_t\vec{\boldsymbol{B}}$ vanishes, leaving

$$\omega_B = \frac{\left|\left(\boldsymbol{v} \cdot \vec{\nabla}\right)\boldsymbol{B} \times \boldsymbol{B}\right|}{B^2} \tag{1.9}$$

1.2.2 Larmor Precession

Larmor precession arises as a result of the torque on a magnetic moment μ inside a magnetic field $\vec{\boldsymbol{B}}$

$$\vec{\boldsymbol{\tau}} = \vec{\boldsymbol{\mu}} \times \vec{\boldsymbol{B}} \tag{1.10}$$

Because, by the Wigner-Eckart theorem, the magnetic moment must be proportional to the spin angular momentum ($\vec{\boldsymbol{\mu}} = \gamma\vec{\boldsymbol{S}}$), the result of the torque on the magnetic moment is for the moment to precess around the magnetic field. Torque is classically defined as the rate of change of the angular momentum $\vec{\boldsymbol{\tau}} = d\vec{\boldsymbol{S}}/dt$; since the torque is perpendicular to the field, $d\vec{\boldsymbol{S}}/dt$ describes the rate at which the spin perpendicular to the magnetic field rotates around $\vec{\boldsymbol{B}}$. Taking the angle between the spin and magnetic field to be θ,

$$\partial_t\vec{\boldsymbol{L}} = \frac{S\sin\theta\Delta\phi}{\Delta t} = S\omega_L\sin\theta \tag{1.11}$$

where ϕ is taken to be the angle in the plane perpendicular to $\vec{\boldsymbol{B}}$, the rate of change of ϕ is the precession/Larmor frequency. Combining Equations 1.10 and 1.11 via the definition of classical torque, the result is

$$\begin{aligned}\vec{\boldsymbol{\tau}} &= \partial_t \vec{\boldsymbol{L}} \\ \vec{\boldsymbol{\mu}} \times \vec{\boldsymbol{B}} &= S\omega_L \sin\theta \\ \mu B \sin\theta &= S\omega_L \sin\theta \\ \gamma S B &= S\omega_L\end{aligned}$$

$$\omega_L = \gamma B \tag{1.12}$$

1.3 Magnetic Scalar Potential and Coil Construction

The standard method of creating bespoke magnetic fields starts with a known coil that produces a field close to what is desired. Then, through adding incremental change, modeling, and repeating, the field will converge to the target field. This method is time-consuming, and its precision is limited by luck or the amount of time one is willing to throw at the problem. There are other methods, but for brevity, the process that produces the best results in the shortest time is the utilization of the Magnetic Scalar Potential (MSP).

The MSP method begins with the desired magnetic field, constrained only by Maxwell's Equations, and then calculates the wire geometries needed to produce the field. This reduces the need to iterate the coil design. Iteration is present only in current discretization. Due to the calculative nature of the discretization process, any guesswork is effectively eliminated.

For a MSP to be constructed, the region of the desired magnetic field, called the target region, must contain no current. The zero current constraint results from a general condition on scalar potential construction. For a scalar field U to exist describing a general field $\boldsymbol{F}$ such that $\boldsymbol{F} = -\vec{\nabla} U$, it is required that $\vec{\nabla} \times \boldsymbol{F} = 0$.

Comparing the definition of a scalar field and Maxwell's Equations for magnetic fields with no currents

$$\overrightarrow{\nabla} \times \boldsymbol{H} = 0 \tag{1.13}$$

$$\overrightarrow{\nabla} \cdot \boldsymbol{B} = 0 \tag{1.14}$$

along with the relation $\boldsymbol{B} = \mu \boldsymbol{H}$, the result is that Eq. 1.13 meets the requirement for a MSP to be constructed. Thus, a potential $U_T(\overrightarrow{\boldsymbol{r}})$ can be constructed such that

$$\boldsymbol{H}_T(\overrightarrow{\boldsymbol{r}}) = -\overrightarrow{\nabla} U_T(\overrightarrow{\boldsymbol{r}}), \tag{1.15}$$

where the subscript T denotes the target region. At a glance, the restriction of no current appears to be a significant limitation. However, considering the role typically held by magnetic fields in experiments, it's not as restrictive. In most experiments, the target region houses particles or apparatuses or is defined by having a uniform magnetic field (all being the case for the n2EDM); under these conditions, it is required that no current-carrying wires are present in the target region. These fields are created by surface-current magnets, in which all of the currents are placed on the surfaces between target and flux return regions.

Generally, the magnetic field normal to the target region's surface cannot be canceled due to conservation of magnetic flux. The normal component of $\boldsymbol{B}$ (denoted $\boldsymbol{B}_n$) that permeates through the target region boundary can cause an extension of the magnetic system, albeit magnetizing objects exterior to the coil or interference with electronic devices. To mitigate the effects of the magnetic leakage, it is necessary to introduce a return region to further confine the fields. Most commonly, the return region is defined so that no magnetic fields extend beyond the return's boundary. However, experimental constraints may call for the boundary to be defined by specified currents or some combination of the two boundary conditions. Inclusion of the return region requires an additional third infinitesimal "region", called the transition

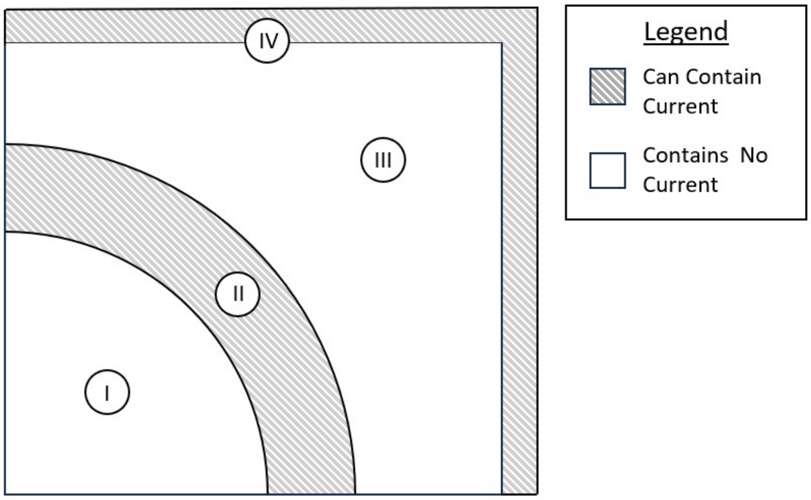


Figure 1.4: This figure details the spatial relation of the three regions and the boundary. I.) indicates the target region where the magnetic field is specified. II.) Is the transition region where all interior current is confined. This is not to scale, considering it is infinitesimally small. III.) denotes the return region. IV.) shows the boundary of the return region and explicitly shows where additional coils may be added to confine the field.

region (See Figure 1.4), that sits between the target and return regions. The defining property of the transition region is that the $\boldsymbol{H}$ field is normal at the boundary shared between the target and transition regions and at the boundary shared between the transition and return regions; thus, $\boldsymbol{H}$ has no tangential component in the transition region, $\boldsymbol{H}_t = 0$. $\boldsymbol{H}_t$ being zero means that the scalar potential is constant on the boundary surfaces in the transition region and, as such, can be set to zero by taking the region to be arbitrarily small and setting the constant of integration appropriately. In this way, the region is considered conceptual. As a result of how the transition region was defined, it has allowed for no further modeling or study of the region. For clarity, with the constraints on the target and return containing no currents, this leaves the only areas having current being the two boundaries of the transition region, which is the coil being solved for, and the exterior of the transition region when boundary conditions allow for them.

Before physical wire construction can be considered, the boundary conditions for the return region need more detail. First, $\boldsymbol{B}_n$ on the boundary between the return and the target must be the same. The transition region is excluded by construction; the region is infinitesimal, and the fields are normal to the surfaces. This allows the region to be skipped over when considering $\boldsymbol{B}_n$. Continuity of $\boldsymbol{B}_n$ results from Eq. 1.14:

$$\Delta\hat{n} \cdot \boldsymbol{B} = \Delta\partial_n (\mu U) = 0 \tag{1.16}$$

For a field confined to the extent of the return region, $\boldsymbol{B}_n$ must also vanish at the exterior boundary. As stated before, this boundary condition on the exterior can be substituted for a constraint of specified surface currents; this is done by specifying the MSP on the surface for the corresponding currents. Combinations of the two boundary conditions also yield unique solutions to the Boundary Value Problem (BVP). However, specifying current on the boundary or mixing boundary conditions comes at the cost of fields leaking out of the return region and modifying the specified target field. This is cause for more attention when modeling and will result in more iterations to compensate.

Wire geometry and current define a coil. For a scalar potential to define a coil, it needs a mechanism to determine the current and geometries that produce the desired field. To find the current start with Ampere's Law.

$$\oint \boldsymbol{H} \cdot d\boldsymbol{\ell} = I \tag{1.17}$$

Taking the Amperian loop around the boundary of the target and transition regions as denoted in Figure 1.5), the result that

$$\Delta U = U_2 - U_1 = I \tag{1.18}$$

emerges naturally, since $\Delta U = 0$ in the transition region (half of the Amperian loop).

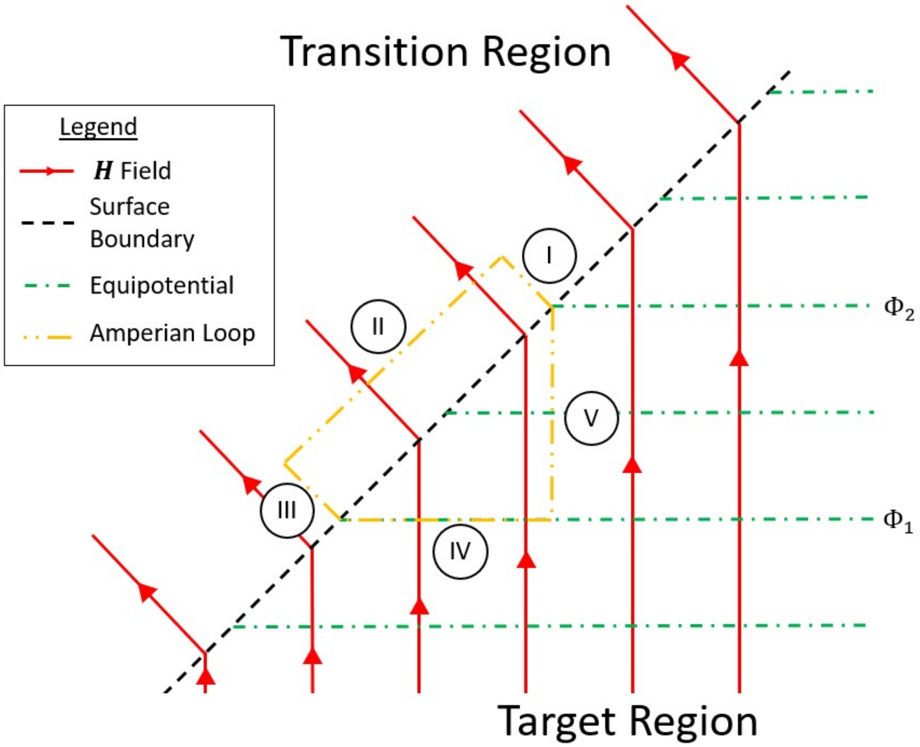


Figure 1.5: The figure shows the dynamics of the magnetic field between the target and transition regions. Also featured is the Amperian loop denoted by segments I-V. From this loop, paths II and IV have zero contribution because they lie perpendicular to the field. A small enough loop means $\boldsymbol{H}_n$ along path I $\approx$ $\boldsymbol{H}_n$ along path II; thus, paths I and II form a reciprocal pair and cancel. Leaving path V as the only contribution, which is $\int H d\ell = \int \left(\overrightarrow{\nabla} U \cdot \hat{\boldsymbol{\ell}}\right) d\ell = U_2 - U_1$.

This is significant because it directly relates the magnetic scalar field to the current. The equipotential surfaces of the MSP inform the geometry of the coils as well. The equipotential surfaces that correspond to integer multiples of a small constant called δU, define a set of potential sheets. These potential sheets cannot intersect; due to this, the paths the sheets follow on the boundary surface make distinct closed contours, with ribbons on the boundary in between adjacent contours. The current in each ribbon is equal to , which is seen from Eq. 1.18. These ribbons, when discretized into wires, create the geometry of the coils. Another way to arrive at this result is to begin with the the continuity equation resulting from Eq. 1.13

$$\hat{n} \times \boldsymbol{H} = \hat{n} \times \overrightarrow{\nabla} U = \boldsymbol{K} \tag{1.19}$$

Integrating along the surface of the right-hand side is equivalent to Amperes law. Using Eq. 1.19 comes with the benefit that the direction of the current is not lost in a dot product as is the case in Amperes law. Additionally, the surface current $\boldsymbol{K}$ is inherently more physical than conceptualizing equipotential surfaces. Dividing the surface into ribbons along the flow lines of $\boldsymbol{K}$ is intuitive. However, the direct relation between the MSP and the current (Eq. 1.18) is more abstract and makes the connection a logical leap.

All that is needed to determine the coil is the explicit definition of the MSP on the boundary surface. From Eq. 1.15, part of the problem has already been solved. The same method applies to the return region, given the return also contains no current, and a scalar potential $U_R(\overrightarrow{r})$ can be constructed such that

$$\boldsymbol{H}_R(\overrightarrow{\boldsymbol{r}}) = -\overrightarrow{\nabla} U_R(\overrightarrow{\boldsymbol{r}}) \tag{1.20}$$

where the subscript R denotes the return region. $U_R(\overrightarrow{\boldsymbol{r}})$ is determined by the boundary conditions present on the exterior boundary of the return region and the boundary shared with the transition region. The exterior boundary conditions vary on the experimental setup, but the boundary shared with the transition region is subject to the continuity equation 1.16. Once $U_R(\overrightarrow{\boldsymbol{r}})$ has been solved for, a coil can be constructed using the method described resulting from Eq. 1.18 and Eq. 1.19 on the boundary of the transition and return regions. Cases persist where coils resulting from $U_R(\overrightarrow{\boldsymbol{r}})$ and $U_T(\overrightarrow{\boldsymbol{r}})$ should be wound separately, like when multiple regions connect at once or with end caps. When, however, the return and target regions are on adjacent sides of the boundary, it is far easier to combine the potentials on the boundary into one such that $U_B(\overrightarrow{\boldsymbol{r}}) = U_T(\overrightarrow{\boldsymbol{r}}) - U_R(\overrightarrow{\boldsymbol{r}})$. Again, the construction method laid out in Eq. 1.18 and Eq. 1.19 for the combined field $U_B(\overrightarrow{\boldsymbol{r}})$ still works and results in a coil that has the most straightforward physical construction.

Copyright© David C. Bowles, 2024.

Chapter 2 Target Magnetic Field Design

Spin polarization is essential in minimizing the error in the neutron Electric Dipole Moment (nEDM) measurement; see Eq. 1.7. Per the experimental design, the neutrons are initially polarized along the $\hat{x}$ direction by a superconducting magnet (SCM). However, the final spin polarization inside the instrument chamber must be in the $\hat{z}$ direction. To accomplish this change, it is necessary to have a Spin Transport Coil (STC) that smoothly rotates neutron spin to the desired direction. The goal is to design a coil that produces a magnetic field that rotates the spin while maximizing the polarization.

The environment surrounding the coil must be defined to create the STC. The critical components determining the design of the STC are the Magnetic Shield Room (MSR), the SCM, and the background magnetic field interior to the MSR called the $\boldsymbol{B}_0$-coil. From the environment, the target region's magnetic field can be found to maximize the adiabaticity and adhere to the field constraints discussed below. This was accomplished using a linear combination of magnetic scalar potentials on the surface of the STC.

2.1 Background Magnetic Field And The MSR

Adiabaticity is calculated using the Total Magnetic Field (see Eq. 1.8). The Total Magnetic Field is the sum of the background ($\boldsymbol{B}_{BG}$) and the designed magnetic field in the presence of the MSP ($\boldsymbol{B}_{MSP}$). The background magnetic field consists of the SCM and the $\boldsymbol{B}_0$-coil. Both SCM and $\boldsymbol{B}_0$-coil's fields are influenced by the magnetic material of the MSR. The MSR is composed of six cubic shells of mu-metal with a quasi-symmetric pattern of access holes for the UCNs, vacuum pumps, high voltage, lasers, and other services to the experiment. The MSR makes for a

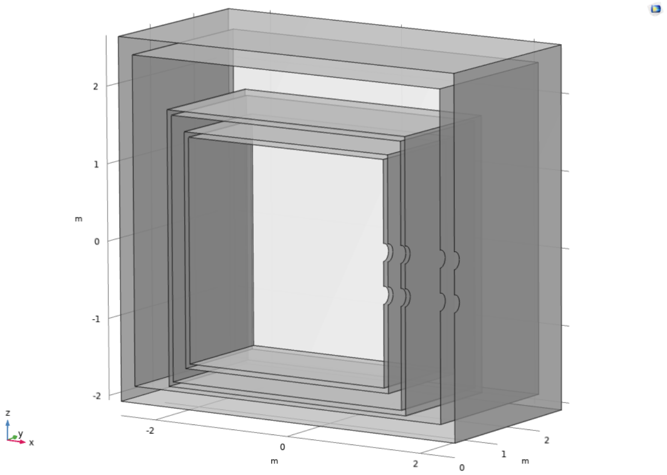


Figure 2.1: The MSR before any meshing. The six cubes are the layers of the μ-metal, and the divots on the right are the holes for UCN guides.

complicated environment to analytically calculate $\boldsymbol{B}_{BG}$. Instead, the magnetic fields were solved numerically using the Finite Element Analysis (FEA) software COMSOL Multiphysics (see B).

2.1.1 The MSR

The MSR is at the center of the design of the STC. As stated above, the MSR comprises six μ-metal cubic sheets with holes in the sides, allowing access for the spin transport system to the center of the room. The cubic sheets have varying thicknesses, physical extents, and centers (see A). One symmetry persists, as all objects are centered $y = 0$. This means the model is symmetric about the xz-plane ($\hat{x}$ taken to be along the axis of the STC). The overall lack of symmetry makes the problem of determining magnetic fields in this environment difficult, not to mention time-consuming, to solve analytically. This problem is complex even before accounting for the magnetic material. As stated above, using the FEA software (COMSOL) is far more efficient in solving for the fields surrounding the MSR. The model size is reduce by exploiting symmetry in the geometry, and manually reducing the mesh size close to the spin transport coils.

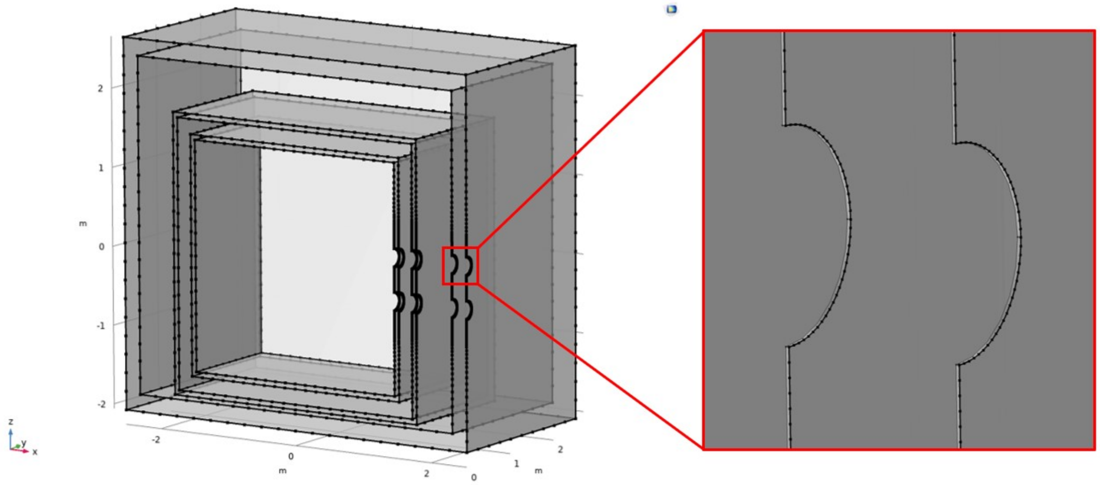


Figure 2.2: The MSR after the exterior edges are meshed. Additional nodes were added surrounding the UCN Guide because this is where the STC is designed.

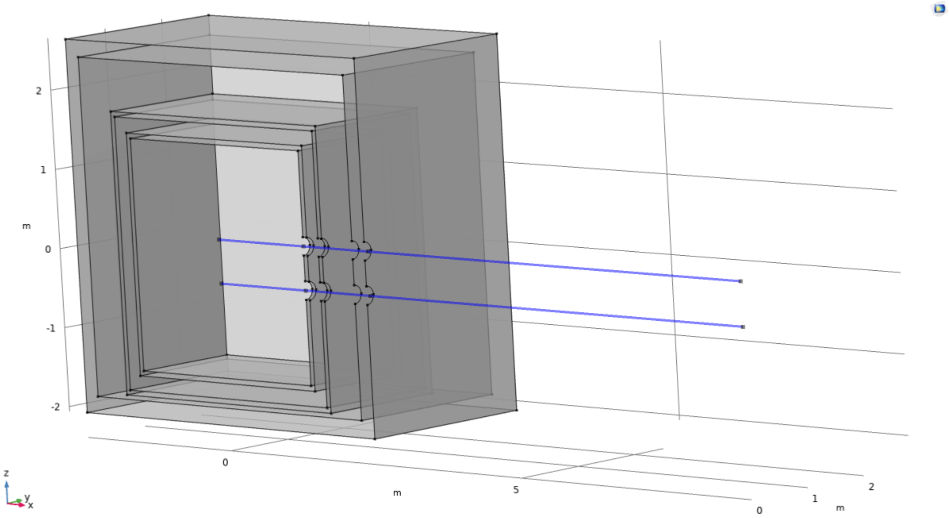


Figure 2.3: Two Meshing lines are added at the centers of the STCs, $z = \pm 275mm$. The mesh spacing on these lines sets the mesh's resolution in the surrounding domain. The adiabaticity along the center line is maximized; therefore, the field needs a higher resolution.

In Figure 2.1, the MSR is shown before any meshing has been done. The areas of interest are inside the MSR's inner volume surrounding the procession chamber, inside the STC along the UCN guide, and on the exterior of the MSR surrounding the UCN switch (See Table 2.1 for export domains), and should have a finer mesh surrounding them. Due to the thinness of the μ-metal layers and their interaction

Export Domain	axis	range
Interior	x	-1350 mm , 1350 mm
	y	0 mm, 1350 mm
	z	-1350 mm , 1350 mm
Top UCN Guide	x	1459 mm , 3000 mm
	y	0 mm , 200 mm
	z	175 mm , 375 mm
Bottom UCN Guide	x	1459 mm , 3000 mm
	y	0 mm , 200 mm
	z	-375 mm , -175 mm
Exterior	x	2522.5 mm, 4876.5 mm
	y	0 mm , 200 mm
	z	275 mm , 400 mm

Table 2.1: The volumes exported for every COMSOL model correspond to the targeted study volumes. Interior surrounds the precession chamber, the Top and Bottom UCN Guide are centered around ±275 mm, respectively and extend up and down by ±100 mm, and the Exterior surrounds the UCN switch and extends out to the face of the SCM.

with the $\boldsymbol{B}_{BG}$,the mesh quality on the surface of the μ-metal layers was also enhanced in these regions by setting the number of nodes on the box edges and increasing the number of nodes on the edges surrounding the UCN guide (see Fig. 2.2). This set the base level of detail in the mesh on the surface of the μ-metal layers and guaranteed a higher density of nodes surrounding the UCN guide. To get the level of detail to continue through the whole region in the UCN guides, lines were included from the center of the MSR, down the center of each guide, to the edge of the defined model (called the world limit) to set the maximum node distance (see Fig. 2.3). The line ran from the center to the edge for smooth node-distance transitions between regions. Once the edges were meshed, a *free triangular*[1] mesh element was added to each exterior face of each μ-metal layer (See Fig. 2.4). To complete the MSR mesh,

---

[1] *free triangular* is a meshing element defined in COMSOL that takes a surface and performs a polygonal triangulation on the surface based on first the *size*; another element that sets the detail of the meshing element for a region. Then, based on whether there are any pre-meshed edges that the surface is connected to.

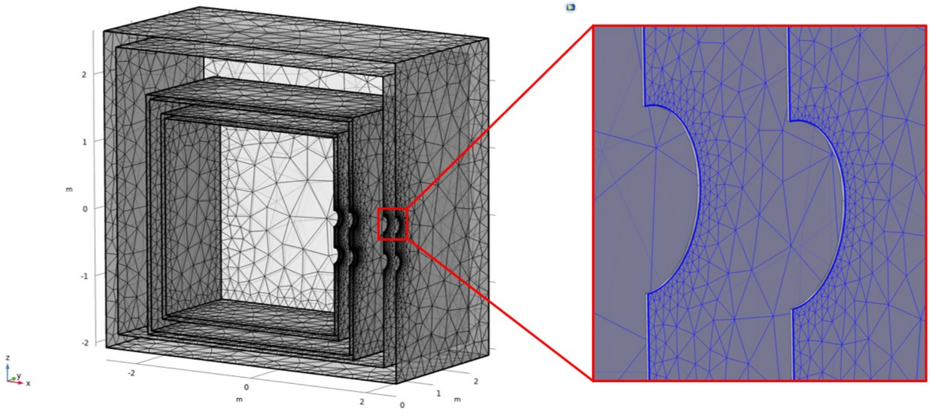


Figure 2.4: *Free Triangular* meshing element is added to the surface of the MSR. The resolution on the surface is reflected in the *distributions* shown in Fig. 2.2.

a *sweep* and *conversion*[2] was performed over the thickness of the μ-metal to turn the layers into fully meshed domains. The model was completed by setting the material of the MSR to *nickel steel MuMetal* in the COMSOL library, which sets all relevant physical quantities, such as the B-H curve.

Two boundary conditions were set to complete the rest of the environment. A symmetry plane was set to mirror all objects and currents across the xz-plane. This was done because of the symmetry about $y = 0$ noted above. The second condition set is that all fields go to zero at infinity. The model cannot be infinitely large. This is accomplished by defining two spheres, one inside the other. The inner sphere is the world volume for the model; all other objects are contained inside this domain. The domain between the smaller and larger spheres was set as an *infinite element domain*[3], which simulates the field in the limit as the outer radius goes to infinity. Setting these two boundary conditions creates a starting environment where all other models can be built.

---

[2] A *sweep* is a meshing element defined in COMSOL that takes a meshed starting face and makes copies of the mesh through an interval onto a destination face. Over the interval in which the copies were made, if there are any exposed faces not in the destination faces, left, then a semi-meshed face is left. This requires a *conversion* element to complete the mesh on the face.

[3] *infinite element domain* is an artificial domain that is set in the definitions component.

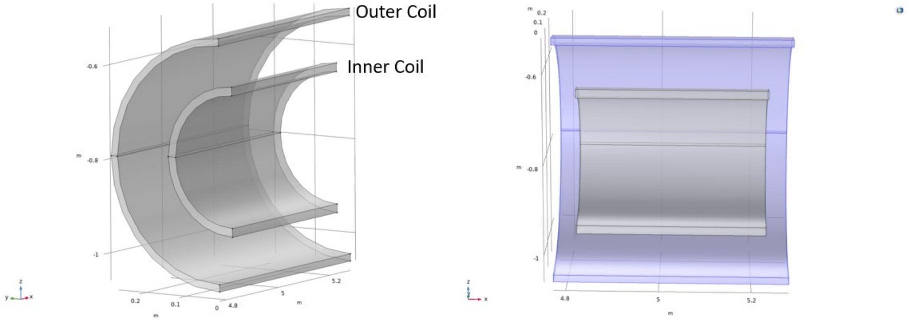


Figure 2.5: Layout of the SCM, which comprises two hollow cylinders with some edge thickness. The cylinders are referenced as Inner and Outer, shown on the left. The coils are coaxial, but the Inner Coil is shorter than the Outer Coil, shown on the right with the Outer Cylinder highlighted.

2.1.2 COMSOL Modeling of Background Magnetic Fields

As stated above, the background magnetic field, $\boldsymbol{B}_{BG}$, consists of two coils, a superconducting magnet (SCM),$\boldsymbol{B}_{SCM}$ that lies outside the MSR and the $\boldsymbol{B}_0$ coil wrapped around the precession chamber inside the MSR. In the limit of a linear B-H curve of the *mu*-metal, the superposition principle holds:

$$\boldsymbol{B}_{BG} = \boldsymbol{B}_{SCM} + \boldsymbol{B}_0 \tag{2.1}$$

This allows each coil to be modeled separately and the resulting fields summed. Modeling the two coils individually is helpful because each constitutes an intricate model with different meshing needs in other regions. Combining all coils resulted in a model that ran slow at best and, had stack overflows at worst.

The SCM spin polarizer consists of two superconducting coils. The two coils were modeled in COMSOL using two centered hollow cylinders with a small thickness, with the interior coil having a shorter axial length than the outer one (See Fig. 2.5). As was the case for the MSR, the SCM needed a high level of detail surrounding the

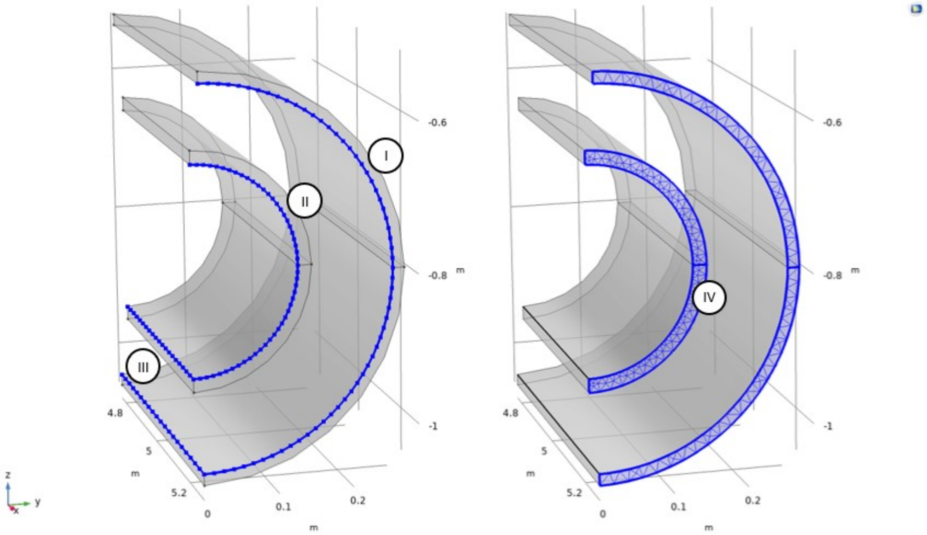


Figure 2.6: Meshing of the SCM. The left shows the *distribution* elements being added: I.) & II.) the interior edge of the Outer and Inner Coil's distribution, respectively. III.) the *distribution* that defines how fine the *sweep* is in Fig. 2.7. IV.) Results of the *Free Triangular* element on the cylinders faces. The resolution of the mesh is set from the *distributions* I.) and II.).

edges of the two coils and along the center line of the coils. This was accomplished by specifying a node *distribution* around the outside edges of both cylinders. With the distribution in place to set the scale, a *free triangular* element was added to one side of the cylinders' face (See Fig. 2.6). A meshing line was added along the center to set the quality in the interior region of the SCM, with the extent of the line extending slightly beyond the edges of the outer SCM coil. After completing the edge and face meshing, a *sweep* was performed over both cylinders (See Fig. 2.7). Lastly, a *conversion* was performed to complete the meshing of the SCM (See Fig. 2.8 This completes the meshing for the SCM. The only remaining part is to include the physics that makes this a coil. Because the SCM has a volume that is being filled, it is best to use the *coil* element, which takes a domain, an edge, and a current as input. The domains for the two coils were the inner and outer coils, respectively, and the

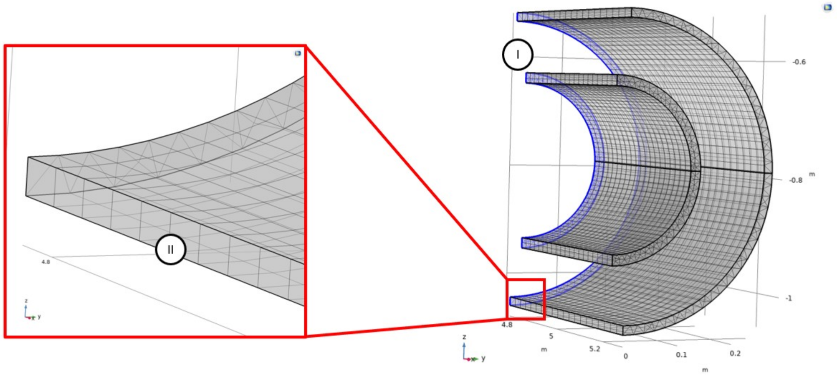


Figure 2.7: Shown is the *sweep* operation being preformed on the SCM. This takes copies of the *free triangular* surface mesh and distributes them at the intervals defined by the *distribution* along the length of the cylinder (see III in Fig. 2.6) to the target face I.) highlight blue. This does not complete the mesh (shown at II) because the face shown has not yet been broken into triangles.

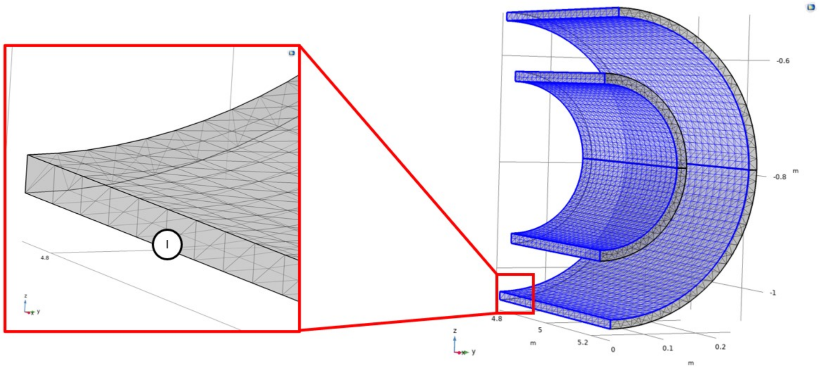


Figure 2.8: Shown is the *convert* element applied to the side not meshed in Fig. 2.7. I.) shows how the *sweep* applied to the surface is converted into triangles.

interior edge was selected for the defining edge for each[4]. Lastly, for the physics to be fully defined, the material must be included for the coils, which is just set from the COMSOL materials library as *copper*. With the SCM complete it is useful to look at the field profile to see if it matches what is expected. Figure 2.9 shows the magnetic field values along the center line. From the manufacture of the SCM it's known that the on-axis magnetic field strength reaches 5 T, which is seen in the model and is shown in Figure 2.9.

---

[4] See Appendix A for details on the dimensions and currents used to define both coils.

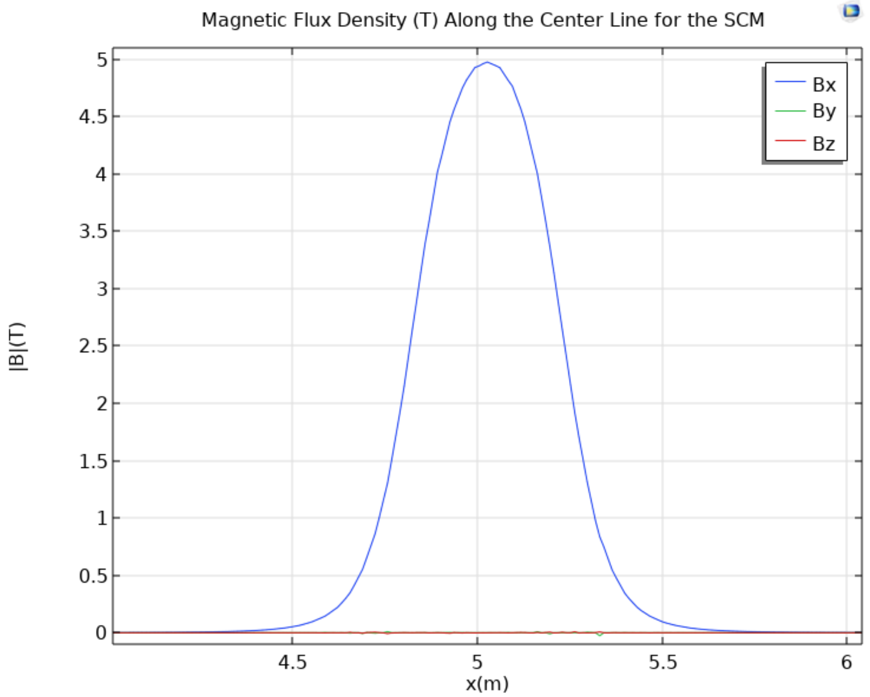


Figure 2.9: SCM magnetic field components along it's center line.

The $\boldsymbol{B}_0$-coil was constructed to complete the background field. Unlike the SCM, the $\boldsymbol{B}_0$-coil does not have a volume of current density. All of the wires for this coil are defined as edges in the model (See Fig. 2.10). In this regard, meshing the $\boldsymbol{B}_0$-coil was simpler. The complication arises in the difference of scale in detail for the $\boldsymbol{B}_0$-coil compared to the surrounding MSR. Due to this, particular consideration was needed for the mesh.

A *distribution* was placed along each edge to start meshing. Because the wires/edges are close to one another, they set the *size* of the *distribution*. At this point, the geometry informed the next steps in meshing. The $\boldsymbol{B}_0$-coil is near the MSR, so the region between them smoothly transitioned from the fine mesh on the edges to the coarser mesh of the MSR. This was accomplished by adding a box element to the geometry that all the edges sit on. This allowed for a *free triangular* mesh to be added to the surfaces between all the edges. This was then connected to the mesh of the MSR via a *free tetrahedral* mesh in the domain between them. This allowed for two *size*s to be

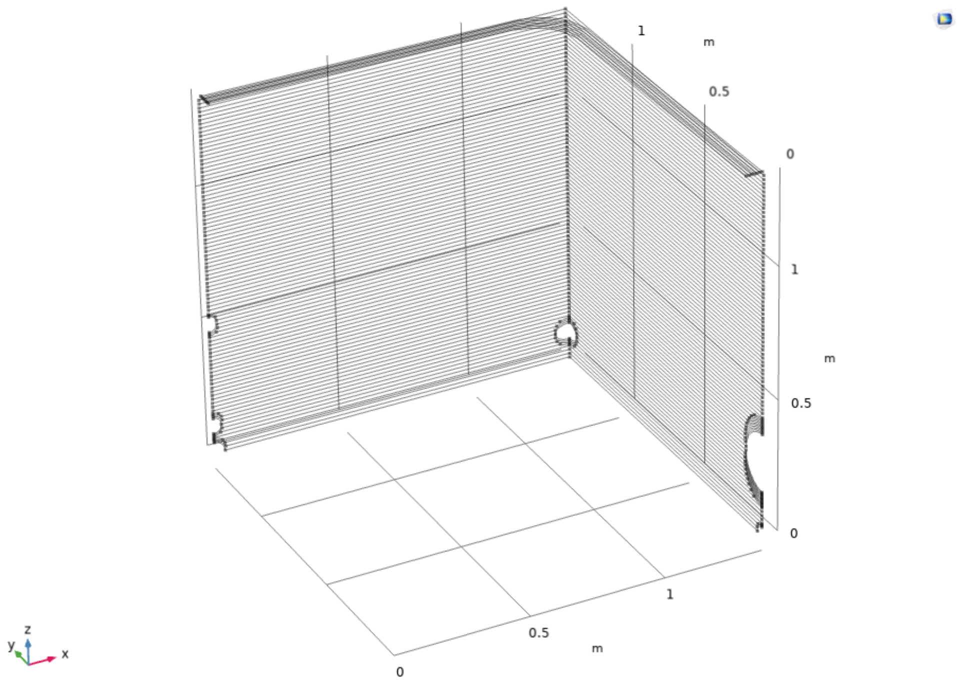


Figure 2.10: The $\boldsymbol{B}_0$ Coil was designed by members of the PSI collaboration and was implemented in the COMSOL model for an accurate background field calculation. Pictured on the right is the hole and winding around, allowing the STC to enter the MSR.

set; one inside the $\boldsymbol{B}_0$-coil and the other between the coil and the MSR. To complete the model, a *copper* material was set for all coil edges along with the current running through them.

2.1.3 Total Background Magnetic Field In The Target Region

The background magnetic field influences the design of the STC. The background field determined the base adiabaticity that should be improved upon. It also set the boundary conditions for the STC to adhere to. Figure 2.11 shows the magnetic field due to the $\boldsymbol{B}_0$ coil. The value of the $\boldsymbol{B}_0$ field at the start of the guide coil (innermost μ-metal layer) set the limit of the design field. The STC field could not modify the $\boldsymbol{B}_0$ field beyond this point. The only way to accomplish this was for the designed coil's field to go to zero as it approached the inside of the MSR. The SCM

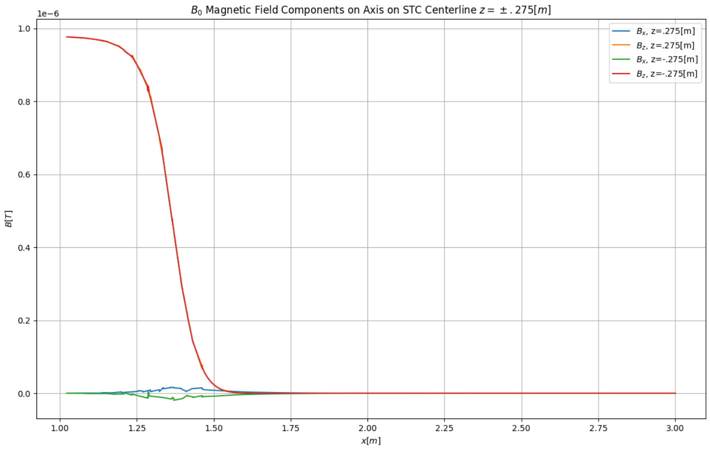


Figure 2.11: $\boldsymbol{B}_0$ coil's magnetic field components along STC center lines ($z = \pm.275$)

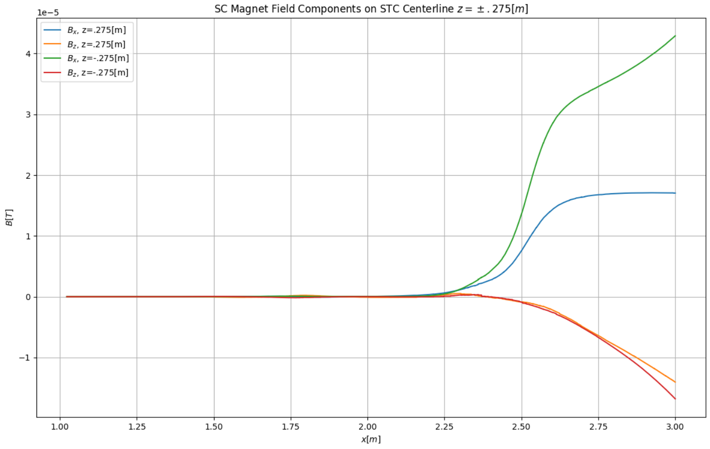


Figure 2.12: SCM magnetic field components along STC center lines ($z = \pm.275$)

contributed the most to the adiabaticity and, as such, was the field of interest for the designed fields. Figure 2.12 shows the SCM field as it enters the STC region. The only on-axis contribution to $\boldsymbol{B}_x$ comes from the SCM. The total background field was the sum of the SCM and $\boldsymbol{B}_0$ coils, shown in Figure 2.13 The total background field adiabaticity shows the state of the expected polarization before any guiding coils were added. Figure 2.14 shows the background adiabaticity, which is indicative of high uniformity, except for when the fields enter into the MSR, which is what is being corrected for.

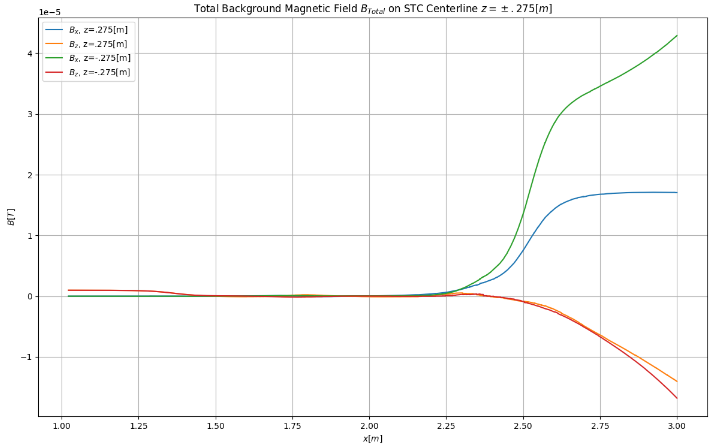


Figure 2.13: $\boldsymbol{B}_0$ and SCM coil's magnetic field components along STC center lines ($z = \pm.275$)

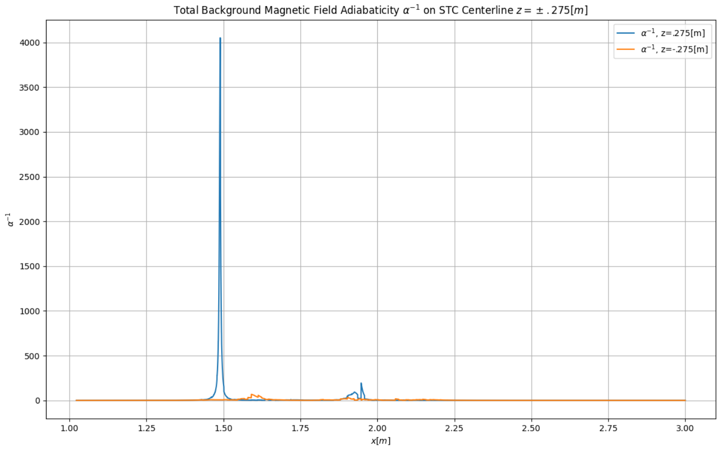


Figure 2.14: Adiabaticity of total background magnetic field along STC center lines ($z = \pm.275$). This plot was generated using a smoothing algorithm to avoid the noise in the first derivative caused by the mesh. To smooth the adiabaticity a running medium of the closest 3 point (on either side) was taken to smooth out the large spikes.

2.2 Finding the Target Field for the MSP

After determining $\boldsymbol{B}_{BG}$, we determined additional fields to maximize spin polarization during and after the spin rotation. This was simplified to make the problem solvable by optimizing the adiabaticity only on the central axis to design the field in the entire volume. This removed the need for a complete simulation after every field change. With this simulation, a slight change in the field is averaged over the many different random paths the particles take, making it harder to quantify the change needed to maximize adiabaticity. The Conjugate Gradient method was used to minimize the inverse adiabaticity. The magnetic flux on the surface of the UCN Guides was utilized in the design to avoid adding constraints to the gradient descent method and to ensure the fields were consistent with Maxwell's Equations with only surface currents outside the UCN guides. This came with two benefits: any field produced by the flux could be generated with surface current coils, and flux was a linear source term. Therefore, the functions describing the flux could be constructed into a basis. Under this construction, finding the maximum adiabaticity was reduced to determining coefficients of these basis functions.

Factoring in the boundary conditions, the field was rotated from $\hat{\boldsymbol{x}}$ (the horizontal direction of the STCs) to $\hat{\boldsymbol{z}}$ (vertical direction of the B_0 field, allowing for the field to be designed to contain no $\hat{\boldsymbol{y}}$ component. This also reduced the complexity of Eq. 1.8, making it faster to calculate. After simplifying, the equation still depended on the velocity's direction. For UCNs, which behave as a non-interacting ideal gas, the drift velocity was slow compared to the instantaneous speed in random directions. Thus we chose to first minimize the adiabaticity as a function of the velocity angle, and then to optimize this as a function of the field profile. This amounts to finding the best worst case. This means that the change only increases the adiabaticity for velocities in any other orientation to the field.

2.2.1 Simplifying Adiabaticity for Conjugate Gradient

Starting with the equation for adiabaticity defined in the introduction,

$$\alpha^{-1} = \frac{\left|\left(\boldsymbol{v} \cdot \overrightarrow{\nabla}\right) \boldsymbol{B} \times \boldsymbol{B}\right|}{\gamma B^3} \tag{2.2}$$

where $\boldsymbol{B} = \boldsymbol{B}_{total} = \boldsymbol{B}_{BG} + \boldsymbol{B}_{MSP}$, γ and $\boldsymbol{v}$ are the gyromagnetic ratio and velocity of the neutron respectively. As stated above, the designed magnetic field, $\boldsymbol{B}_{MSP}$ is chosen to have zero $\hat{\boldsymbol{y}}$ component. In Section 2.1.3 $\boldsymbol{B}_{BG}$ was shown in Fig. **??** to have no on-axis $\hat{\boldsymbol{y}}$ component outside artifacts of the *mesh*. Combining the two, $B_y = 0$. No currents are inside the STC, so Maxwell's Equations are Eq. 1.14 and Eq. 1.13. Expanding these when $B_y = 0$:

From Eq. 1.14

$$\partial_x B_x + \partial_z B_z = 0$$

$$\partial_x B_x = -\partial_z B_z. \tag{2.3}$$

From Eq. 1.13, taking μ to be linear in the constitutive relation $\boldsymbol{B} = \mu \boldsymbol{H}$,

$$\left(\partial_y B_x\right) \hat{\boldsymbol{e}}_z + \left(\partial_z B_x - \partial_x B_z\right) \hat{\boldsymbol{e}}_y + \left(\partial_y B_z\right) \hat{\boldsymbol{e}}_x = 0$$

Separating into components

$$\partial_y B_x = 0 \tag{2.4}$$

$$\partial_z B_x = \partial_x B_z \tag{2.5}$$

$$\partial_y B_z = 0 \tag{2.6}$$

Using these relations Eq. 2.2 can be reduced: First expanding α^{-1}

$$\begin{aligned}\gamma^{-1}B^{-3}\left(\boldsymbol{v}\cdot\overrightarrow{\nabla}\right)\boldsymbol{B}\times\boldsymbol{B} &= \gamma^{-1}B^{-3}(\epsilon_{lmn}v_i(\partial_i B_l)B_m\hat{\boldsymbol{e}}_n)\\ &= \gamma^{-1}B^{-3}\Big((v_i(\partial_i B_x)B_y - v_i(\partial_i B_y)B_x)\,\hat{\boldsymbol{e}}_z+\\ &+(v_i(\partial_i B_z)B_x - v_i(\partial_i B_x)B_z)\,\hat{\boldsymbol{e}}_y+\\ &+(v_i(\partial_i B_y)B_z - v_i(\partial_i B_z)B_y)\,\hat{\boldsymbol{e}}_x\Big)\end{aligned}$$

Using $B_y = 0$ only the $\hat{\boldsymbol{e}}_y$ component remains. Dropping the $\hat{\boldsymbol{e}}_y$ and writing out the sums

$$= \gamma^{-1}B^{-3}\Big(\,(v_x\partial_x B_z + v_y\partial_y B_z + v_z\partial_z B_z)\,B_x - (v_x\partial_x B_x + v_y\partial_y B_x + v_z\partial_z B_x)\,B_z\Big)$$

Plugging in Eq. 2.4 and Eq. 2.6 this reduces to

$$= \gamma^{-1}B^{-3}\Big(\,(v_x\partial_x B_z + v_z\partial_z B_z)\,B_x - (v_x\partial_x B_x + v_z\partial_z B_x)\,B_z\Big)$$

Using Eq. 2.5 and Eq. 2.3, the fields can be written in terms of ∂_x acting on some component of $\boldsymbol{B}$

$$= \gamma^{-1}B^{-3}\Big(\,(v_x\partial_x B_z - v_z\partial_x B_x)\,B_x - (v_x\partial_x B_x + v_z\partial_x B_z)\,B_z\Big)$$

Because all derivatives are expressed in terms of ∂_x, the partial can be dropped in favor of a prime. Using this notation change and grouping in terms of v_i

$$= \gamma^{-1}B^{-3}\Big(v_x(B'_zB_x - B'_xB_z) - v_z(B'_xB_x + B'_zB_z)\Big) \tag{2.7}$$

Making a switch to a polar representation of $\boldsymbol{B}$, $B_x = B\cos(\theta)$ and $B_z = B\sin(\theta)$, allows for the problem to be simplified further and helps conceptualization. Plugging this representation into Eq. 2.7.

$$= \gamma^{-1}B^{-3}\Bigg(v_x\Big((B'\sin\theta + B\theta'\cos\theta)B\cos\theta - (B'\cos\theta - B\theta'\sin\theta)B\sin\theta\Big)+$$

$$- v_z\Big((B'\cos\theta - B\theta'\sin\theta)B\cos\theta + (B'\sin\theta + B\theta'\cos\theta)B\sin\theta\Big)\Bigg)$$

$$= \gamma^{-1}B^{-3}\Bigg(v_x\Big(BB'\cos\theta\sin\theta - BB'\cos\theta\sin\theta + B^2\theta'\cos^2\theta + B^2\theta'\sin^2\theta\Big)+$$

$$- v_z\Big(BB'\cos^2\theta + BB'\sin^2\theta - B^2\theta'\cos\theta\sin\theta + B^2\theta'\cos\theta\sin\theta\Big)\Bigg)$$

This simplifies Eq. 2.2 to

$$\alpha^{-1} = \frac{v_x B\theta' - v_z B'}{\gamma B^2} \tag{2.8}$$

This is the most that can be done with just the constraints on the $\boldsymbol{B}$ field. However, if the velocity is in the worst possible orientation, this would remove reference to the velocity's direction in α^{-1}. Substituting for a polar representation of $\boldsymbol{v}$, $v_x = v\cos\xi$ and $v_z = v\sin\xi$, the worst point is where the angle ξ maximizes α^{-1}. This angle corresponds to when $\partial_\xi\alpha^{-1} = 0$.

$$\begin{aligned}\partial_\xi\alpha^{-1} &= \partial_\xi\left(\frac{vB\theta'\cos\xi - vB'\sin\xi}{\gamma B^2}\right) = 0\\ 0 &= \frac{-vB\theta'\sin\xi - vB'\cos\xi}{\gamma B^2}\\ \tan\xi &= -\frac{B'}{B\theta'}\end{aligned}$$

Taking this critical angle and plugging it back into Eq. 2.8 while taking advantage of the relations $\sin(\tan^{-1} x) = x/\sqrt{x^2+1}$ and $\cos(\tan^{-1} x) = 1/\sqrt{x^2+1}$

$$\begin{aligned}
\alpha^{-1} &= \frac{v}{\gamma B^2}\left(\frac{B\theta'}{\sqrt{\left(\frac{B'}{B\theta'}\right)^2+1}} - \frac{B'\left(-\frac{B'}{B\theta'}\right)}{\sqrt{\left(\frac{B'}{B\theta'}\right)^2+1}}\right) \\
&= \frac{v}{\gamma B^2}\left(\frac{(B\theta')^2}{B\theta'\sqrt{\left(\frac{B'}{B\theta'}\right)^2+1}} + \frac{(B')^2}{B\theta'\sqrt{\left(\frac{B'}{B\theta'}\right)^2+1}}\right) \\
&= \frac{v}{\gamma B^2}\left(\frac{(B\theta')^2+(B')^2}{\sqrt{(B\theta')^2+(B')^2}}\right)
\end{aligned}$$

This can be simplified further into

$$\alpha^{-1} = \frac{v\sqrt{(B\theta')^2+(B')^2}}{\gamma B^2} = \frac{v\sqrt{(B'_x)^2+(B'_z)^2}}{\gamma B^2} \tag{2.9}$$

This relation says that the adiabaticity on-axis can be determined by the speed of the neutrons (v), the neutron's gyromagnetic ratio (γ), the magnitude of the magnetic field (B), the rate at which the field strength is changing on-axis ($B' = \partial_x B$), and the rate that the field rotates from $\hat{\boldsymbol{x}}$ to $\hat{\boldsymbol{z}}$ on-axis ($\theta' = \partial_x \theta$). This is significant because it does not reference the direction of the neutrons, thus removing the need for simulations in this design stage. There is room to simplify this further in the numerator if one term dominates the other inside the square root. However, this depends on the shape and strength of the field.

2.2.2 Designing Magnetic Flux Basis Functions

The criteria of the basis functions are as follows. First, the fields produced by the basis functions should smooth the transition from the SCM field to the $\boldsymbol{B}_0$ Coil field. This is best accomplished by considering two sets of basis functions, one set to guide the neutrons along the UCN Guide and the second to help smooth the field's transition outside the MSR (see Fig. 2.15) Due to the construction of the MSR and surrounding experimental apparatus, the exterior coil is constrained slightly off the

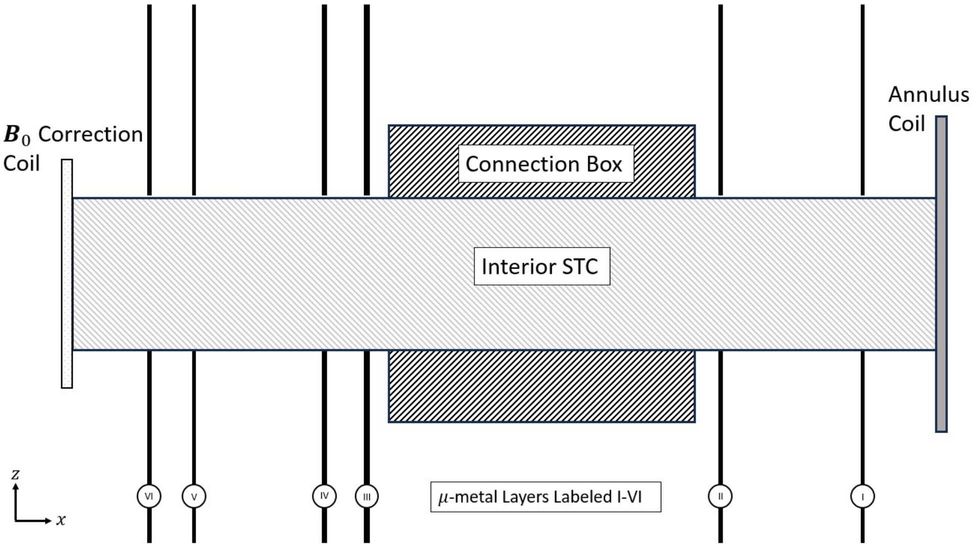


Figure 2.15: Overview of the coils and regions names in and around the MSR.

surface of the MSR[5], confining the coil to the shape of an annulus. Second, each basis function should satisfy Laplace's Equation ($\nabla^2 U = 0$) in the spin transport region. Any linear combination of functions will also satisfy Laplace's Equation and can, therefore, be considered a basis. Third, the fields resulting from the basis functions should have no $\hat{\boldsymbol{y}}$ component, which would only decrease the adiabaticity. This limits the functions considered to have no dependence on y; otherwise, they will have a $\hat{\boldsymbol{y}}$ component. Finally, the field constructed field should match the $\boldsymbol{B}_0$ field inside the MSR. In Figure 2.11, $\boldsymbol{B}$ in the interior is stable around 1 μT and only begins to deviate as one approaches the innermost μ metal layer of the MSR at $x = 1.462$ m. This sets the constraint at the edge of the STC that the total field should equal the interior field of $1\mu T$ and have zero slope, thus extending the constant range of the $\boldsymbol{B}_0$ field. The transition from the B_0 to the STC is dealt with in the coil design phase, using an additional coil, called the $\boldsymbol{B}_0$ Correction Coil, to account for wires of the B_0 Coil being pushed above and below out of the way of the hole in the MSR for the UCN guides. In the field design phase, it was sufficient to assume an ideal transition from B_0 to the STC.

---

[5] See Appendix A for details on coil placements.

The basis functions were calculated from prescribed patterns of magnetic flux on the inner STC and annulus surfaces (See Fig. 2.15). The flux was used instead of the scalar potential because the open end of the STC broke the hermiticity of the boundary surface, preventing calculation of windings for arbitrary fields inside the STC. Constraining the flux to the surface of the STC coils, guaranteed the ability to construct resuling field. Thus the magnetic flux defining the basis functions was confined to the surface of a cylinder, labeled Interior STC in Figure 2.15. This did not include all details of the surface-current STCs, for example a cylindrical envelope around the connection box between layers 2 and 3 that the coils must be wound around (see Fig. 2.15), but it was sufficient to guarantee constructability.

The basis functions used were:

- STC Surface Flux:

  $$\Phi_k = \cos(\psi)\left(\cosh\left(k(x - \ell_{inner})\right) - 1\right), \tag{2.10}$$

  where k indexes the basis function and length-scale of attenuation, ψ is the angle in the yz-plane measured from the center line of the STC, ($z = \pm 275$ mm): $\tan\psi = (z \mp 275\text{ mm})/y$, and $x = \ell_{inner} = 1462$ mm is at the innermost μ-metal layer (layer six). These functions are chosen to die off exponentially to the B_0 field inside the MSR.

- Annulus Coil Surface Flux:

  $$\Phi_{n,m} = (\rho - r_{STC})^n \cos(m\psi) \tag{2.11}$$

  Where both n, m index the basis functions, ρ is the radius in the yz-plane measured from the center of the STC $\rho = \sqrt{y^2 + (z \mp 275\text{ mm})^2}$, $r_{STC} = 110$ mm is the radius of the STC, and ψ is the same as above. These basis

> functions were chosen to extend the vertical guide field outside the MSR, to match up with the fringe of the SCM and rotate the field from longitudinal to vertical farther outside the MSR, to allow for a longer adiabatic taper from $\sim 30\ \mu$T down to $B_0 = 1\ \mu$T.

Linear combinations of these basis functions informed how the coils should be wound. To optimize the adiabaticity, we calculated the magnetic field of each of these basis functions using the COMSOL model described above. By linearity of Maxwell's equations and materials inside the STC and outside the MSR, scaling a flux boundary condition produced field scaled by the same value. The same parameters that minimized the inverse adiabaticity were applied to the corresponding flux in the basis functions to determine the total flux on the surface. From the flux boundary conditions shown in equations Eq. 2.10 Eq. 2.11, the corresponding magnetic fields were calculated with corresponding indices, $\boldsymbol{B}_{n,m}$ for the Annulus Coil and $\boldsymbol{B}_k$ for the STC.

2.2.3 Modeling Basis Functions In COMSOL

To model the basis functions, one quarter symmetry was used: both y-symmetry mentioned above, and z-symmetry, which is only broken by placement of the SCM below the center of the MSR. Additionally, the μ-metal layers were simplified to extend into the *infinite layer domain*[6] Since the field was only simulated inside the STC and outside the MSR, only the outer μ-metal layer contributed to the fields, and the field was very insensitive to both the thickness and height of this layer. Figure 2.16 shows the geometry of the basis functions used to model the functions of Equations 2.10 and 2.11 on the respective surfaces (shown in Figure 2.17).

A separate field map file was created for each basis function. All basis functions

---

[6] This was originally the beginning of the project, and the improved models were yet available. This is why, in part, this model is more simplistic than previous sections.

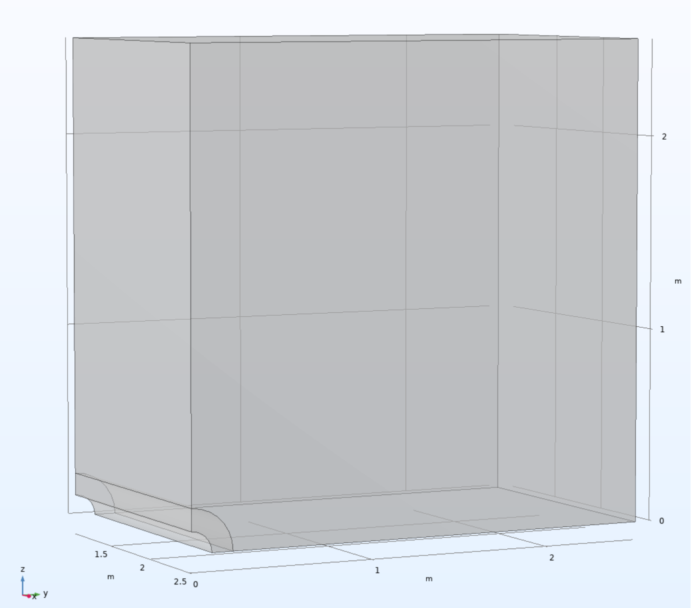


Figure 2.16: The geometry of the basis functions model in COMSOL.

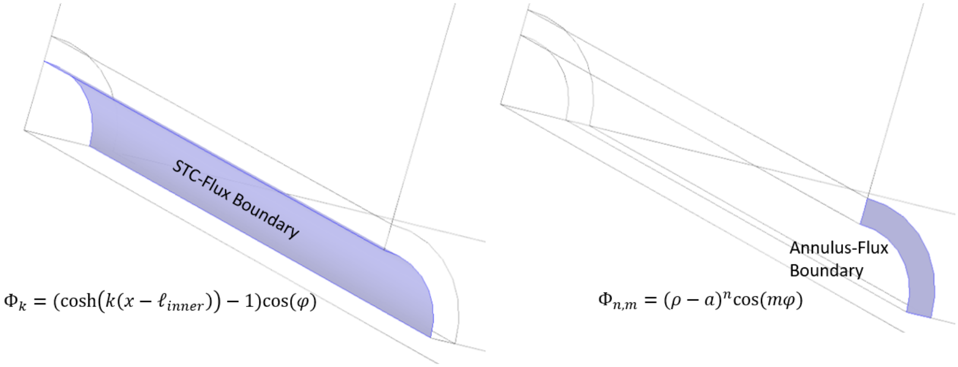

STC-Flux Boundary
$\Phi_k = (\cosh(k(x-\ell_{inner}))-1)\cos(\varphi)$
Annulus-Flux Boundary
$\Phi_{n,m} = (\rho - a)^n \cos(m\varphi)$

Figure 2.17: On the left are the surfaces to which the STC flux is applied. The right shows the surfaces of the Annulus flux.

were modeled simultaneously using the parametric sweep feature of COMSOL. This cut down on modeling time and effort to set up the model. This was accomplished simply by including two parameters called *Annulus Switch* and *STC Switch*, which had values $\{1, 0\}$. The two switch values were set to 1 when the corresponding flux was being studied and 0 when it was not. Defining the flux as the switch variable multiplied by the corresponding flux allowed the basis to be selected without extra effort. This made parametric sweeps over the parameters k and m, n for the respective coils a viable option. To finish the boundary conditions, all μ-metal were fixed to ground potential as well as the xy-plane, and mirror symmetry (zero flux) was set on the $y = 0$ plane.

The mesh for this model focused on the details on the STC surface and its center line. These correspond to the areas where the flux is defined and the axis of interest for the export. This is accomplished by *distribution* elements set along all edges of the STC to set the scale of the *free triangular* element applied to the surface. Then, an additional *distribution* element is applied to the center line; this ensured the axis being exported along had high resolution. Finally, the remaining volume was meshed with *free tetrahedral* elements.

A subset of the basis functions was used to narrow the optimization parameter space. Functions were selected based on the expected field profile. The on-axis $\boldsymbol{B}$ fields are shown for all considered basis functions in Figures 2.18 and 2.19. For the STC basis fields, only integers $k = 1, 2, ..., 10$ were considered, with the addition of $k = .1$. These k values were selected for simplicity and to allow maximal tuning of the magnetic field near the entrance of the MSR. Only the $m = 0$ terms contributed to the on-axis field for the annulus coil. Additionally, higher order n terms had scaled fields very close to the $n = 1$ case, so only the $n = 1, m = 0$ basis function was used on the annulus.

2.2.4 Conjugate Gradient and Optimal Coefficients

The final step of field design was to optimize the adiabaticity, Eq 2.9, along the central axis, as a function of the magnetic field; all other parameters are constants. From the basis fields of Sec. **??**, the total field is the background field plus the linear combination

$$\boldsymbol{B}^{total} = \sum_k a_k \boldsymbol{B}^k + \sum_{n,m} a_{n,m} \boldsymbol{B}^{n,m} + \boldsymbol{B}^{BG}, \tag{2.12}$$

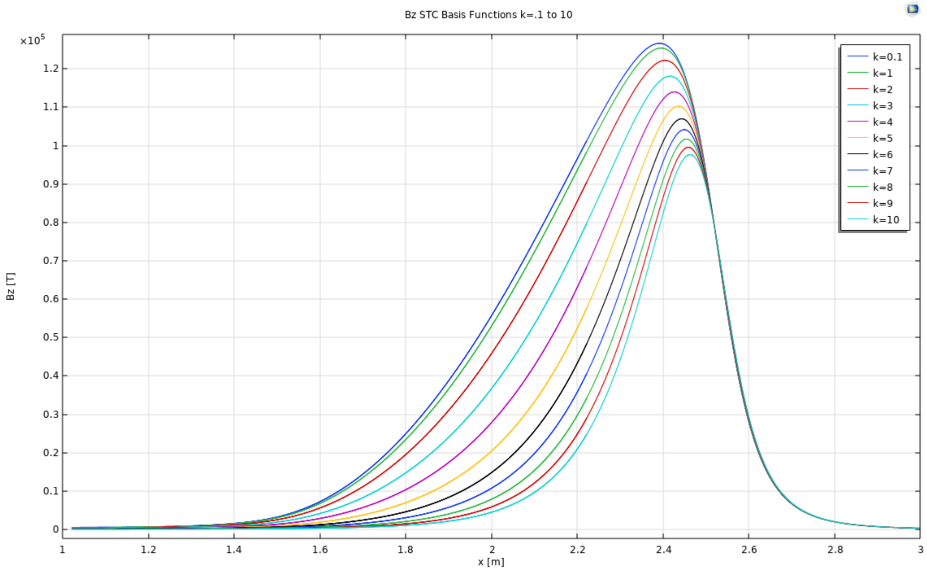


Figure 2.18: Basis function for the STC. Values of k were chosen to allow for the most control in tuning to occur nearest the entrance of the MSR.

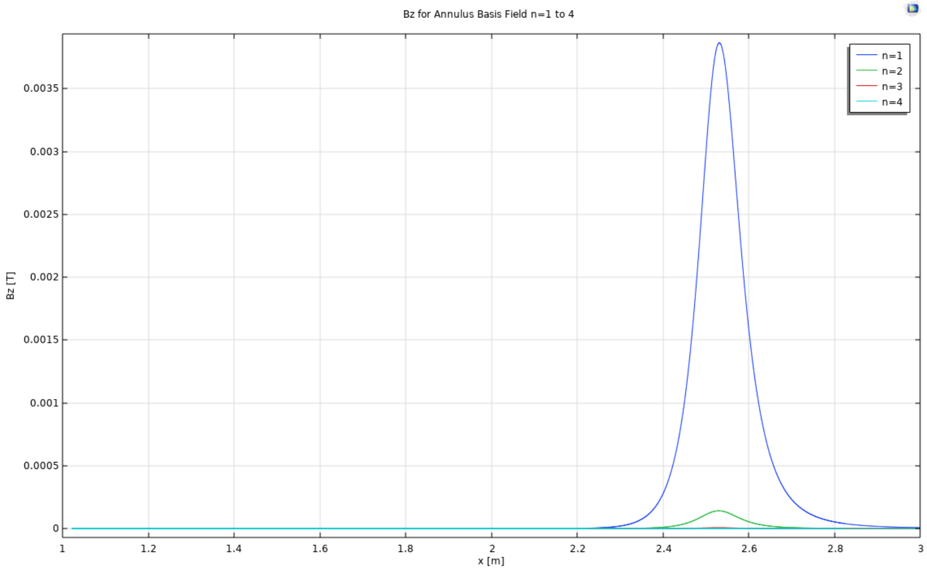


Figure 2.19: Basis functions for the Annulus coil. All terms above $m = 1$ contributed zero on-axis to the magnetic field, and all $n > 1$ scaled the $n = 1$ case. This leaves $m = 1, n = 1$ as the only Annulus basis function.

where $\boldsymbol{B}^k$ and $\boldsymbol{B}^{n,m}$ are the fields generated from the STC flux Eq. 2.10 and Annulus flux Eq. 2.11 respectively, and the $\boldsymbol{B}^{BG}(=\boldsymbol{B}_{BG})$ is the total background field defined in Eq. 2.1. This reduced the problem to finding the set of coefficients a_k and $a_{n,m}$ that minimize α^{-1}. This was accomplished using the Conjugate Gradient Method [21, Sec. 10.6].

The basic algorithm for the Conjugate Gradient Method is as follows:

1. Initialization:

$$\boldsymbol{g}_0 = \boldsymbol{h}_0 \tag{2.13}$$

2. Calculate Hessian:

$$H_{ij}(\boldsymbol{g}_k) = \partial_i \partial_j f(\boldsymbol{g})|_{\boldsymbol{g}_k} \tag{2.14}$$

3. Calculate Scalar λ_k:

$$\lambda_k = \frac{\boldsymbol{g}_k \cdot \boldsymbol{g}_k}{\boldsymbol{h}_k H(\boldsymbol{g}_k)\boldsymbol{h}_k} \tag{2.15}$$

4. Update to $\boldsymbol{g}_{k+1}$:

$$\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \lambda_k H(\boldsymbol{g}_k)\boldsymbol{h}_k \tag{2.16}$$

5. Calculate Scalar γ_k:

$$\gamma_k = \frac{\boldsymbol{g}_{k+1} \cdot \boldsymbol{g}_{k+1}}{\boldsymbol{g}_k \cdot \boldsymbol{g}_k} \tag{2.17}$$

6. Update to $\boldsymbol{h}_{k+1}$:

$$\boldsymbol{h}_{k+1} = \boldsymbol{g}_{k+1} + \gamma_k \boldsymbol{h}_k \tag{2.18}$$

7. Repeat:

   Repeat 2.) - 6.) until convergence or number of specified iterations is reached.

Because an explicit function exists for the adiabaticity, Equation 2.9, the Hessian can be calculated directly. The parameters being minimized are the magnetic field coefficients, the a_k and $a_{n,m}$ terms in Equation 2.12. If the terms are relabeled a_i with the single index i over the set $\{a_k, a_{n,m}\}$, then

$$H_{ij}(\boldsymbol{a}) = \left.\frac{\partial^2 \alpha^{-1}}{\partial_{a_i}\partial_{a_j}}\right|_{\boldsymbol{a}} \tag{2.19}$$

To find the Hessian in Equation 2.19, we calculate the derivatives[7].

$$\begin{aligned}
\partial_i \alpha^{-1} &= \frac{v}{\gamma}\partial_i\left(\frac{\sqrt{(B_x^{'})^2 + (B_z^{'})^2}}{B^2}\right) \\
&= \frac{v}{\gamma}\left(\frac{B_x^{'}\partial_i(B_x^{'}) + B_z^{'}\partial_i(B_z^{'})}{B^2\sqrt{(B_x^{'})^2 + (B_z^{'})^2}} - 2\Big(B_x\partial_i(B_x) + B_z\partial_i(B_z)\Big)\frac{\sqrt{(B_x^{'})^2 + (B_z^{'})^2}}{B^3}\right)
\end{aligned}$$

The term $\partial_i \boldsymbol{B}$ or $\partial_i \boldsymbol{B}^{'}$ selects out the corresponding basis field $\boldsymbol{B}^i$ or $\boldsymbol{B}^{'i}$ respectively, where $\boldsymbol{B}^i$ are indexed in the same manner as a_i. Another simplification was made by introducing the form $B' = \sqrt{(B_x^{'})^2 + (B_z^{'})^2}$ and so $\frac{1}{2}\partial_i(B')^2 = B_x^{'}B_x^{'i} + B_z^{'}B_z^{'i}$

$$\partial_i \alpha^{-1} = \frac{v}{\gamma}\left(\frac{\partial_i B' - 2B'^2\partial_i B}{B^2 B'}\right)$$

To find the second derivative, note $\partial_i\partial_i \boldsymbol{B} = 0$.

$$\begin{aligned}
\partial_j\partial_i \alpha^{-1} &= \frac{v}{\gamma}\partial_j\left(\frac{\partial_i B' - 2B'^2\partial_i B}{B^2 B'}\right) \\
&= \frac{v}{\gamma}\left(\frac{\partial_j\partial_i B' - 4B'\partial_j B'\partial_i B - 2B'^2\partial_j\partial_i B + (2BB'\partial_j B + B^2\partial_j B')(\partial_i B' - 2B'^2\partial_i B)}{B^2 B'}\right) \\
&= \frac{v}{\gamma}\left(\frac{\partial_j\partial_i B' - 4B'\partial_j B'\partial_i B - 2B'^2\partial_j\partial_i B +}{B^2 B'}\right. \\
&\quad \left. + \frac{2BB'\partial_j B\partial_i B' + B^2\partial_j B'\partial_i B' - 4BB'^3\partial_i B\partial_j B - 2B^2 B'^2\partial_i B\partial_j B'}{B^2 B'}\right)
\end{aligned}$$

---

[7] $\partial_i = \partial/\partial a_i$

Parameter	Value
$a_{k=1}$	3.7×10^{-4}
$a_{k=2}$	-9.761×10^{-5}
$a_{k=5}$	1.383×10^{-6}
$a_{n,m=1,1}$	-2.59×10^{-3}

Table 2.2: Target field minimization parameters for Equation 2.12

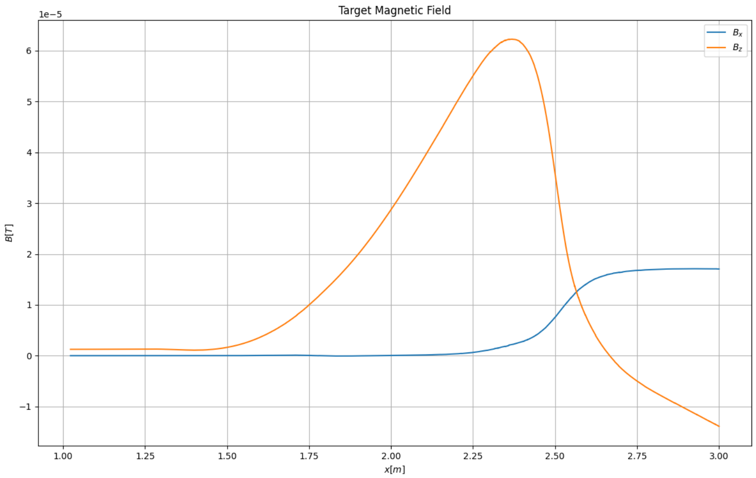


Figure 2.20: Total target magnetic field including the background field. Outer *mu* metal layer at $x = 2.3585$m and inner layer at $x = 1.468$m.

None of the terms cancel from here, so each was programmed in for the Hessian. Running the algorithm described above, the resulting target field parameters are shown in Table 2.2. These parameters served as a starting point that was further optimized with a full simulation of UCN trajectories and spin tracking along the neutron flight path in the STC by Geza Zsigmond at PSI. The fully optimized parameters are shown in Table 3.1. Figure 2.20 shows the resulting field from the first minimization. The target field has an adiabaticity plot shown in Figure 2.21. This is a dramatic improvement from the background adiabaticity shown in Figure 2.14.

Copyright© David C. Bowles, 2024.

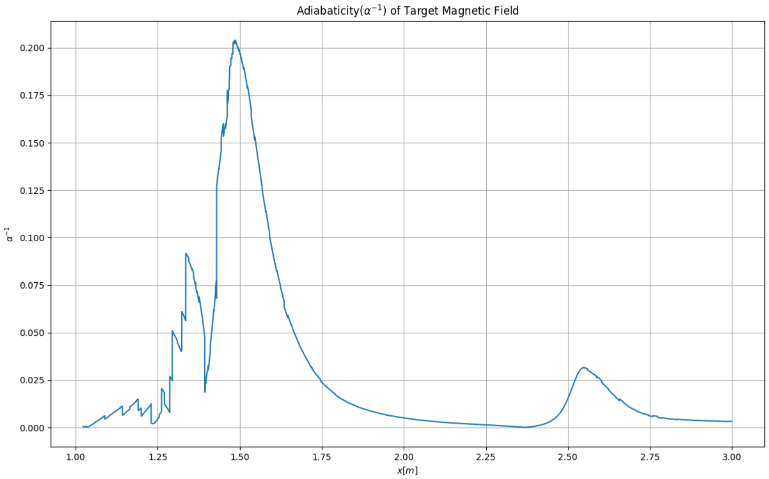


Figure 2.21: Target field adiabaticity. Adiabaticity was smoothed similarly to Figure 2.14.

Chapter 3 Coil Design and Modeling

With the target magnetic field solved in Chapter 2, the next task was to solve for the wire geometries. This was done in three steps: First, the isocontours in the scalar potential were found on the coil surfaces. Second, the discrete wires was modeled in COMSOL. Third, the design was iterated if necessary to account for discretization of the potential into wires. The isocontours were found by modeling the total flux (found in Chapter 2 Sections 2.2.2 and 2.2.4), and solving for the scalar potential. From the scalar potential, isocontours were created by specifying the desired current in the wires. The isocontour geometry was then exported and ordered into paths. To compare to the calculated field, the wire paths were imported back into COMSOL as physical geometry. The modeled coils field was then compared to the target field to verify the agreement.

3.1 Finding Isocontours

The wire geometries were determined by the isocontours of the magnetic scalar potential on the surface of the coil (See Chapter 1 Section 1.3). To find the scalar potential, the target magnetic field, found in Chapter 2 Section **??**, was used to create a new COMSOL model. The model found the combined MSP from the interior and exterior on the coil's surface. To find the total MSP the model was broken up into two separate parts: the target region, which lies inside the coil, and the return region, which encompasses all regions outside the coil (See Figure 3.1). COMSOL requires this separation because flux boundary conditions can only be specified on exterior surfaces. This was accomplished using the *Assembly* geometry finalization in COMSOL. The *Assembly* allowed for two models to be built without having a union of the geometry elements' exterior surface. Only one-quarter symmetry was used for

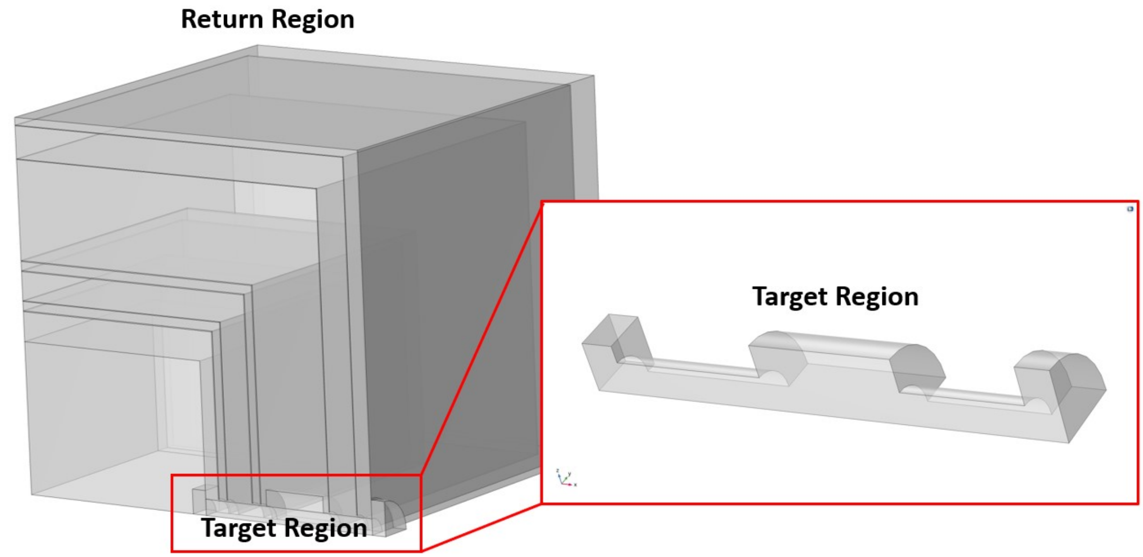


Figure 3.1: The two regions of the model. Left: the whole model with both target and return regions visible. Right: just the target region.

this model because it simplified the design of the coil and reduced the manufacturing requirements by making the top and bottom coil haves symmetric. It prevented flux of the B_0 Coil from taking a long detour out the top guide, and back in through the bottom guide, which was exasperated by the flux isolation boundary conditions. Most importantly, the model is more sensitive to the placement of the layer positions along x and is not sensitive to the roof placement in z.

The Boundary conditions for this model were in two categories: specifying surface scalar discontinuities/flux Densities and symmetries/boundary matching. The parameters for MSP discontinuity and flux boundary conditions are listed in Table 3.1. The discontinuity boundary condition are:

- $\boldsymbol{B}_0$ coil MSP discontinuity

$$V_{m,d} = \frac{B_0}{\mu_0}(z - z_{STC}) + V_m \tag{3.1}$$

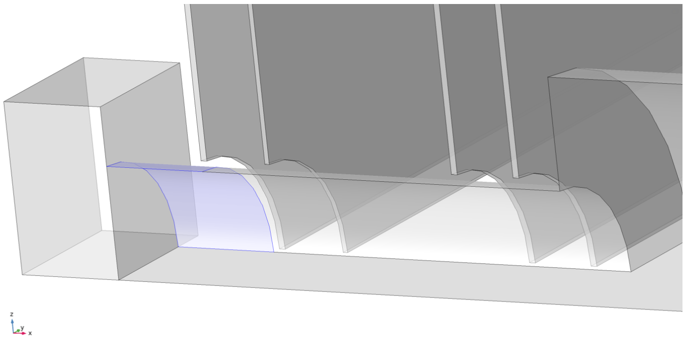


Figure 3.2: Flux boundary condition applied to the surface highlighted. This boundary is separate from the other Flux Boundaries in the STC because this surface was not modeled in the basis function.

- Flux boundary condition applied to the surface shown in Figure 3.2

$$B_z = B_0 \tag{3.2}$$

- The Inner, CB Inner, CB, CB Outer, Middle and Outer Coil Surfaces (See Figure 3.3 for the names of surfaces) flux boundary condition

$$\begin{aligned} B_x &= -\sum_i A_i \sinh(k_i x - \phi_i) \sin\Big(k_i(z - z_{STC})\Big) \\ B_z &= B_0 - \sum_i A_i \cosh(k_i x - \phi_i) \cos\Big(k_i(z - z_{STC})\Big) + C_i \end{aligned} \tag{3.3}$$

- Annulus flux boundary conditions

$$B_x = D \frac{(z - z_{STC})\Big(\rho_A - \sqrt{y^2 + (z - z_{STC})^2}\Big)}{\sqrt{y^2 + (z - z_{STC})^2}} \tag{3.4}$$

In the symmetries/boundary matching category, there are two additional boundary conditions: *Zero Magnetic Scalar Potential*, and *Continuity*. The *Zero Magnetic Scalar Potential* was used to set the ground as being the plane defined by

Variable Name	Value	Description
V_m	-	*mfnc* potential function
B_0	$1\mu T$	Value of the $\boldsymbol{B}_0$ coils field strength
A_1	$-8.5961952 \times 10^{-5}[T]$	Fitted flux coefficient for k=1
A_2	$1.03594365 \times 10^{-5}[T]$	Fitted flux coefficient for k=2
A_5	$-1.425978108 \times 10^{-8}[T]$	Fitted flux coefficient for k=5
k_1	$.7146[m^{-1}]$	Fitted parameter k=1
k_2	$1.65106[m^{-1}]$	Fitted parameter k=2
k_5	$5.263562[m^{-1}]$	Fitted parameter k=5
ϕ_1	1.03639	Fitted parameter k=1
ϕ_2	2.39755	Fitted parameter k=2
ϕ_5	5.902724	Fitted parameter k=5
C_1	$8.5464672 \times 10^{-5}[T]$	Fitted parameter k=1
C_2	$-1.003543935 \times 10^{-5}[T]$	Fitted parameter k=2
C_5	$1.759375152 \times 10^{-8}[T]$	Fitted parameter k=5
ρ_A	$225[mm]$	Radius of Annulus
D	$-9.065 \times 10^{-4}[T]$	Annulus Flux fitted coefficient
z_{STC}	275[mm]	Position of center line of the STC

Table 3.1: Equations 3.1, 3.2, 3.3, and 3.4 parameters

$z = z_{STC} = .275$ m, which is the symmetry of perpendicular fields at the boundary. The other external boundaries employ the default ”No flux” boundary condition, which implies the symmetry of parallel fields to the surface. The *Continuity* boundary condition is used to connect the fields of the two separate assemblies in the model.

Because there are two geometries in the same model, the mesh requires extra attention. The mesh for the guide coils follows the ordered scheme: *distribution* elements on all edges, followed by *free triangular* elements on the boundaries with a *free tetrahedral* element in the volume. This process was mostly repeated for the two geometries. For the model that does not include the interior of the Guide coils, the exception was that the *free tetrahedral* element was not required for the interior region. Because the two models share boundaries, careful attention to which boundary is selected was necessary for the model to be meshed properly.

3.2 Extracting the Coil Geometries

Once the MSP has been modeled on the coil surfaces, the next step was to find the contours that define the wire geometry. From Chapter 1 Section 1.3, the winding geometry of the wires carrying current I is specified by the isocontours separated by potential difference $\Delta V = I$ on the boundary.

There are many factors to consider when choosing the current. What surfaces will be at the same current? The construction informs the surface selection of the coil. Because the STC is being built around a Connection Box (CB), the faces on the cylinder of the CB must be wound separately for access to the CB. The density of the winding informs the choice of current. If a sparse winding density occurs, the resulting field will deviate from the target field due to the contributions of fringes around the wires. If the wire density is high, several issues may occur. A density that is too high can cause the wires of finite thicknesses to overlap along the wire paths, making the paths unphysical. Higher winding density requires more wire, increasing cost, and is difficult to manufacture. The power supply quantity and capabilities constrain the current as well. The current cannot exceed what the supply can produce, and there cannot be more required currents than sources available. Temperature requirements inform the choice of current and the gauge of wire. A large current in a wire with a small cross-sectional area produces excess heat, which affects the system's dynamics. Because the fields in the STC various by over a factor of 30, it is necessary to wind it in segments of decreasing current going into the MSR to keep the winding density approximately constant.

Considering the constraints, the STC was broken into four separate currents on five coils (See Figure 3.3 for the naming scheme of coil components).

Because the STC contains a $\boldsymbol{B}_0$ correction coil component, we decided to match the current in this section of the coil to the that of the $\boldsymbol{B}_0$ coil at $11.75mA$, to have the same wire spacing at the interface between the two coils. (See Figure 3.4). The CB

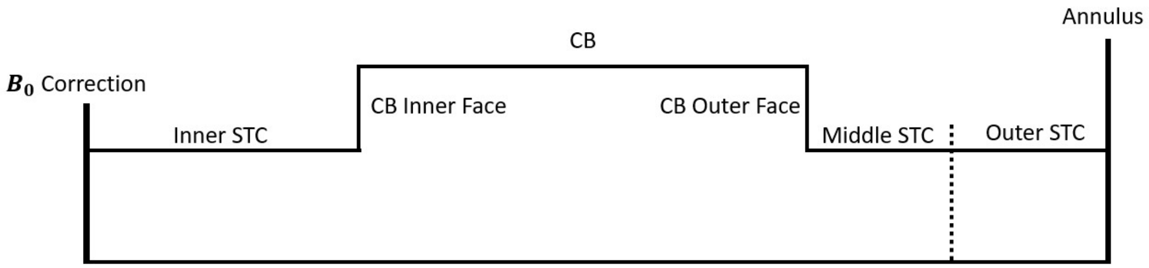


Figure 3.3: Names of the Guide Coil components. $\boldsymbol{B}_0$ Correction refers to the correction coil that compensates the $\boldsymbol{B}_0$ coils fringes. In the context of coils, the CB refers to the coil that resides on the lateral area of the cylinder surrounding the CB. The inner and outer faces of the CB refer to the faces of the cylinder closest/furthest to the center of the MSR, respectively. The middle and outer STC coils are divided by layer 1 of the μ-metal with a slight offset pushing further into the MSR.

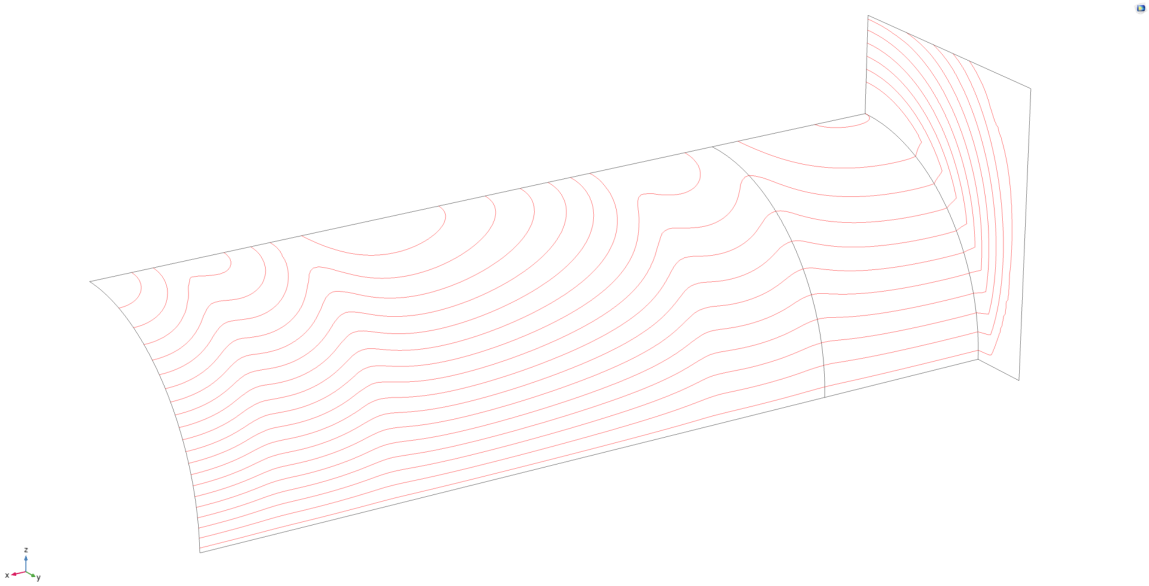


Figure 3.4: Wire geometries for the inner coils and $\boldsymbol{B}_0$ correction coil. These constitute a single coil at 11.75mA.

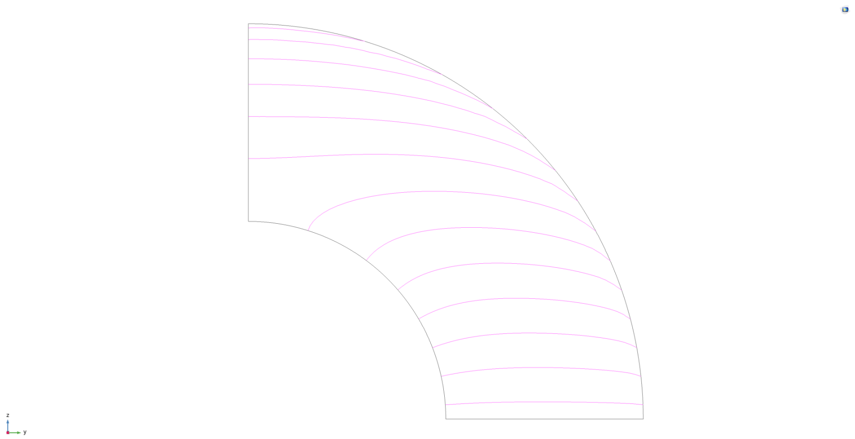


Figure 3.5: Wire geometries for the CB's inner face coil. These constitute a single coil at $35mA$.

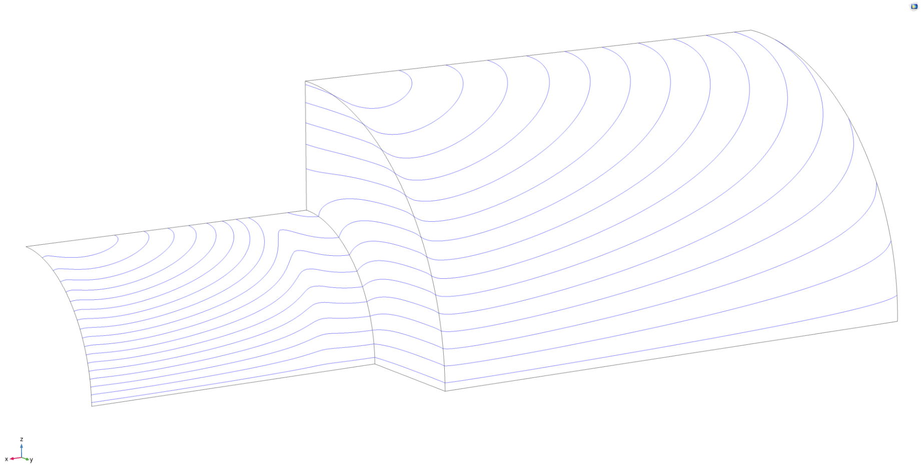


Figure 3.6: Wire geometries for the CB, CB's outer surface, and middle STC coils. These are three separate coils, though they are all at $100mA$.

had special considerations for its construction because of the presence of a supporting bar for the CB, which separated windings on the inner and outer surfaces of the CB. Keeping the current at $11.75mA$ resulted in a windings that were too dense on the CB's inner surface, so the current was increased to $35mA$ (See Figure 3.5). The CB and the CB outer surface are also wound separately because of support bar for the CB, which also broke the hermeticity of the surface. $35mA$ produced a high winding density for these surfaces, so the current was increased to $100mA$(See Figure 3.6). The outer STC and annulus coils are one coil and are at a current of $300mA$ (See 3.7).

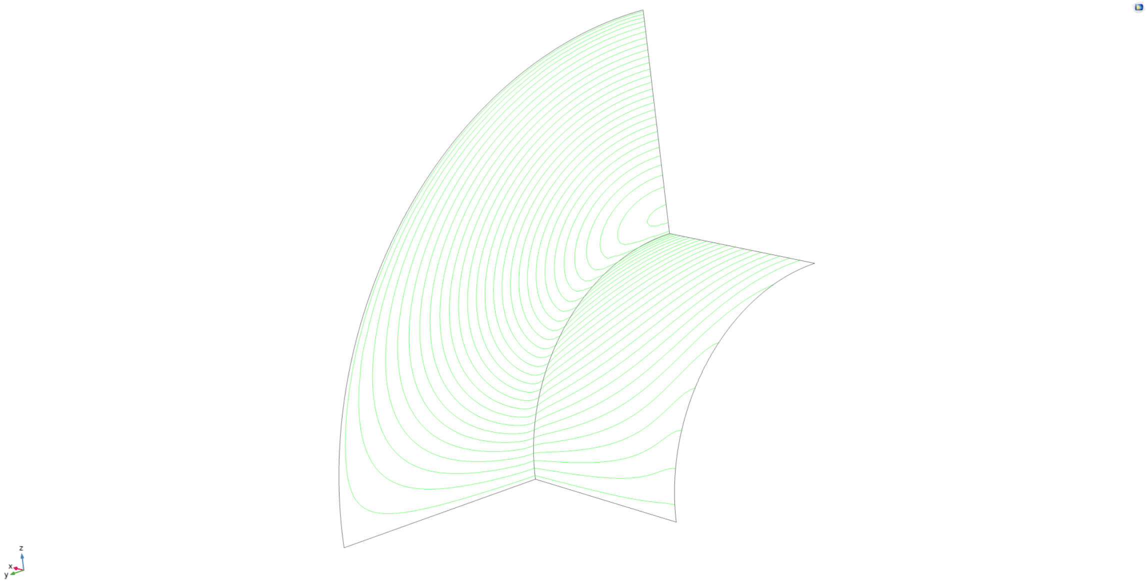


Figure 3.7: Wire geometries for the annulus and outer STC coils. These constitute a single coil at $300mA$.

It is expected that higher currents are required for portions of the guide coil close to the exterior of the MSR because the guide coil's fields were designed to increase as one left the MSR to connect the fields of the SCM to the interior $\boldsymbol{B}_0$ field. The current was chosen intentionally to increase by a factor of three between each section to avoid abrupt jumps in current. The wire gauge was chosen to match the current, varying between AWG 18 on the outside, to AWG 24 on the inside.

The separate coils culminate into the total guide coil shown in Figure 3.8. The paths were exported for further refinement with COMSOL's export functionality for plot groups, which saves point clouds that describe the contours on the targeted surfaces. Each coil: $\boldsymbol{B}_0$ Correction, Inner STC, CB Inner Face, CB, CB Outer Face, Middle STC, Outer STC, and Annulus coils was exported separately to simplify the refinement process. The export format for each coil consisted of an unordered point cloud associated with an "IsoLevel"[1] in the form of a tab-separated columnated list.

---

[1] IsoLevel is the name assigned to the column. These values are the equipotentials.

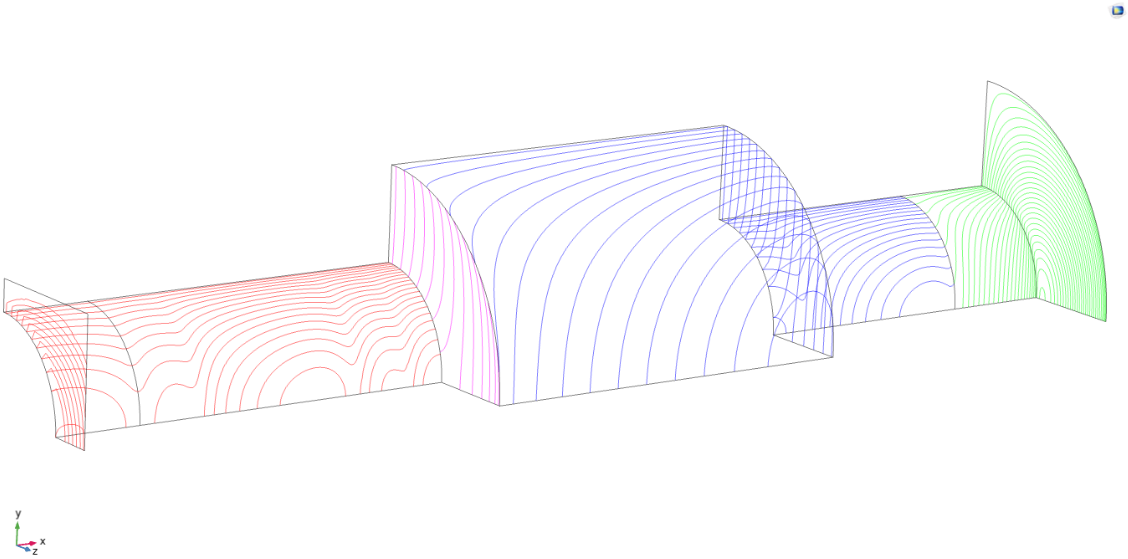


Figure 3.8: Calculated winding geometry for the total guide coil system.

3.3 Refining Wire Exports

Wire refinement was a multi-step process that required a few auxiliary programs and interfaces. Wire refinement refers to the process of converting the COMSOL contour export files into separate ordered points that constitute wires in the CSV and OBJ formats (see Appendix C for the structure of OBJ files) and cleaning the wire paths of redundant points. To accomplish this, contours were first separated into their distinct wires. From there, the points in the wire were ordered into paths that follow the current flow; this streamlined the process of modeling the coils in COMSOL for field verification. Once the points were ordered, they were exported in the CSV and OBJ formats; the CSV files were used to model the wires in COMSOL, and the OBJ files were used for visualizations in the rendering software Blender. To further aid in visualizations, additional programs were created for manipulating OBJ files (see Appendix **??** for a list and overview of the programs used to manipulate OBJs).

3.3.1 Sorting Contours into individual Wires

Each contour file contains seven lines of boilerplate from COMSOL that includes information about the model that generated the file, the date, the COMSOL version, and other useful bits of information. The headers for each column are on the eighth line. So all the pertinent data starts on the ninth line onward. To parse the contours into the unordered points for each wire, the Python function *ParseEquPot* (Parse Equal Potentials) was created (See Code 3.1). to form an array of IsoLevel values. The IsoLevel array indices correspond to the index of an array of wire points to which the spatial coordinates were added.

```
def ParseEquPot(fname):
    f=open(fname,’r’)
    Data=f.readlines()
    EquPotLines= [ ] #stores a collection of x,y,z points sorted by equipotential
    EquPots= [ ] #Is the list of IsoLevel values.
    for i in range( 8 , len( Data ) ): #starts at 8 to avoid boilerplate
        x,y,z,p=Data[ i ].strip().split()
        #this checks if the potential has already been added to the lines
        if(p in EquPots):
            index= EquPots.index(p)
            EquPotLines[ index ].append( [ float(x) , float(y) , float(z) ] )
        else:
            EquPots.append( p )
            EquPotLines.append( [ [ float(x) , float(y) , float(z) ] ] )

    return EquPotLines
```

Code 3.1: Decleration of the function ParseEquPot, which takes the file name (fname) as an input parameter and returns an unsorted list of points corresponding to the IsoLevel (EquPotLines).

3.3.2 Sorting Wires into Paths

Many approaches were tested for a generalized wire sorting algorithm that did not refer to the underlying geometry of the problem. However, this proved infeasible due to point distribution along wire paths, the geometry of the wires containing sharp angles, and contours containing multiple wire segments. The points defining the wire are not equally distributed along the wire path, meaning a distance-based sorting[2] would not work (See Figure 3.9). In short, it is possible to have a next-nearest neighbor on one side be closer than the nearest neighbor on the other. For equally distributed points, this is not possible because the distance for the left and right-hand sides' next nearest neighbor's points is greater than the distance to the nearest neighbor's points. There are two ways to account for this. One could provide a starting point for the wire, which would fix this issue. However, specifying a starting point would either lose generality because it references the geometry or requires giving a starting location by hand for all the wires, which defeats the purpose of automation. Or one could look to the dot product for information on the flow of the points. A solution to a simple wire would be to check the dot product of the vectors $v_{0\to c}$ and $v_{c\to sc}$, defined as the $v_{0\to c}$ being the starting point (p_0) to the closest point (p_c) and $v_{c\to sc}$ being the vector from p_c to the second closest point (p_{sc}). This would result in $v_{0\to c} \cdot v_{c\to sc} > 0$ when the points are in the same direction (or along the flow of the wire) or $v_{0\to c} \cdot v_{c\to sc} < 0$ when in the opposite direction (or against the flow of the wire); and $v_{0\to c} \cdot v_{c\to sc} = 0$ when forming a right angle. This would work if there were no sharp/right angles in the wire paths or separate wire segments for a contour. However, this is not the case for the wire contours solved in the last section. This proves to be more problematic because specifying a starting position with the algorithm does not fix the issue. The complications introduced in Figure 3.10 condemn a completely generalized sorting algorithm without reference to the coil geometries (that is, without considerably more time and effort). Sorting the contours into wires is only possible if limitations are made, and some elements of the geometry are explicitly specified. First, the wire segments were separated to contain no sharp

---

[2] Distance-based sorting refers to ordering points solely by the distances between points.

[3] This is only the case because p_1 occurs earlier in the file than p_4. If the points were flipped, the algorithm would still order correctly but with the opposite flow of points.

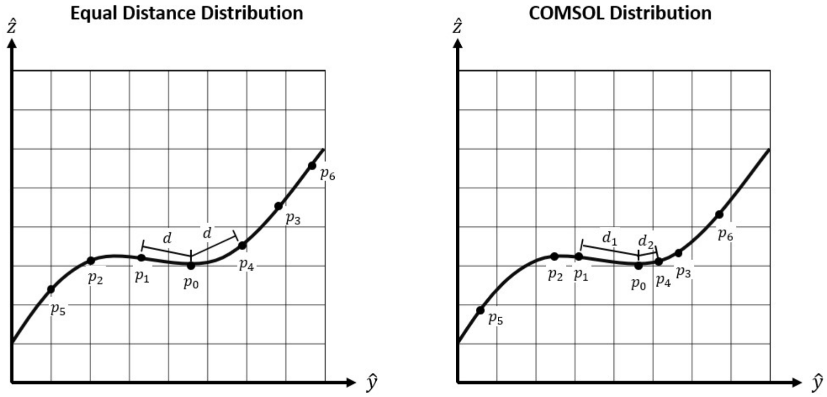


Figure 3.9: Demonstration of why distance-based ordering would not work. For the COMSOL files point distribution, the distance from p_0 to p_3 is less than that from p_0 to p_1. This would result in the algorithm incorrectly identifying p_0 as an endpoint or assigning p_4 as the start of the right-hand side of the wire and p_3 as the start of the left-hand side. Where an equal distribution would assign p_1 as the start of the left and p_4 as the start of the right[3].

edges. For the Guide Coils, the $\boldsymbol{B}_0$ Correction, CB Inner Face, CB Outer Face, and Annulus Coils were stored separate from the CB Coil, and the Inner, Middle, and Outer STCs. Second, information was provided about the maximum allowed space between points before they are separated into new wires[4]. The distance between points is much smaller than the distance between separate wires, so no situations occurred as described in Figure 3.10 III. Finally, an algorithm was used to specify the starting points of the wires, which required explicit reference to the coil geometries. The starting points were specified before the sorting algorithm was applied.

For simplicity, three functions were written to find the starting points: one for wires oriented along the x-axis, one for wires oriented in the yz-plane (this has a caveat), and the last for the Annulus Coil. The current flowed from the inside to the outside of the MSR; so for coils oriented along the x-axis (these are the CB Coil, the Inner, Middle, and Outer STCs), the point closest to the inside the MSR was the starting point (See Code 3.2).

---

[4] This is needed when specifying the currents in COMSOL because if a line was placed between points that should be separate wires, then current would also be applied to it.

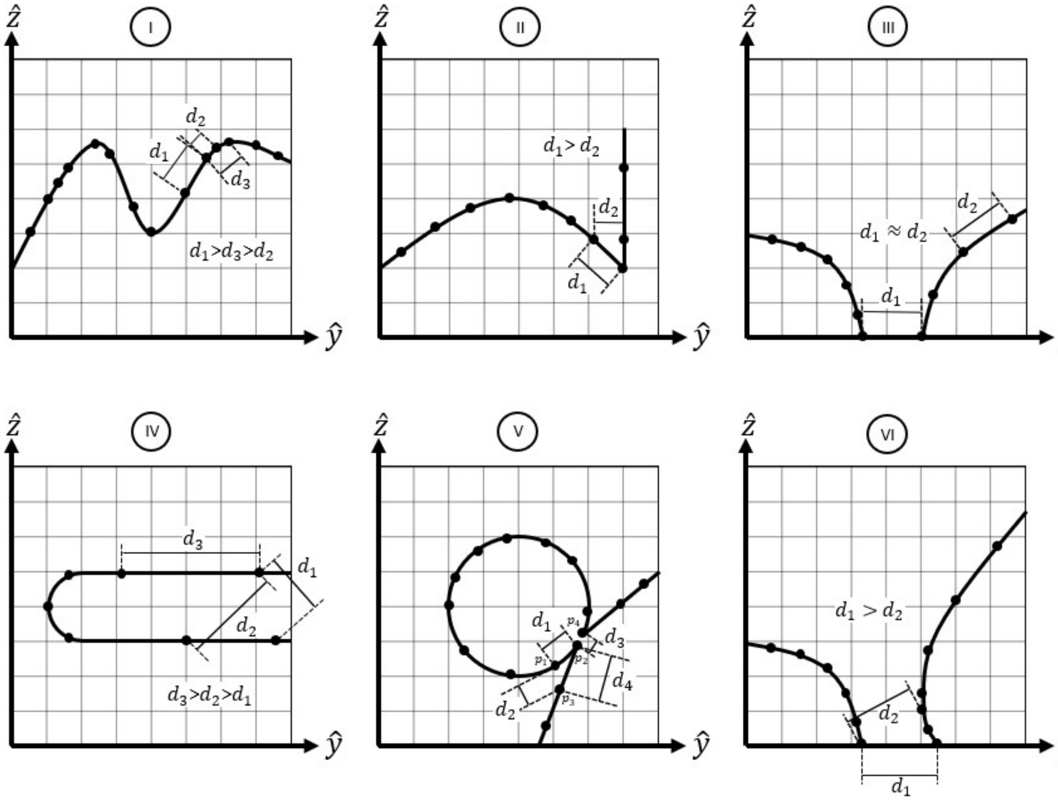


Figure 3.10: Shown are complications, some hypothetical, some actual to the guide coils, for a general sorting algorithm of non-equally distributed points. I.) The next nearest neighbor is closer than the nearest neighbor on the opposite side. This is solved by using a dot product. II.) Along the vertical line is closer and would have a positive dot product. This is solved by manually separating wires so no sharp angles occur or by normalizing the vectors before the dot product and setting thresholds on the dot product value. III.) Two Distinct wires have a separation of the same order as other point separations. This can be solved by limiting the dot product values. IV.) Has two wires separated by a distance smaller than the point separation, resulting in small positive dot product values for points that jump between wires. This could be limited by making the dot product value threshold smaller, but curvature segments would be separated into distinct wires, like on the left side of the wire. V.) No sorting method specified would work unless a starting point is specified and the sharp edges removed. $\hat{v}_{p_1 \to p_2} \cdot \hat{v}_{p_2 \to p_4} > 0$ and small, with a d_3 smaller than d_4 so specifying $p_2 \to p_3$ or that p_2 and p_4 are connecting points would be needed. VI.) Results in three wires being produced instead of two. This can be mitigated by introducing a second application of a starting point finding algorithm.

```
def StartingPointAlongX(Pts):
    #the starting point is the point with the smallest x-coordinate
    min_x=Pts[ 0 ][ 0 ]
    start_index= 0
    #loop to find the smallest x
    for i in range( 1 , len( Pts ) ):
        #if this x is smaller than the smallest found so far, then update the smallest
        if(Pts[ i ][ 0 ] < min_x):
            start_index= i
            min_x=Pts[ i ][ 0 ]
        #one wire had the two minimum x value points, so it needs to choose the point
            ↪ closest to y=0 out of the two.
        if(Pts[ i ][ 0 ]== min_x and Pts[ i ][ 1 ] < Pts[ start_index ][ 1 ]):
            start_index= i
            min_x=Pts[ i ][ 0 ]
    return start_index
```

Code 3.2: that describe all points in a wire.]Function that finds the starting point for points oriented along the x-axis. Takes an array of points ordered [x,y,z] that describe all points in a wire.

For wires oriented in the yz-plane (these are the $\boldsymbol{B}_0$ Correction, CB Inner Face, CB Outer Face, and Annulus Coils) sorting can be preformed my first specifying a starting point. A starting point is found by selecting the points at the radius of the STC (r_{STC}) or those closest to r_{STC} at $y = 0$. For all but the Annulus, finding the smallest y value was the simplest solution. The Annulus had two wires that do not contact the STC surface, thus having both endpoints at the same minimum $y = 0$. The Annulus Coil's starting points are found by searching for the smallest y value with a polar radius (ρ) smaller than some threshold radius ρ_T (See Figure 3.11). To find the starting point for all yz-plane oriented coils, except the Annulus coil, Code 3.2 worked with the x and y reversed, i.e., min_x was substituted for min_y and Pts[i][0] for Pts[i][1], etc. The new function bears the name StartingPointsInYZPlane. Here lies the caveat: this works for the CB Inner Face and the $\boldsymbol{B}_0$ correction coil[5], but not for the CB Outer Face. For the CB Outer Face, this algorithm resulted in

---

[5] This would not be the case if not for the wires endpoints being slightly higher in y than the starting points. This is not the case generally and would require revisiting if the potential changes.

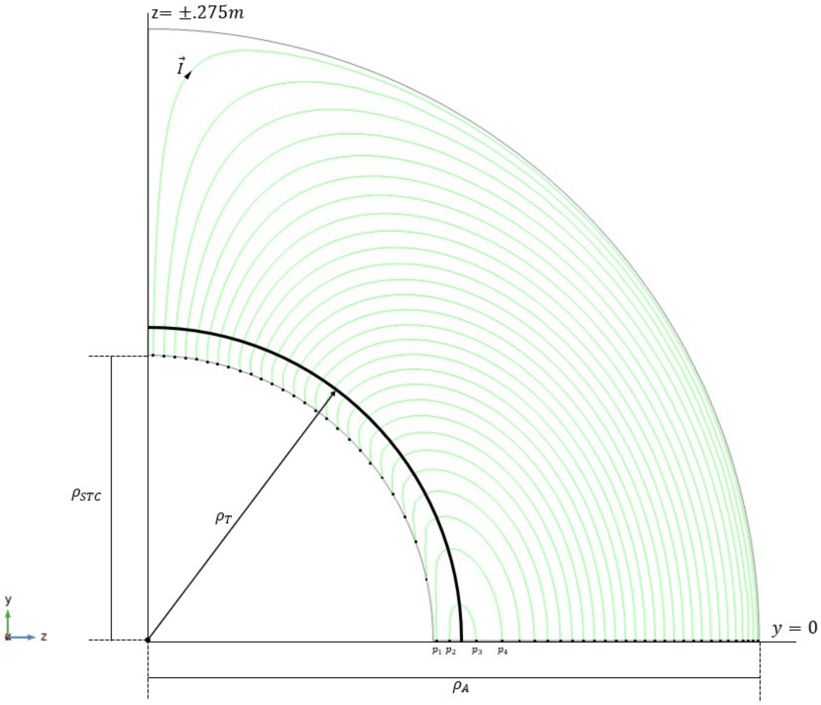


Figure 3.11: The Annulus Wires all end at $y = 0$, with two wires beginning and ending at $y = 0$ (the lines $p_1 \rightarrow p_3$ and $p_2 \rightarrow p_4$). To stop the degeneracy, only points less than $\rho_T = .115m$ are considered. This also helps with the ordering issue. If the starting points were instead defined $y = 0$ and $\rho > \rho_T$, this would cause the wires' flow to be opposite the currents' direction $\vec{I}$.

the endpoints, rather than starting points, when considering the current flow through the wires. To account for this, the resulting paths from the sorting were reversed. The starting points for the Annulus coils are found using Code 3.3.

To see if all the sorting was done correctly, the Code **??** in Appendix D Section D.2 was implemented to visualize the starting positions of the wires. This worked as a test before pulling the wires into COMSOL(See Figures 3.12 and 3.13). The visualizations were useful because they drew attention to small details that would have been overlooked otherwise.

With the starting points determined and verified, the points were ordered. Considering the potential problem cases shown in Figure 3.10, the elements of the algorithm had a maximum distance before splitting a contour into separate wires. They were sorted by the closest points unless an exception occurred. By manually separating

```
def StartingPointAnnulus( Pts , zOffSet ):
    start_index=-1
    start_point=[0,10,0] #set to be higher than any possible y value to find the minimum.
    for i in range(len(Pts)):
        test_rho=np.sqrt(Pts[ i ][ 1 ]**2 + (Pts[ i ][ 2 ] - zOffSet )**2)
        #.115 is a limit set by the degenerate Annulus coil paths.
        if(test_rho<.115 and Pts[ i ][ 1 ]<start_point[ 1 ]):
            start_index=i
            start_point=Pts[ i ]
    return start_index
```

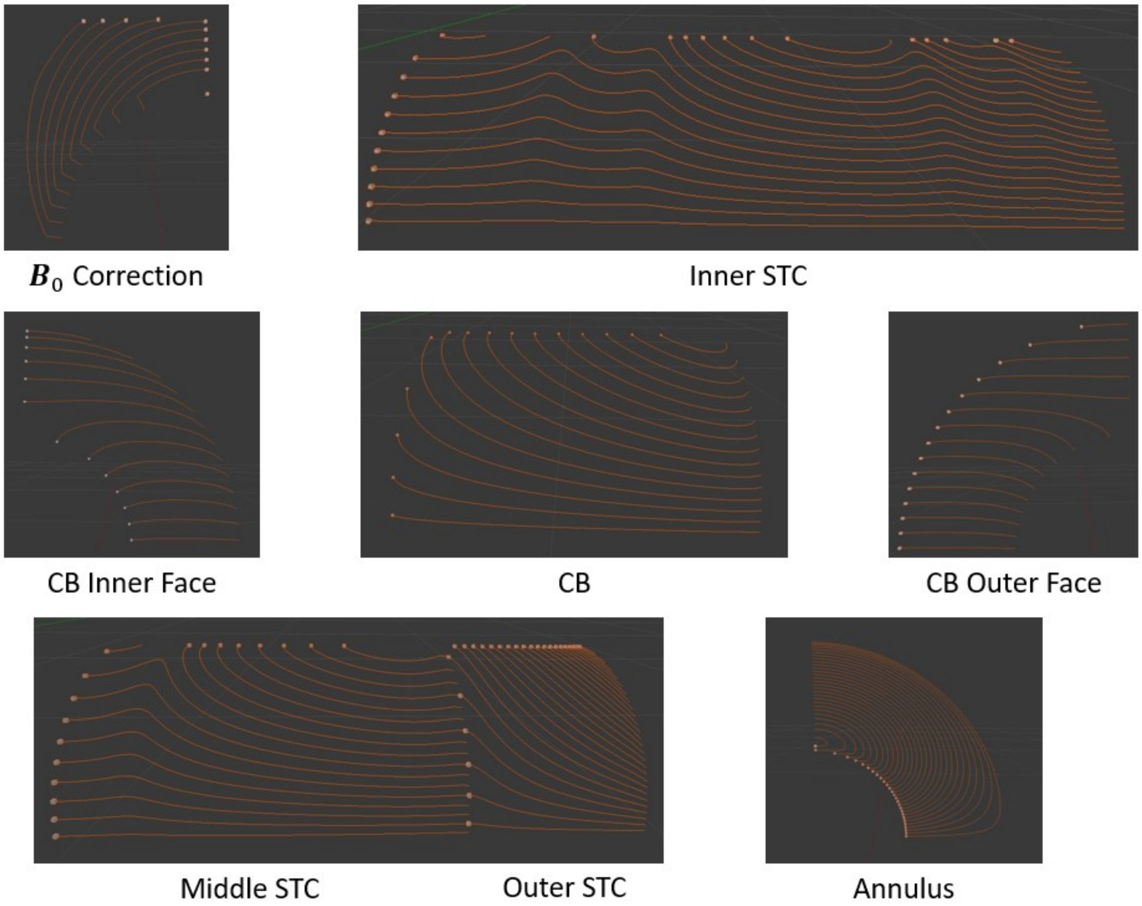
Code 3.3: The starting point of the Annulus wires are found by finding the smallest y values with $\rho < \rho_T = .115m$ where $\rho = \sqrt{y^2 + (z \mp .275m)^2}$. The term $\pm.275m$ comes from the zOffSet referenced in the code, depending on whether the wires are for the Top Annulus ($+.275m$) or Bottom Annulus($-.275m$).

the contours so that no path contained sharp angles, the need to check the dot product was avoided. However, due to a special case in the wires (See Figure 3.10 VI.), an additional call to a starting point function is necessary[6]. The function shown in Code 3.4 takes a line (or contour), which is an unordered list of points, two functions for finding the starting points, the maximum length before a wire split, and the z offset, which was $\pm.275m$ depending on if the coils are in the top or bottom STC region. The function then found the starting point for the specified contour orientation. After this, a distanced-based sorting was implemented while checking if the next closest point was far enough away to start a new wire segment. If a new segment was detected, the starting point for the new segment was found by a different starting point function (the second starting point function finds the point with the minimum y value instead of x. The only wires with splits were those oriented along the x-axis, and all had a starting point at $y = 0$.)

The sorting function was applied to each set of contours for the respective coils via the function defined in Code 3.5. With the contours sorted into wires, the lists were exported into OBJ and CSV formats for visualizations and COMSOL, respectively.

---

[6] The can also be avoided by checking the distance from both ends of the wire and adding the point to the head or tail of the list depending on which it was closest to. However, a second starting point call is simpler.



B_0 Correction

Inner STC

CB Inner Face

CB

CB Outer Face

Middle STC

Outer STC

Annulus

Figure 3.12: All the Guide Coil components are shown with their starting points displayed as small cubes.

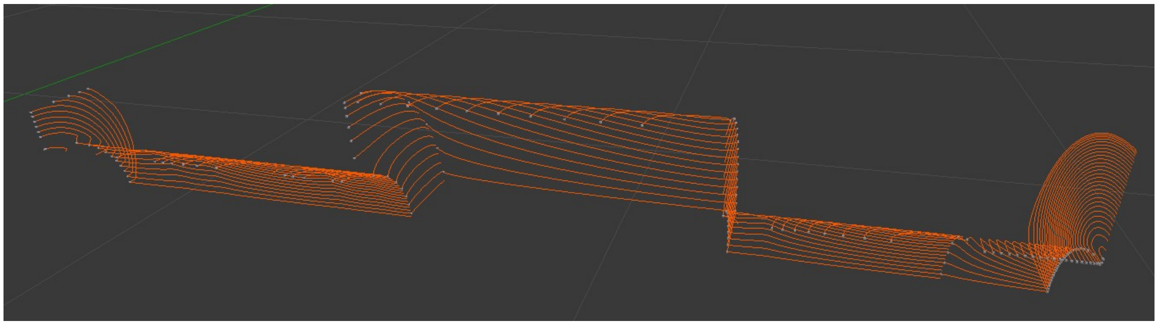


Figure 3.13: Total Guide coil visualization in Blender with starting points emphasized.

```
def OrderPoints(Line , Start_Point_Function , Split_Start_Point_Function , MaxLen,
    ↪ zOffSet):
    Pts_remaining=[Line[i] for i in range(len(Line))]
    #function that calls the correct format of function.
    start_Pt_index=Call_Function(Start_Point_Function,Line,zOffSet)
    del Pts_remaining[start_Pt_index]
    Wires= [ ] #list of Wires
    last_Pt=Line[start_Pt_index]
    Wire=[last_Pt] #a Wire is the ordered points
    for i in range(1,len(Line)):
        distaces_from_last_Pt=[np.linalg.norm(np.array(last_Pt)−np.array(pts)) for
            ↪ pts in Pts_remaining]
        ordered_distances=np.sort(distaces_from_last_Pt) #sorts descending
        test_next_point=Pts_remaining[np.where(distaces_from_last_Pt==
            ↪ ordered_distances[0])[0][0]]
        if(ordered_distances[0]<MaxLen): #if not greater than max length add poin
            last_Pt=test_next_point
            Wire.append(last_Pt)
            Pts_remaining.remove(last_Pt)
        else: #this only gets called if a new wire is being created.
            Wires.append(Wire)
            start_Pt_index=Line.index(Pts_remaining[Call_Function(
                ↪ Split_Start_Point_Function,Pts_remaining,zOffSet)])
            last_Pt=Line[start_Pt_index]
            Wire=[last_Pt]
            Pts_remaining.remove(last_Pt)
    Wires.append(Wire)
    return Wires
```

Code 3.4: Point ordering algorithm.

The exporting functions are defined in Appendix D Section D.1.

3.4 Modeling Coils In COMSOL from Paths

Wires were imported into COMSOL using the CSV format. From the CSV, an Interpolation curve was created in COMSOL. In half the top Guide coil, there were 175 distinct wire paths, with the symmetric design of the wires that resulted in 700 total wires to assign interpolation curves[7]. Some automation was applied here

---

[7] There is a way to import the 175 coils through mirroring their geometry to get the 700. This avoided importing all the wires individually. However, the mirroring function reversed the direction of the curve, and so the current applied to it, thus undoing the effort to get the order right.

```
#Start_Point_Function= lambda Line,zOffSet: SP.some_function()
#−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
def AddWires(Lines,Start_Point_Function,Split_Start_Point_Function,zOffSet,MaxLen
    ↪ ,Cube_Name_Boiler,Wire_Name_Boiler, Wires, Wire_Starting_Points ,
    ↪ Wire_Names, Cube_Names):
    for i in range(len(Lines)):
        #SW is the file that contains the wire sorting functions
        Wire=SW.OrderPoints(Lines[i] , Start_Point_Function ,
            ↪ Split_Start_Point_Function , MaxLen, zOffSet)
        for j in range(len(Wire)):
            #gets the starting points so cubes can be placed there
            Wire_Starting_Points.append(Wire[j][0])
            #names the cubes
            Cube_Names.append(f”{Cube_Name_Boiler}_{IO.get_numstr(3,i)}_{j}”)
            #names the wire
            Wire_Names.append(f”{Wire_Name_Boiler}_{IO.get_numstr(3,i)}_{j}”)
            Wires.append(Wire[j]) #adds the wire to the wires list.
```

Code 3.5: Function that takes the contours and splits them into ordered wire paths.

as well via the live link interface with MATLAB. With the built-in functionality, all wires CSVs were assigned an interpolation curve, and all curves associated with each segment of the guide coil were grouped. Once the wires were in the model, rerouting was added to conserve current. After specifying the currents, each wire and volume surrounding them was meshed. Finally, the fields were combined with the background fields (calculated in Chapter 2 Section 2.1) and compared to the total target field (calculated in Chapter 2 Section 2.2).

3.4.1 Importing Wires Into COMSOL and Rerouting

To reduce the size of the model geometry component, the guide coils were treated similarly to the MSR, in that all the geometry information was contained and edited in a single file, and a .mbin file is exported. The .mbin file was imported into new models to access the coil geometries. The MATLAB live link interface was used to automate the importing of wires via interpolation nodes (an example of one set of coils is shown in Code 3.6; this was implemented for each coil.). Another advantage

```
ind=1; %interpolation index
ng_ind=1; %node groiup index
Wires=dir(WiresDirectory);%looking at the wire SCV files
grpS=”grp”+string(ng_ind);
%Creating node group
model.component(”comp1”).geom(”geom1”).nodeGroup().create(grpS);
Group=model.component(”comp1”).geom(”geom1”).nodeGroup(grpS);
ng_ind=ng_ind+1; %increment node group index
for k=3:length(Wires)
    file=strcat(WiresFiles,’\’,Wires(k).name);
    icS=”ic”+string(ind);
    ind=ind+1;
    model.component(”comp1”).geom(”geom1”).create(icS, ”InterpolationCurve”);
    Geom=model.component(”comp1”).geom(”geom1”).feature(icS);
    Geom.set(”source”, ”file”);
    Geom.set(”filename”,file);
    Geom.set(”endcond”, ”zerocurv”);
    Group.add(icS);
end
grpS=”grp”+string(ng_ind);
%Creating node group for unions
model.component(”comp1”).geom(”geom1”).nodeGroup().create(grpS);
model.geom(’geom1’).run;
mphsave(model,COMSOL_fname);
```

Code 3.6: Function for importing coil geometries into COMSOL for a single wire directory. A running tab of the interpolation index and node group index is needed for additional function calls.

of defining the wires in a .mbin file was that each interpolation curve needed a unique identifier (in the code, this is the variable “icS”). However, there was no simple way to get the last unique identifier for a geometry element type. This meant that a blank COMSOL model was required each time the code was run, so either a complete model would need to be created any time the wires are updated, or a new geometry file was created. This code also assigned wires in the same coil into a node group, which allowed for easier navigation of the model. An additional code was implemented for all grouped wires to be joined via a *Union*. This code was semi-automated in that all the wire unique identifiers (the ic variables) were hard-coded due to an unresolved string error (a short snip of code is included as a reference for what the other 31

Unions looked like in Code 3.7). The code generates a COMSOL model that is shown in Figure 3.14.

```
model=mphload(COMSOL_fname);
model.component(”comp1”).geom(”geom1”).create(”uni1”, ”Union”);
Union1=model.component(”comp1”).geom(”geom1”).feature(”uni1”).selection(”input”);
Union1.set({ ’ic1’ ’ic2’ ’ic3’ ’ic4’ ’ic5’ ’ic6’ ’ic7’ ’ic8’ ’ic9’ ’ic10’ ’ic11’ });
model.geom(’geom1’).run;
mphsave(model,COMSOL_fname);
```

Code 3.7: Example of *Union* declaration for COMSOL wire groups.

Rerouting was performed on all wire paths that did not align between faces. This occurred when the faces were generated with different currents, i.e., Inner STC → CB Inner Face → CB and Middle STC → Outer STC. If currents were applied in the present configuration, current would not be conserved. The rerouting was added as a *parametric path* in the path of a semi-circle at each corresponding edge (See Figure 3.15). Because the model had a symmetry about the plane $y = 0$, any wires that ended on the plane were mirrored so that the current formed a closed loop.

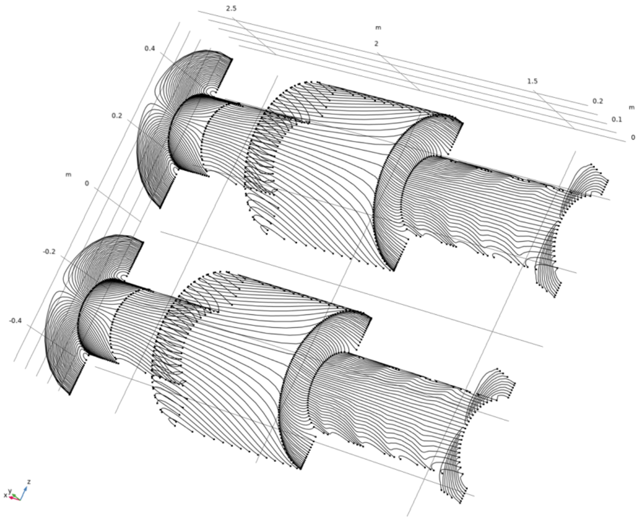


Figure 3.14: All wire geometries brought into COMSOL as interpolations of CSV files.

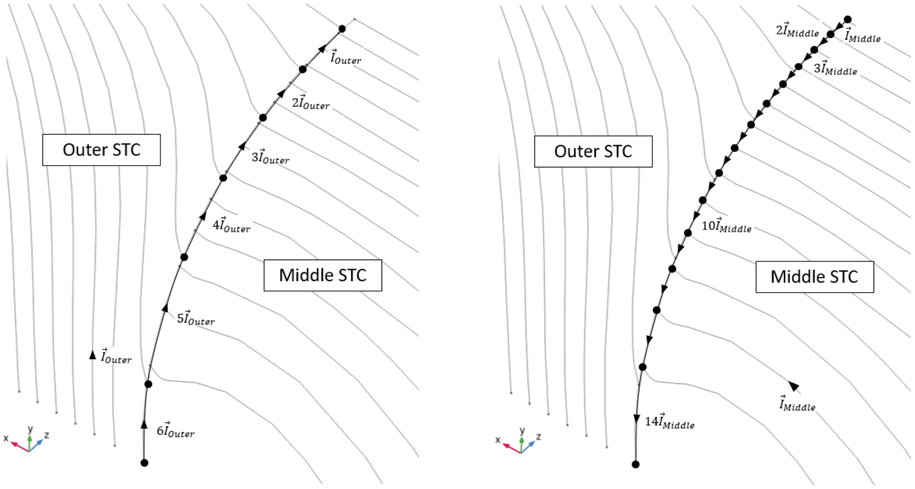


Figure 3.15: Current is conserved at each junction where the wire connects with the rerouting edge. This is done for both coils touching the rerouting edge individually to avoid adding complexity to the model.

3.4.2 Meshing The Coil Paths

Unlike the $\boldsymbol{B}_0$ coil, the Guide coil did not have a meshing surface that the wires resided on. This was due to the sharp points and small faces that resulted from putting the paths on a surface. Instead, the wires were surrounded by a thin shell. This allowed the wires to be finely meshed along the wire's edges with a transition region within the shell to connect with a coarser mesh outside (See Figure 3.16). The wires were meshed with *distribution* elements set so that the distance between nodes along the wire was comparable to distances between adjacent wires (See Figure 3.17). The smallest element separation among the wires determined the mesh *size* on and inside the shell. After meshing the wires, the shell was meshed with *distribution* elements along all edges, a *free triangular* element on its surface followed by a *free tetrahedral* element in the volume.

3.4.3 Field Exports and Adiabaticity

Figure 3.18 shows B_x and B_z on the central axis of the STC. The Guide coils com-

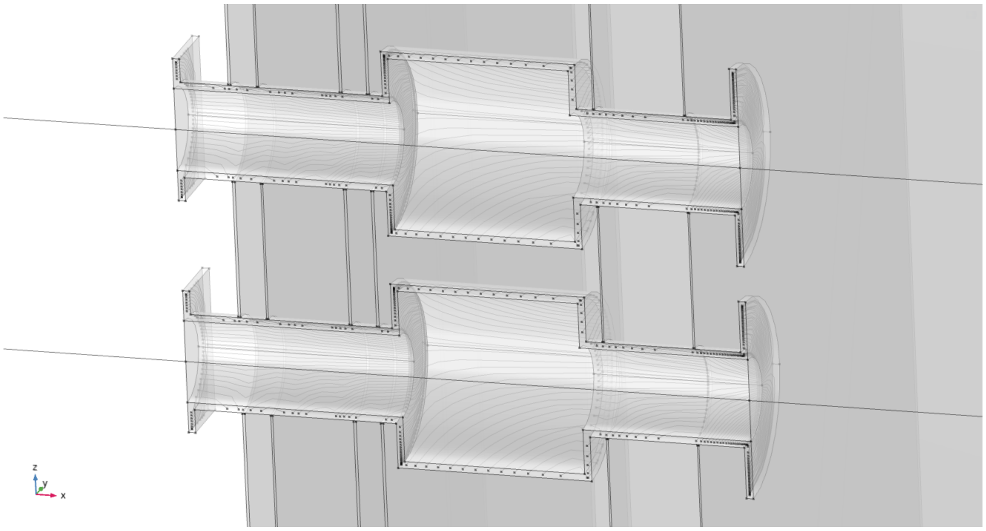


Figure 3.16: The Guide coil meshing shell with coil paths inside with the MSR surrounding it.

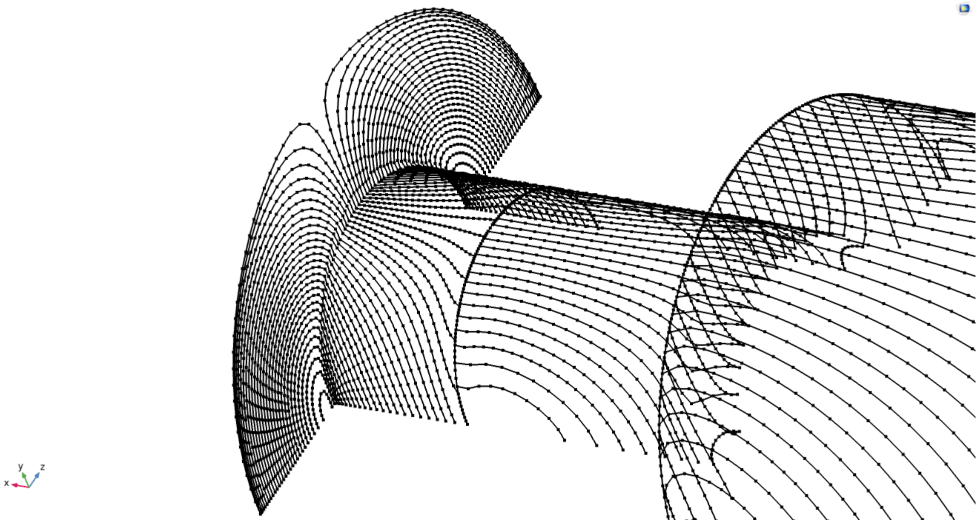


Figure 3.17: Segment of the Guide coil meshed so that nodes spacing along the wire's edge are roughly the same as the distance between wires.

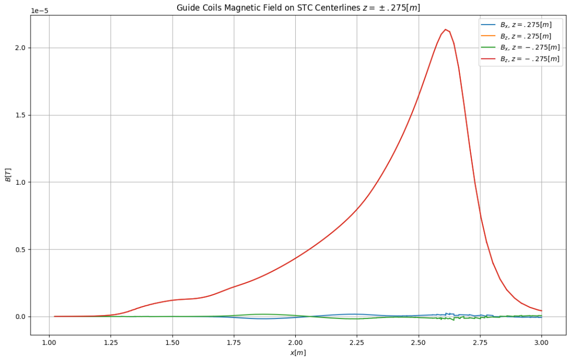


Figure 3.18: Guide coil fields on the center lines of the STC ($z = \pm.275$)

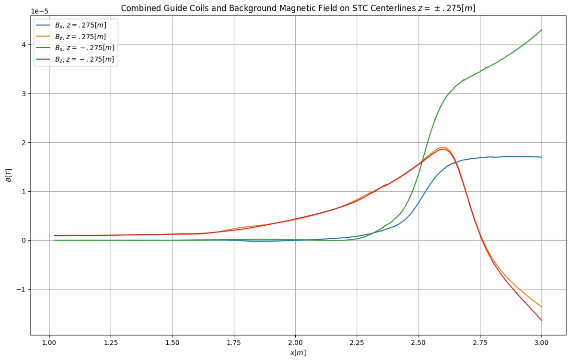


Figure 3.19: Guide coil fields with background field from Figure 2.13 on the center lines of the STC ($z = \pm.275$)

bined with the background fields from Section 2.1 are shown in Figure 3.19. This produced an adiabaticity plot shown in Figure 3.20. This is contrasted with the target adiabaticity shown in Figure 3.21. The designed fields produce a final spin polarization, determined by spin tracking simulations produced by Geza at PSI, in excess of 99.3%, which exceeded the design requirements.

Copyright© David C. Bowles, 2024.

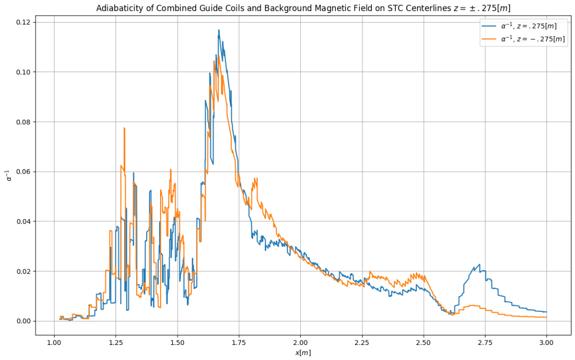


Figure 3.20: Adiabaticity of guide coil with background fields on the center lines of the STC ($z = \pm.275$)

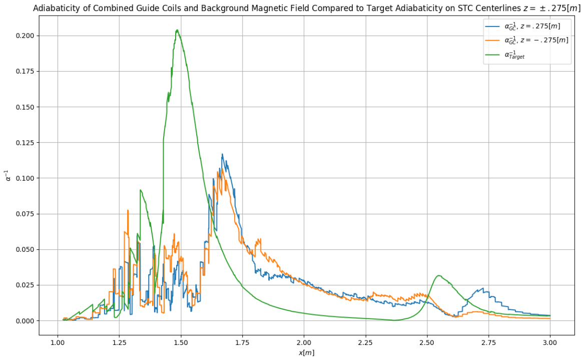


Figure 3.21: Adiabaticity comparison of total guide coil field and the target field. Additional optimization brought the adiabaticity at the start of the MSR down further.

Chapter 4 Building the Coils & Results

Three construction methods were considered to take the coils from the modeled object to the actual physical coils. In the first method, as a prototype, the wires were wound by hand onto paths printed on sheets and are structurally reinforced by a semitransparent meshed plastic canvas. The second method is to 3D print shells into which the wires can be placed. The third method is to have the paths etched into rigid and flexible PCBs.

4.1 Images of Wire Paths For Hand Wound Coils

Winding the wires by hand requires an image of the coil paths to which the wires can be affixed. Because some of the coils are on the surface of a cylinder and the image is flat, some work was required to get the paths in a printable format that preserves the spatial distances. To accomplish this, points have their y, z components mapped

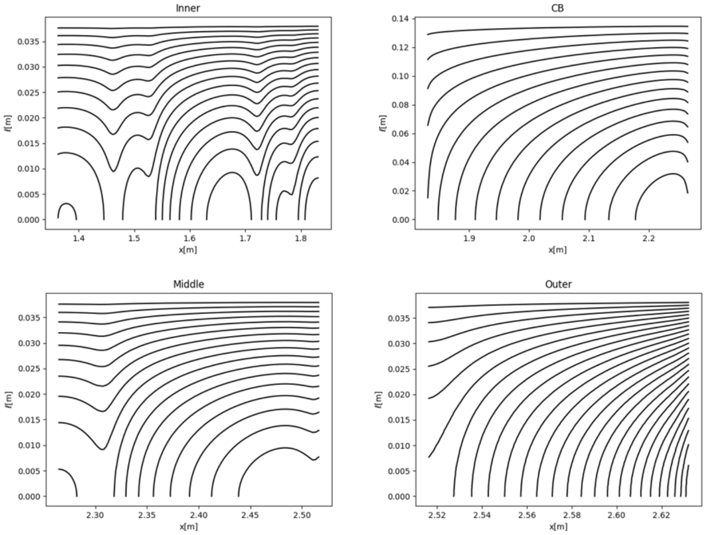


Figure 4.1: Images of the coils along the x-axis adjusted on the vertical axis of the images to represent the arc length of the respective coil.

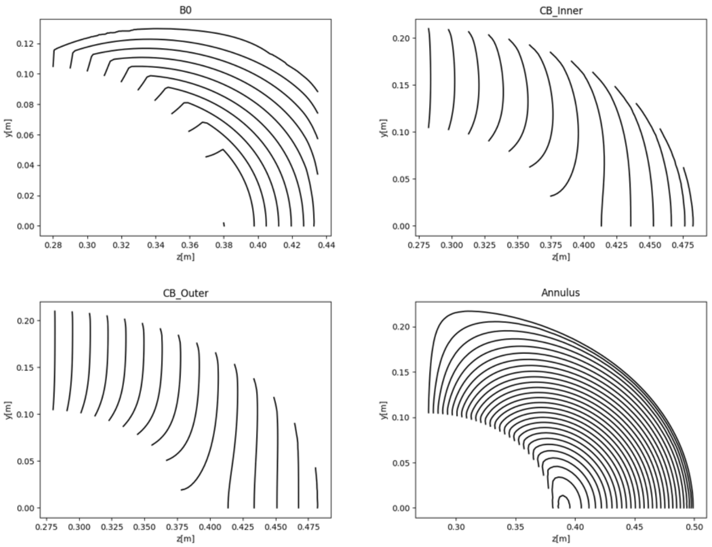


Figure 4.2: Images of the coils in the yz-plane

to an axis corresponding to their arc length on the cylinder surface by Equation 4.1.

$$\ell = \rho\theta = \rho \tan^{-1}\left(\frac{y}{z}\right) \tag{4.1}$$

Where ρ can be ρ_{CB} or ρ_{STC} depending on the coil being converted. For coils oriented along the x-axis, images would be ℓ vs x (See Figure 4.1). For any coil that is oriented in the yz plane, no conversion was necessary (See Figure 4.2). Prototype coils were fabricated during Spring 2023 by first-year undergraduates Gracie Burrows, Luke Cross, and Makaya Brashares in the UK STEMCats program, with help from sophomore Zack Wasson and senior Gabija Ziemyte. For fabrication, sheets of plastic canvas (for yarn craft) were mounted on cardboard frames for structural support and glued together. Printouts of the winding pattern were glued underneath and could be seen through the mesh of the plastic canvas(See Figure 4.3). The inner and junction coils were wound with 24 AWG magnet wire, and outer coil with 18 and 22 AWG segments. The coils were installed in the MSR with help from PSI staff. A picture of the Annulus and $\boldsymbol{B}_0$ coils installed are shown in Figure 4.4. Additionally pictured is the CB and one of the CB face coils installed around the bar that supports the

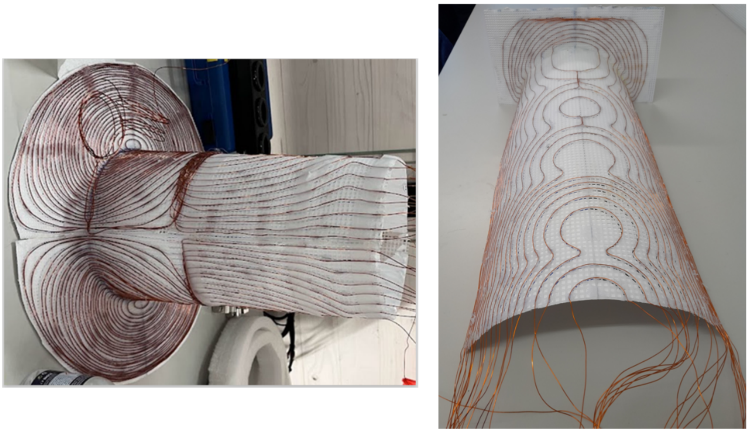


Figure 4.3: Wound coils on plastic mesh.

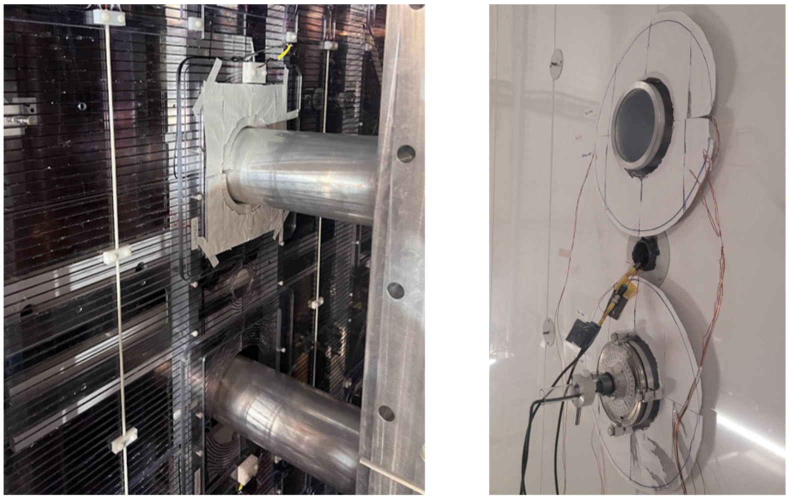


Figure 4.4: Installed $\boldsymbol{B}_0$ Coil on left. Installed Annulus Coil on Right. Both are pictured installed in place in and on the MSR.

CB in Figure 4.5. Magnetic field maps of each coil were acquired, sliding a 3-axis fluxgate along a tube through the guide by hand, positioned at three longitudinal positions. The measured field and harmonic interpolation are shown in Figure 4.6. These maps were used to determine the polarity and tune the current in each segment. Preliminary polarization measurements of UCNs passing into and back out of the precession chamber indicate spin transport efficiency in excess of 99%, which satisfies the design requirements.

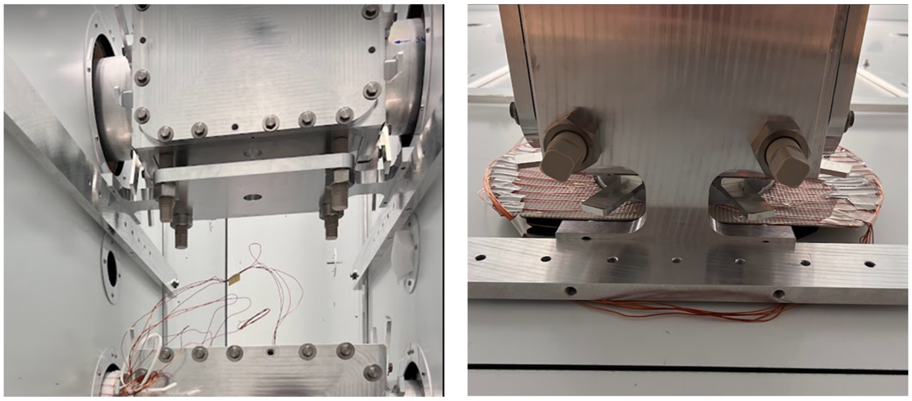


Figure 4.5: CB and CB face Coil around CB support bar

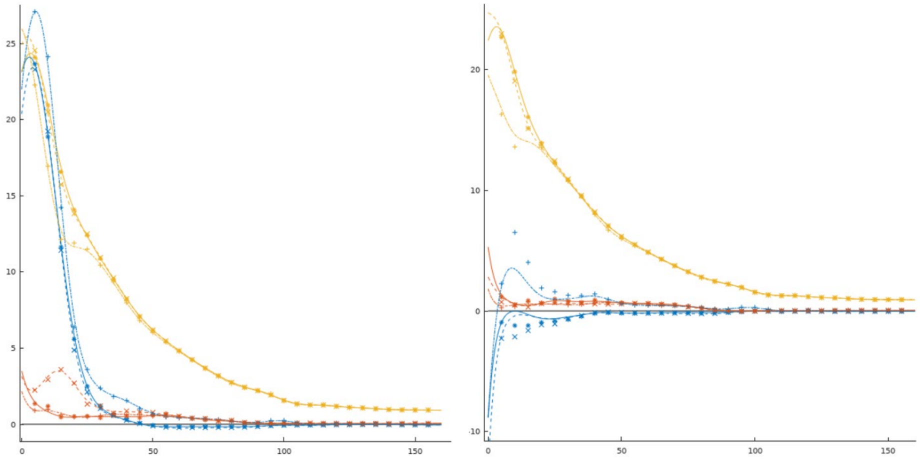


Figure 4.6: Measured fieldmap [μT] on the top (left) and bottom (right) guides, with harmonic interpolation curves. The blue, red, and yellow curves are B_x, B_y, B_z, respectively. The solid line is scanned along the center of the guide, dashed lines 5 *cm* to the left, and dot-dashed lines 5 *cm* below.

4.2 3D Printing Coil Shells

Alternatively, coils can be constructed via 3D printing technologies. To affix wires to a surface in a predetermined path, one could either create trenches on the surface for the wire to be pushed into, clips that hold the wire in place could be added to the surface to direct the wire flow or some combination of both. The type of printer needs to be considered as well. Fused Deposition Modeling (FDMs) printers have an extruder head that moves in a determined path while depositing filament to create the objects. These printers are cheap to purchase, the material is low cost, are relatively quick to print, and have large print volumes. This comes at the cost of limited precision, making clips infeasible (unless small extruder heads are installed, which comes at the cost of long print times). Stereolithography (SLA) printers are expensive to purchase at the scale required to make a coil, have toxic waste, and have long build times. However, the resolution of SLA printing is 0.001″ or better, and print time scales only with size, not so the desired detail. This results in high precision wire placement and a small expected deviation from actual fields. The accuracy of the fields and the time constraints on production time can inform what type of printer to use.

4.2.1 Trenching Wires On Surface

To produce a coil's shell that wires can be placed inside, three steps must take place. First, the surface on which the wire paths will be carved must be modeled. This is accomplished using either AutoCAD or Blender. Second, the wires must be converted from paths into 3D objects that can be subtracted from the surface. This has been accomplished via Python code that does the conversion. Finally, the new 3D wire objects are subtracted from the surface, leaving trenches for the wire. Modeling the surfaces depends on the geometry and the structural needs of the coil. In the case of the Guide coils designed in Chapter 3, a combination of thin hollow half-cylinders and half-punctured disks could serve as the surfaces in which wire paths are carved. The choice of semi-cylinders accommodates UCN guide tube, allowing the coils to be clamped around them.

To create the wire trenches, the wire's path and gauge, along with the surface normal along the wire path, are required. The gauge of the wire determines the width of the trench; the goal is to make the trench wide enough that the wire can fit taut inside so as not to slip out. For the wire to be given a volume, a shape must be specified to be swept along the wire's path. This shape is typically a square, but an

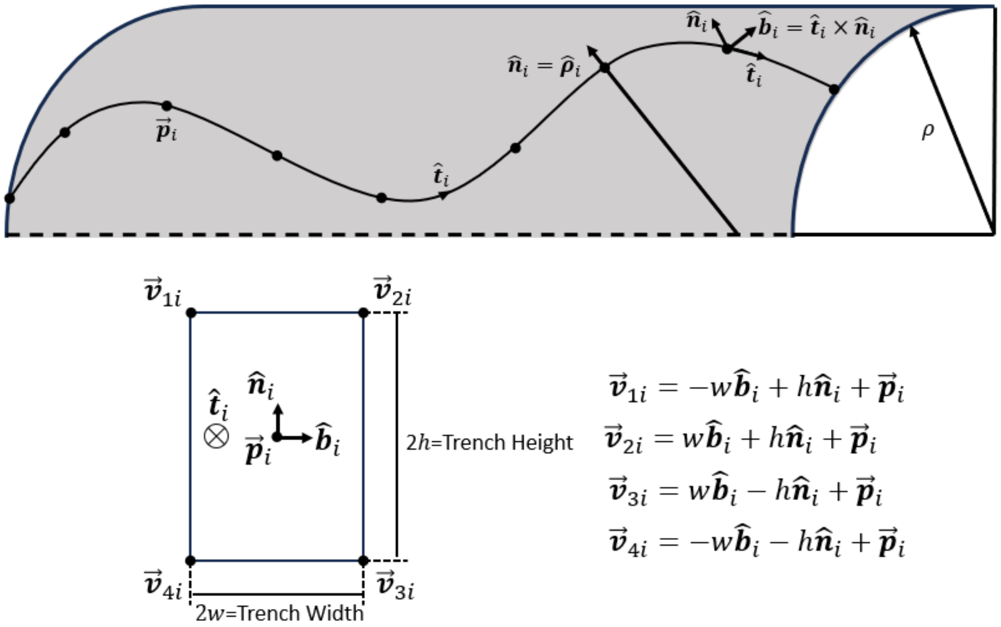

$\hat{n}_i$
$\hat{b}_i = \hat{t}_i \times \hat{n}_i$
$\hat{n}_i = \hat{\rho}_i$
$\hat{t}_i$
$\vec{p}_i$
$\hat{t}_i$
ρ
$\vec{v}_{1i}$
$\vec{v}_{2i}$
$\hat{n}_i$
$\hat{t}_i$
$\vec{p}_i$
$\hat{b}_i$
2h=Trench Height
$\vec{v}_{4i}$
$\vec{v}_{3i}$
2w=Trench Width
$\vec{v}_{1i} = -w\hat{b}_i + h\hat{n}_i + \vec{p}_i$
$\vec{v}_{2i} = w\hat{b}_i + h\hat{n}_i + \vec{p}_i$
$\vec{v}_{3i} = w\hat{b}_i - h\hat{n}_i + \vec{p}_i$
$\vec{v}_{4i} = -w\hat{b}_i - h\hat{n}_i + \vec{p}_i$

Figure 4.7: At points $\vec{p}_i$ along the wires path the tangent $\hat{t}_i$, the surface normal $\hat{n}_i$ and binormal $\hat{b}_i$ vectors are calculated or specified. From this basis, the rectangles swept along the wire's paths can be oriented by representing the vertices that compose the rectangle around $\vec{p}_i$.

oval or any other shape would suffice as long as the wire fits and will not come loose. The path and normal at each point define a basis for the shape to be constructed (See Figure 4.7). At points along the wire path, components of a trench can be added where the shape's surface normal is congruent with the wire's tangent. The shape's vertices can be expressed as linear combinations of the wire's normal and binormal vectors. Once a collection of the shapes has been added and oriented along the wire's path, the shapes can be connected together into an object resembling a distorted bar or tube. Care is required when setting the surface normals for the faces along the distorted bar. If a surface normal is facing the interior of the bar, this object is not considered "airtight" and will not subtract from the surface properly. This can be addressed by declaring the unit normals in the OBJ file and checking the face direction; however, the surface normals used to generate the shapes that constitute

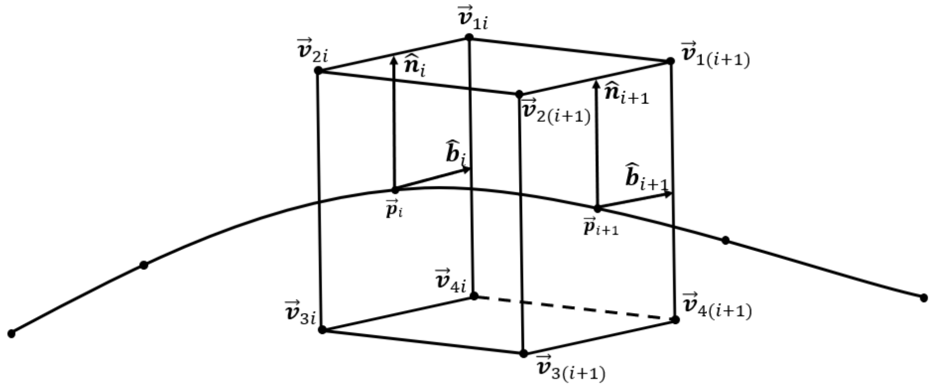


Figure 4.8: For the points $\vec{p}_i$ and $\vec{p}_{i+1}$ four faces need declaring in the OBJ: f_1 $\{\vec{v}_{1i}\ \vec{v}_{2i}\ \vec{v}_{2(i+1)}\ \vec{v}_{1(i+1)}\}$, f_2 $\{\vec{v}_{1i}\ \vec{v}_{1(i+1)}\ \vec{v}_{4(i+1)}\ \vec{v}_{4i}\}$, f_3 $\{\vec{v}_{4i}\ \vec{v}_{4(i+1)}\ \vec{v}_{3(i+1)}\ \vec{v}_{3i}\}$, f_4 $\{\vec{v}_{2i}\ \vec{v}_{3i}\ \vec{v}_{3(i+1)}\ \vec{v}_{2(i+1)}\}$. In general, $\hat{e}_i \neq \hat{e}_{i+1}$ where $\hat{e} = \{\hat{t}, \hat{n}, \hat{b}\}$ so the face normals for $\{f_1, f_2, f_3, f_4\}$ do not equal either of the binormals or surface normals. So, the order of f_i described above is needed to specify the face normals in the OBJ file.

the bar are not the same as the face's surface normal generated by connecting the vertices between shapes(See Figure 4.8). Instead, using the fact that vertex order forms a right-hand basis when determining the faces' normal, vertices can be ordered in face declaration to avoid the issue (See Appendix C). Once complete, this object can be subtracted from the coils surface thus forming the trench.

The surface and the wires 3D object are imported into Blender to complete the subtraction. A subtraction of the objects results in a 3D printing-ready object (See Figure 4.9).

4.2.2 Clip Generation and Placement

For wires that are too close to form trenches or that need extra secure attachments, a clip for the wire is preferable. A clip should be able to hold a wire in place and orient the wire along the path. As of the writing of this paper, no finalized clip design has been deemed optimal, but the general scheme of the design is presented in Figure 4.10. Clip Placement should depend on the geometry of the wire. Clips

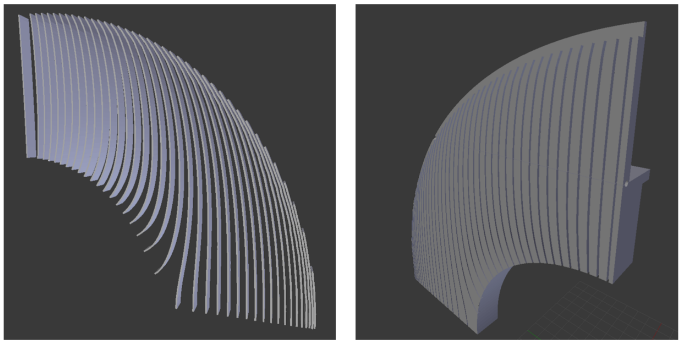


Figure 4.9: On the left are the 3D wire objects that, when subtracted from the CB face's surface, result in the image on the right. Note that the height of the trenches on the left differs from the shell's trench depth. Generally, a large clearance from the surface being subtracted is required.

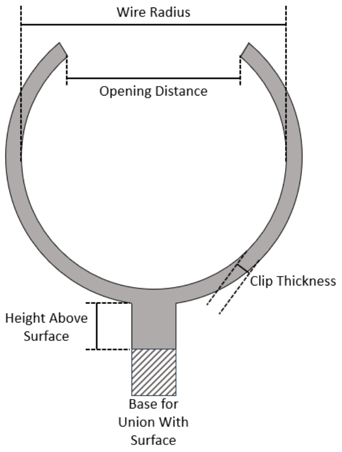


Figure 4.10: Described are the basic considerations for a clip's design. The Clip should allow the wire to be pressed inside without the clip breaking, and pressing the wire in should not make the clip loose. This describes the balance between the opening distance and the thickness of the clip. Additionally, the clip should be left space to bend while having defined points of contact with the surface for the union.

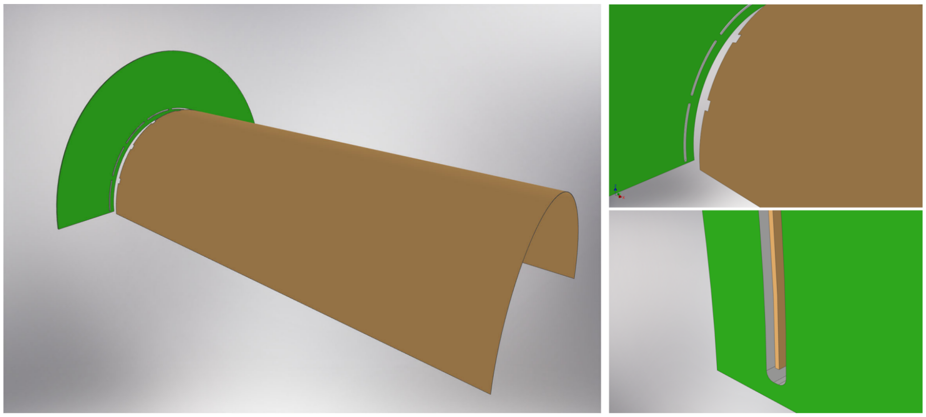


Figure 4.11: Construction of spin transport coils from rigid and flexible PCBs. The flexible circuit is inserted through gaps in the rigid board and the two circuits are soldered together.

should be placed frequently along the wire's path to not deviate from the shape of the wire. This can be automated by considering the curvature of the wire confined to the surface and specifying a maximum length before an additional clip is placed. Points of high curvature should have a higher density of clips, whereas low curvature can have the minimum density determined by the minimum spacing length.

4.3 PCB Coils

The most current design, as of the writing of this paper, is for the final production Guide coils to be fabricated using a combination of flexible and rigid printed circuit boards (PCB), soldered together to form cylinders with flanges as shown in Figure 4.11. This will replace the hand-wound coils, initially for the $\boldsymbol{B}_0$, Inner, CB Inner, CB, and CB Outer coils, later updating the whole system if further improvement to the field is necessary. The middle, outer, and Annulus coils have larger fields; thus, the spin transport is less sensitive to gradients in these regions. The larger currents in the outer coil make it less amenable to the flexible printed circuit design. Now that the switch is installed, installation of the outer coil would require complete removal of the junction box and glass guides. The new guides will be manufactured at a commercial PCB facility in early 2024, and installed in the MSR in March before the

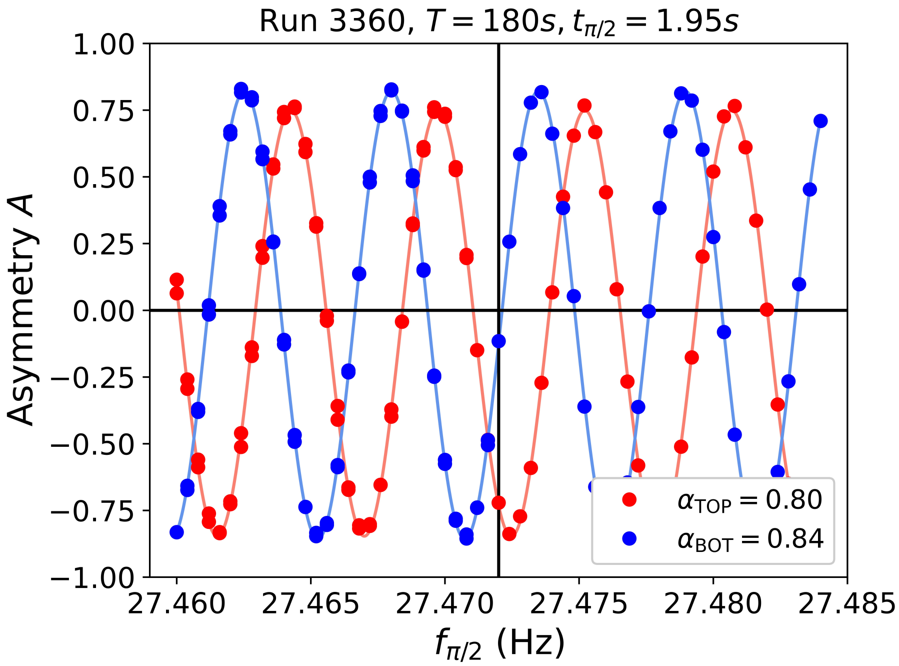


Figure 4.12: Plot courtesy of Efrain Patrick at PSI. Shown is the visibility/Asymmetry A of the Ramsey Cycle.

2024 beam cycle begins.

4.4 Results

The prototype coils were installed at PSI and preliminary data was taken on the visibility of the Ramsey cycles preformed. The measured visibility was in access of 80% as shown in Figure 4.12, which was the design goal for the experiment. This visibility was produced with the prototype coils, which were designed with outdated μ-metal layer positions. Despite this shortcoming, the simulated neuron polarization was 98.08% (See Figure 4.13). Geza Zsigmond produced this result at PSI in the simulated prototype field. The updated design improved the field uniformity and simulated spin polarization. All that was changed between the old model was the positions of the μ-metal and better accounting of rerouting paths. The resulting fields are shown in Figures 4.14 and 4.15. The new design produced polarization in access of 99.2%, again as simulated by Geza Zsigmond as shown in Figure 4.16

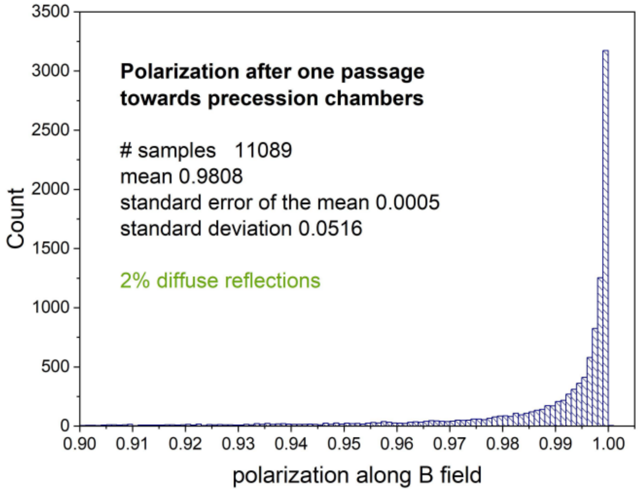


Figure 4.13: Plot Courtesy of Geza Zsigmond at PSI. Shown is the simulated polarization of the neutron through a single pass from the SCM to the precession chamber to the spin analyzers for the prototype coil design.

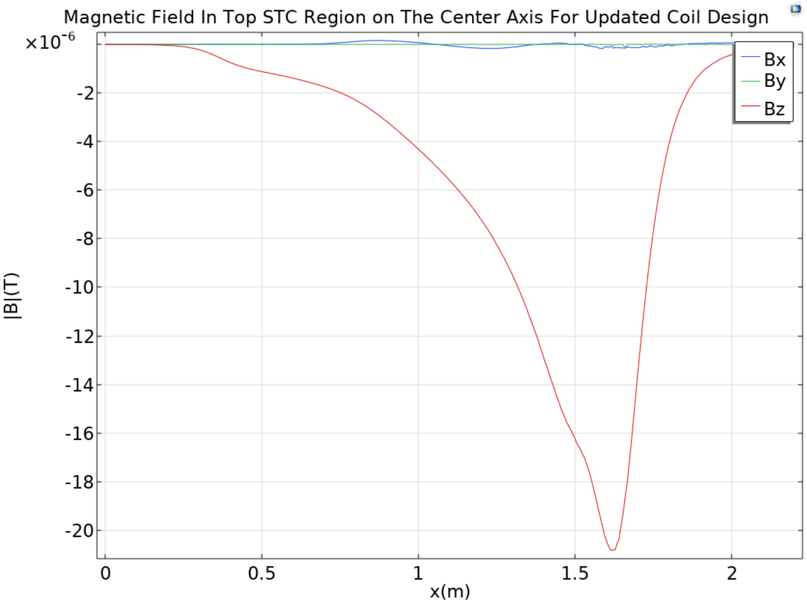


Figure 4.14: Magnetic field in Top STC Region along the center line at $z = .275$m.

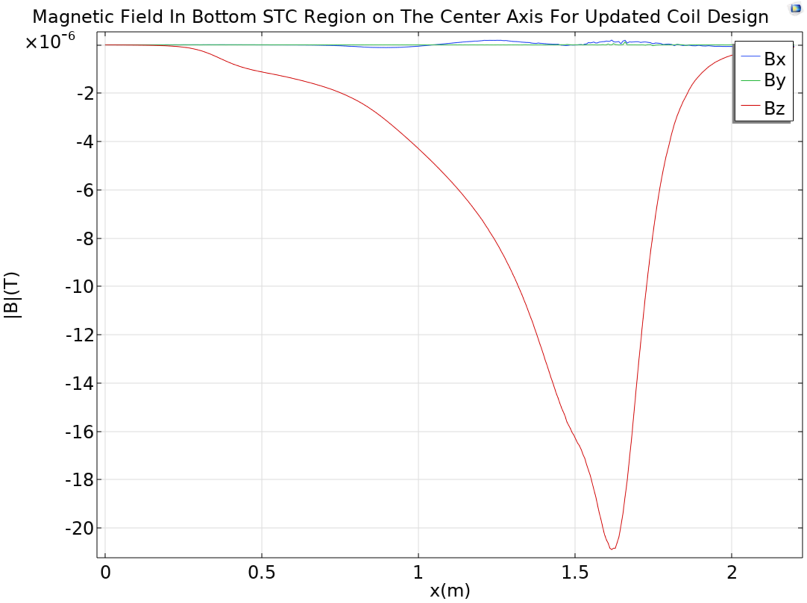


Figure 4.15: Magnetic field in Bottom STC Region along the center line at $z = -.275$m.

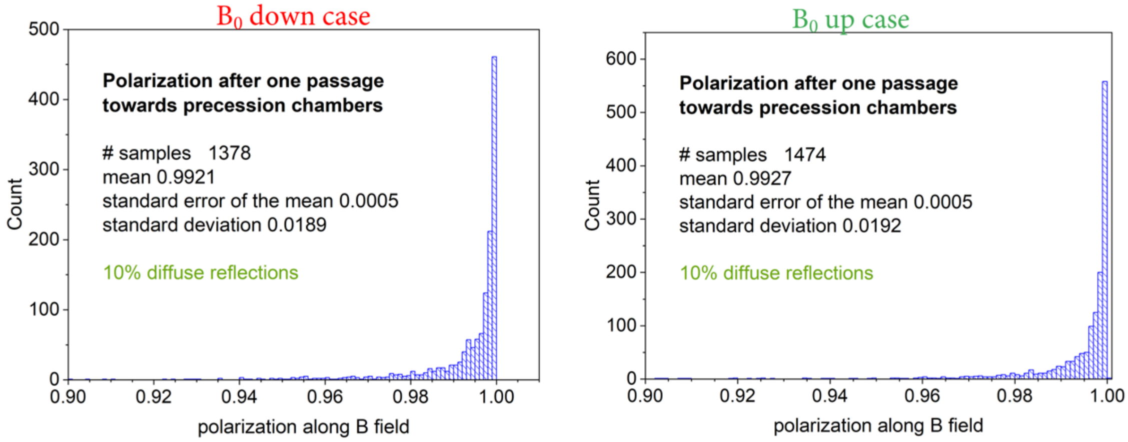


Figure 4.16: Plot Courtesy of Geza Zsigmond at PSI. Shown is the simulated polarization of the neutron through a single pass from the SCM to the precession chamber to the spin analyzers for the updated coil design.

Copyright© David C. Bowles, 2024.

Appendices

Chapter A Technical Specifications of COMSOL Models

There are four components in the COMSOL model: The Magnetic Shield Room (MSR), the superconducting magnet (SCM), the B_0 Coil, and the Spin Transport Coil (STC).

A.1 The MSR

The MSR comprises six μ-metal layers. The μ-metal layers are labeled such that the layer furthest from the MSR's center is Layer 1. Each layer can be considered a box with a given thickness, width, height, depth, and center. The width is the extent along the x-axis and is measured from the outside edge to the outside edge. The depth is measured along the y-axis, and height is along the z-axis. The layer dimensions are listed in Table A.1, and the μ-metal thicknesses are in Table A.2 shown below. Each box's centers are not aligned, though only layers 1 & 2 are off-center; they are shifted up by 300 mm along the z-axis.

The last feature that needs to be specified for the MSR is the holes for the STC on the positive x side of the MSR. The center lines for the STC sit at ±275 mm. The radius of the STC is 105 mm, and there is a 10 mm gap between the STC and the μ-metal, making the hole's radius in the MSR 115 mm. A complete model for the MSR is shown in Figure A.1.

A.2 The SCM

The SCM consists of two hollow cylinders labeled Inner and Outer (Shown in Figure A.2). These cylinders are approximations of the SCM installed at PSI as the exact details are proprietary. However, the fields provided for the SCM did not include an environment with a high permeable magnetic material object, like the MSR, not to

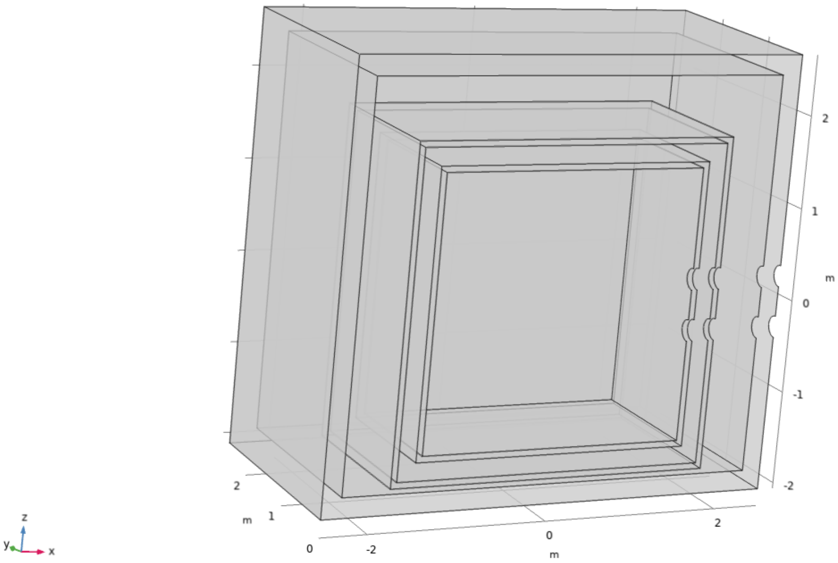


Figure A.1: Shows the MSR modeled in COMSOL. The holes in the MSR on the right-hand side of the model allow the STC to extend into the interior.

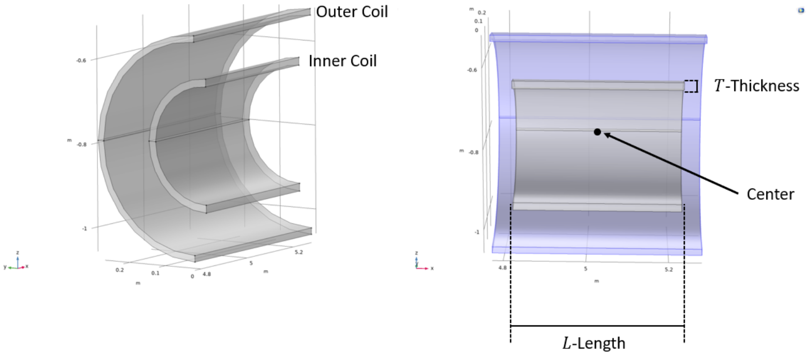


Figure A.2: Shown on the left are the names of the corresponding cylinders. On the right shows that the lengths of the respective cylinders are not the same but their centers correspond.

Layer	Dimension	Value
Layer 1	Width	5044.5 mm
	Depth	5044.5 mm
	Height	4717.0 mm
Layer 2	Width	4623.5 mm
	Depth	4623.5 mm
	Height	4284.0 mm
Layer 3	Width	3580.0 mm
	Depth	3580.0 mm
	Height	3576.5 mm
Layer 4	Width	3453.0 mm
	Depth	3453.0 mm
	Height	3447.0 mm
Layer 5	Width	3065.0 mm
	Depth	3065.0 mm
	Height	3059.0 mm
Layer 6	Width	2936.0 mm
	Depth	2936.0 mm
	Height	2930.0 mm

Table A.1: μ-metal layer dimensions: Dimensions are measured from outside edge to outside edge and, therefore, include the thickness of the μ-metal in the listed dimension.

μ-metal layer	Thickness
layer 1	3.75 mm
layer 2	3.75 mm
layer 3	6 mm
layer 4	6 mm
layer 5	4.5 mm
layer 6	6 mm

Table A.2: μ-metal thicknesses.

mention its geometry. The dimensions, location, and current are shown in Table A.3

A.3 The $\mathbf{B}_0$ Coil

The $\mathbf{B}_0$ coil was provided in a COMSOL file; thus specifics of the geometry are not generalized. Important to the modeling of the $\mathbf{B}_0$ coil was the current $I = 11.75[mA]$. All other geometry is provided in the corresponding files that accompany

Parameter	Value	Description
r_{Inner}	$159.1818[mm]$	radius of the outside of the Inner cylinder
L_{Inner}	$405.365[mm]$	Face to face length of Inner cylinder
T_{Inner}	$18.4[mm]$	Thickness of the Inner cylinder
T_{Inner}	$1.2926 \times 10^6[A]$	Current in Inner cylinder
r_{Outer}	$268.9948[mm]$	radius of the outside of the Outer cylinder
L_{Outer}	$506.564[mm]$	Face to face length of Outer cylinder
T_{Outer}	$15.48[mm]$	Thickness of the Outer cylinder
T_{Outer}	$-0.4242132 \times 10^6[A]$	Current in Outer cylinder
x	$5027[mm]$	Location of cylinder center on x-axis
z	$-789[mm]$	Location of cylinder center on z-axis

Table A.3: SCM parameters, origin of the frame is at the MSR's center.

this dissertation.

A.4 The STC

The STC is characterized by the elements shown in Figure A.3. The important features are the radius of the connection box $r_{CB} = .21$m, the radius of the STC $r_{STC} = .105$, the connection box length $\ell_{CB} = .45$m, the length of the STC $\ell_{STC} = 1.272$m, the center of the connection box $x = 2.034$m $y = 0$m and $z = \pm.275$m, the center of the STC $x = 1.996$m $y = 0$m and $z = \pm.275$m, the radius of the annulus $r_{annulus} = .225$m, and center of the annulus $x = 2.632$m $y = 0$m and $z = \pm.275$m. The Currents are elaborated on in Section 3.2, but are 11.75mA for the B_0 Correction and Inner STC, 35mA for the CB Inner Face, 100mA for CB, CB Outer Face, and Middle STC, and lastly 300mA for the Outer STC and Annulus coils.

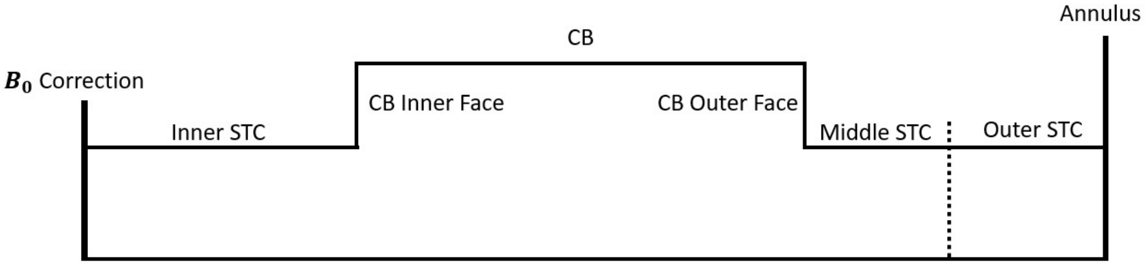


Figure A.3: Names of the Guide Coil components. $\boldsymbol{B}_0$ Correction refers to the correction coil that compensates the $\boldsymbol{B}_0$ coils fringes. In the context of coils, the CB refers to the coil that resides on the lateral area of the cylinder surrounding the CB. The inner and outer faces of the CB refer to the faces of the cylinder closest/furthest to the center of the MSR, respectively. The middle and outer STC coils are divided by layer 1 of the μ-metal with a slight offset pushing further into the MSR.

Chapter B Components of a COMSOL Model

COMSOL models are determined by four components: geometry, physics, mesh, and solver. Geometry is all the domains, boundaries, edges, and points needed to define the problem. Physics refers to the dynamics of the system solved for and all accompanying boundary conditions and quantities that need to be specified. The mesh is a collection of nodes, which are points in space at which degrees of freedom are defined, where the degrees of freedom are determined by the physics being solved for[18]. Lastly, the solver sets the conditions for which the model converges and can be used to iterate over user-defined parameters.

The modeled physical problem and the mesh determine the geometry. The geometry elements fall into three groups: geometry of physical objects (i.e., the MSR, STC, and SCM), geometry for physics and boundary conditions (i.e., surfaces of specified potential or flux, no current regions, and specified current edges), and geometry to define meshing. The first two types of geometry elements are set by the problem being solved. Geometry elements for the mesh are informed by the other geometry elements' size and shape and the problem's precision requirement.

The model physics sets the degrees of freedom being solved for, utilizes material properties, allows for the specification of boundary conditions, and defines innate symmetries of the problem. The degrees of freedom depend on what packages of COMSOL the model utilizes. For the case of the designed STC, the physics studied was the mf (magnetic field) and the PDE (partial differential equation) modules. Material properties are referenced in the physics section of the COMSOL model. These materials can be added manually or sourced from a built-in library. COMSOL has many predefined symmetries and boundary conditions that allow for the problem to be uniquely determined. In the boundary condition, selection allows for currents to be set in the wires and for mirroring symmetry planes to be specified, which are crucial in modeling coils.

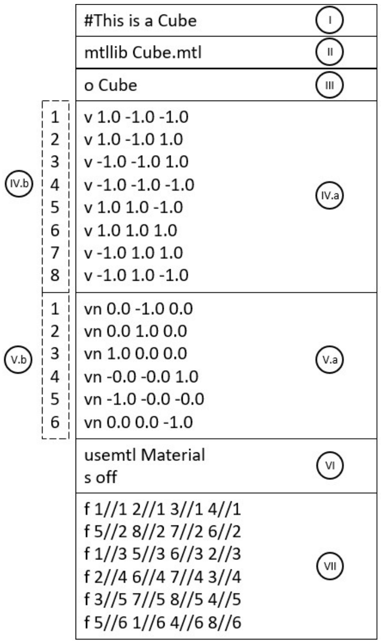
The geometry and physics are largely determined externally, where the mesh is a component that takes a level of creativity. The fineness of the mesh determines how smooth the resulting fields are in the selected regions. If a mesh is too coarse, non-material artifacts will appear in the resulting field maps. There is a balance to be struck because too fine a mesh and the size of the model will run away, becoming too large to run or, worse, cause the machine it is running on to crash. An equilibrium is found in maximizing the quality of the mesh in regions where a smooth field is required and letting areas outside of explicit study exist at a minimum required quality with smooth transitions between the two. Smoothly transitioning between the different qualities in the mesh can be difficult when the scale of objects varies wildly near one another, as is the case with the MSR and portions of the STC. This can be overcome by introducing new geometry elements that isolate transition domains in the mesh quality.

Finally, the solver is the portion of the model that requires the slightest oversight. The solver allows the user to define limits on the convergence of the model; however, the default values result in a quality field map when the mesh is constructed well. The area of the solver that has the most utility is the parametric study functionality. If the model is defined in terms of variables (i.e., a distance from a wall d, a radius r, a current I, a permeability μ, ect...), the parametric study allows for several values to be swept trough at once in one solution. This is a helpful way to model a changing variable's dynamics without running many simulations.

Chapter C OBJ File Structure

For 3D printing and visualization alike, OBJ files are handy. There are many other types of 3D file formats. However, few are as human-readable as the OBJ. Because of the OBJ simple file structure, it is straightforward to read, write, and modify the file contents without extensive effort or other packages/software. The only requirement for external software is in the rendering of the OBJ.

The file structure for a cube's OBJ file is shown in Figure C.1; this serves as a template for all components to be discussed in a 3D object file. The figure shows that all comments are denoted with a pound sign (#). A materials file can be specified using mtllib <*file name*>, where a material file is used to set the colors and texture properties of the OBJ. Material files need not be explored further because the de-



```
#This is a Cube
mtllib Cube.mtl
o Cube
v 1.0 -1.0 -1.0
v 1.0 -1.0 1.0
v -1.0 -1.0 1.0
v -1.0 -1.0 -1.0
v 1.0 1.0 -1.0
v 1.0 1.0 1.0
v -1.0 1.0 1.0
v -1.0 1.0 -1.0
vn 0.0 -1.0 0.0
vn 0.0 1.0 0.0
vn 1.0 0.0 0.0
vn -0.0 -0.0 1.0
vn -1.0 -0.0 -0.0
vn 0.0 0.0 -1.0
usemtl Material
s off
f 1//1 2//1 3//1 4//1
f 5//2 8//2 7//2 6//2
f 1//3 5//3 6//3 2//3
f 2//4 6//4 7//4 3//4
f 3//5 7//5 8//5 4//5
f 5//6 1//6 4//6 8//6
```

I
II
III
IV.b
1
2
3
4
5
6
7
8
IV.a
V.b
1
2
3
4
5
6
V.a
VI
VII

Figure C.1: The figure above shows all used elements of a 3D object's OBJ file. I.) Denotes a comment II.) Material declaration III.) Object's name IV.a) Specifying the vertices IV.b) Vertices index V.a) Specifying the vector normals V.b) Vector normal's index VI.) Using the material and "s" off is turning off smooth shading VII.) Constructing all faces.

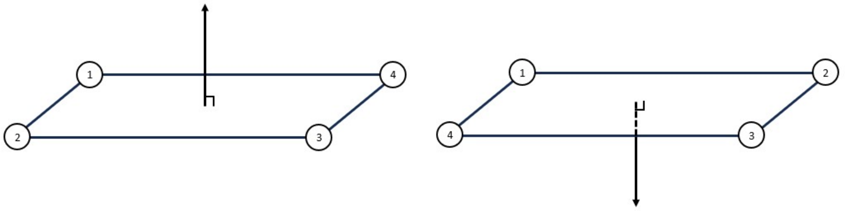


Figure C.2: Shown is how point order determines the vector normal when no normal is provided. The points are considered right-handed to determine the vector's normal direction.

fault files are sufficient for OBJ rendering. Object names are declared using the o symbol, o <*object name*>. Multiple objects can be declared in a single file; however, additional objects must be added at the end of the face declarations (this will be expanded after face construction is discussed). Next, the vertices are specified denoted by a v followed by the $< x, y, z >$ coordinates separated by spaces, v <x> <y> <z>. Vector normals are vectors that are normal to the faces in the object. Vector normals can be excluded, in which case the normal is determined by the order that the vertices appear in the face declaration (See Figure C.2) Similar to vertices, vector normals are declared by vn followed by the vector's components separated by spaces, vn <xn> <yn> <zn>. Vertices and vector normals are indexed (starting at 1) in the order that they appear in the file, and when constructing faces later in the file, they are referenced by their index. Before faces can be defined, the material and smoothing need to be specified. The material is specified by usemtl and is by default referencing Material. Smoothing refers to smooth shading and is not used to render objects of interest. Faces are declared by referencing the indices of vertices and vector normals in the format <*vertex index*>//<*vector normal index*> for all elements that form the face. The total face is then declared by the symbol f followed by the vertex vector normal pairs in the face, f $v_1//vn_1$ $v_2//vn_2$ $v_3//vn_3$ ... where v_i are the vertex indices and vn_i are the vector normal indices. For multiple objects in the same file, the structure is mtllib is declared once, and then all components are present under their respective object; that is to say, for each o <*object name 1*> follows all vertices, vector normals usemtl setting, smoothing setting s and faces in that object. However, the index count for vertices continues with the new object. For example, two cubes would have a total of 16 vertices (8 per cube), and 12 vector normals. The index for the second cube's indices would begin at 9 instead of 1, and the vector normals indices would begin at 7.

There are only a few differences between 3D object OBJs and line object OBJs. Notably, vector normals are not needed (because there are no surfaces/faces), and to specify the line l is used instead of f. To declare a line, the syntax is l v_1 v_2 v_3 ... for each line. Just as multiple faces can make up an object, multiple lines can make up a single object. However, it is more useful to have the wires as separate objects when viewing in a renderer. Creating multiple line objects works the same as specified above, except the faces (f) are replaced with line elements (l), and vector normals are not specified.

Chapter D Additional Code

Many functions were created for the main program of sorting and refining wire paths. This includes basic reading and writing to files and manipulating data in the OBJ file format.

D.1 Input and Output Functions

Described are the contents of the file IO.py, which contains all input and output functions. First is a function that cleans the contents of a folder. When generating the wire CSVs, if one iteration of code identified more wire paths than another iteration, some CSVs would persist from the previous run. This made using the file difficult because the naming scheme adopted for the wire CSVs did not refer to information about the run.

```
import os
#this will delete all CSV files in a folder’s location!!!!!!!
def clean_folder(fpath):
    files=os.listdir(fpath)
    for i in range(len(files)):
        if(files[i].strip().split(’.’)[1]==”csv”):
            os.remove(fpath+files[i])
```

Code D.1: Shown is the code that removes all CSV files in the location of fpath, which is a path to a file.

D.2 OBJ Manipulation Functions

```
#write a group of wires to a CSV
def WriteCSVs(Wires,WireNames,fpath):
    print(”Writing␣CSVs\n”)
    print(f”{len(Wires)}␣total␣Contours”)
    for i in range(len(Wires)):
        print(f”writing␣{fpath}{WireNames[i]}.csv”)
        f=open(fpath+WireNames[i]+”.csv”,’w’)
        for pnt in Wires[i]:
            f.write(f”{pnt[0]},{pnt[1]},{pnt[2]}\n”)
        f.close()
```

```
#reads a CSV with no hedder line
def ReadCSV(fname):
    Points=[] #[[x,y,z] ]
    f=open(fname,’r’)
    Data=f.readlines()
    f.close()
    for i in range(len(Data)):
        x,y,z=Data[i].strip().split(’,’)
        Points.append([float(x),float(y),float(z)])
    return Points
```

```
#this function will write the wire Obj from an imput of a segmented list of points. ie what
    ↪  im calling the Wires in the makeObj function
def WriteWireObj(Wires,WireNames,fpath):
    f=open(fpath,’w’)
    f.write(”mtlib␣Wires.mtl\n”)
    vindex=1
    for w in range(len(Wires)):
        f.write(f”o␣{WireNames[w]}\n”)
        vert_txt=f’’
        line_txt=f’usemtl␣Material\ns␣off\nl␣’
        for p in range(len(Wires[w])):
            vert_txt+=f”v␣{Wires[w][p][0]}␣{Wires[w][p][1]}␣{Wires[w][p][2]}\n”
            line_txt+=f”{vindex+p}␣”
        f.write(vert_txt)
        f.write(line_txt+’\n’)
        vindex+=len(Wires[w])
        print(f”{WireNames[w]}␣Write␣Complete”)
    f.close()
```

```python
#this program will read in the obj for the wire path and will return a list of wires whose
    ↪ contants will be the ordered verticies
def ReadWireObj(fName):
    Verts=[]
    Wires=[]
    VertWires=[]
    f=open(fName,'r')
    Lines=f.readlines()
    for i in range(len(Lines)):
        if(Lines[i].split()[0]=='v'):
            Vert=[]
            for V in Lines[i].split()[1:4]:
                Vert.append(float(V))
            Verts.append(Vert)
        elif(Lines[i].split()[0]=='l'):
            Wire=[]
            for L in Lines[i].split()[1:]:
                Wire.append(int(L)-1)
            Wires.append(Wire)
    #convert the indicies into vertex points in each wire
    for i in range(len(Wires)):
        VertWire=[]
        for j in range(len(Wires[i])):
            VertWire.append(Verts[Wires[i][j]])
        VertWires.append(VertWire)
    return VertWires
```

```
#this reads in a general OBJ
def ReadObj(fName):
    Verts=[]
    SNorms=[]
    Faces=[]
    f=open(fName,’r’)
    Lines=f.readlines()
    for i in range(len(Lines)):
        if(Lines[i].split()[0]==’v’):
            Vert=[]
            for V in Lines[i].split()[1:4]:
                Vert.append(float(V))
            Verts.append(Vert)

        elif(Lines[i].split()[0]==’vn’):
            Norm=[]
            for N in Lines[i].split()[1:4]:
                Norm.append(float(N))
            SNorms.append(Norm)

        elif(Lines[i].split()[0]==’f’):
            Face=[]
            for F in Lines[i].split()[1:]:
                comps=F.split(’//’)
                Face.append([int(comps[0]),int(comps[1])])
            Faces.append(Face)
    return [Verts,SNorms,Faces]
```

```
#ObjComps−>[verticies,normals,faces]
#faces should be in pairs f1=[ [vertex,normal] , [vertex,normal],...] this is for each face
#wFname=file name of written obj.
#this writes a single object
def WriteObj(ObjComps,wFname,mtl_name,obj_name):
    file=open(wFname,’w’)
    verticies=f””
    normals=f””
    faces=f””

    for V in ObjComps[0]:
        verticies+=f”v␣{V[0]}␣{V[1]}␣{V[2]}\n”
    for VN in ObjComps[1]:
        normals+=f”vn␣{VN[0]}␣{VN[1]}␣{VN[2]}\n”
    for Face in ObjComps[2]:
        face=”f␣”
        for fcomp in range(len(Face)):
            face+=f”{Face[fcomp][0]}//{Face[fcomp][1]}␣”
        faces+=f”{face}\n”
    file.write(f”mtlib␣{mtl_name}\no␣{obj_name}\n”)#boiler plate
    file.write(verticies)
    file.write(normals)
    file.write(”usemtl␣Material\ns␣off\n”)
    file.write(faces)
    file.close()
```

```
#ObjList−>[[verticies,normals,faces][]]
#faces should be in pairs f1=[ [vertex,normal] , [vertex,normal],...] this is for each face
#wFname=file name of written obj.
#this will write multiple OBJs to a single file
#Obj_Names has the same length as ObjList
def WriteObjs(ObjList,wFname,mtl_name,Obj_Names):
    file=open(wFname,’w’)
    file.write(f”mtlib␣{mtl_name}\n”)#boiler plate

    V_index_Mod=[0]
    VN_index_Mod=[0]
    for i in range(len(ObjList)−1):
        V_index_Mod.append(V_index_Mod[−1]+len(ObjList[i][0]))
        VN_index_Mod.append(VN_index_Mod[−1]+len(ObjList[i][1]))


    for objs in range(len(ObjList)):
        file.write(f”o␣{Obj_Names[objs]}\n”)
        Faces=””
        Verts=””
        VecNorms=””
        #getting the faces
        for f in range(len(ObjList[objs][2])):
            Face=’f␣’
            for vinf in range(len(ObjList[objs][2][f])):
                v_mod=ObjList[objs][2][f][vinf][0]+V_index_Mod[objs]
                vn_mod=ObjList[objs][2][f][vinf][1]+VN_index_Mod[objs]

                Face+=f”{v_mod}//{vn_mod}␣”

            Faces+=f”{Face}\n”
        #getting the verticies
        for V in ObjList[objs][0]:
            Verts+=f”v␣{V[0]}␣{V[1]}␣{V[2]}\n”

        #getting the normals
        for VN in ObjList[objs][1]:
            VecNorms+=f”vn␣{VN[0]}␣{VN[1]}␣{VN[2]}\n”

        file.write(Verts)
        file.write(VecNorms)
        file.write(”usemtl␣Material\ns␣off\n”)
        file.write(Faces)
```

```
#ObjVerts−>[list of object verticies]
#T−>vector to translate
def TranslateObj(ObjVerts,T):
    ObjTr=[]
    for i in range(len(ObjVerts)):
        ObjTr.append([ObjVerts[i][0]+T[0],ObjVerts[i][1]+T[1] ,ObjVerts[i][2]+T[2] ])

    return ObjTr
```

```
import IO
def MoveCubes(Ts,Obj_Names,fname):
    Cube=IO.ReadObj(”../OBJs/Cubes/Cube.obj”)
    Cubes=[]
    for T in Ts:
        Cubes.append([TranslateObj(Cube[0],T),Cube[1],Cube[2]])
    IO.WriteObjs(Cubes,fname,”Cube.mtl”,Obj_Names)
```

Glossary

BVP Boundary Value Problem. 17

CB Connection Box. viii, ix, 50, 52, 53, 54, 55, 59, 61, 68, 74, 75, 76, 80, 81, 91

CKM Cabibbo–Kobayashi–Maskawa. 6

EDM Electric Dipole Moment. iv, 1, 2, 7

FEA Finite Element Analysis. 22

MSP Magnetic Scalar Potential. iv, vii, 14, 15, 17, 18, 19, 21, 33, 34, 48, 49, 52

MSR Magnetic Shield Room. iv, v, vii, viii, ix, 21, 22, 23, 24, 25, 26, 29, 30, 31, 37, 38, 42, 43, 53, 55, 59, 66, 70, 72, 75, 87, 88, 90, 91, 92, 93

nEDM neutron Electric Dipole Moment. iv, 1, 2, 5, 6, 7, 9, 10, 21

QCD Quantum Chromodynamics. 6, 7

SCM superconducting magnet. v, vii, ix, x, 21, 24, 26, 27, 28, 29, 30, 31, 32, 37, 55, 83, 84, 87, 90, 92

SM Standard Model. 4, 6, 7

STC Spin Transport Coil. vi, vii, viii, 21, 22, 23, 30, 31, 32, 34, 38, 39, 40, 41, 42, 43, 44, 46, 49, 50, 51, 52, 53, 54, 55, 59, 61, 63, 68, 69, 71, 72, 74, 87, 88, 90, 91, 92, 93

UCN Ultra Cold Neutron. 2, 7, 8, 9, 22, 23, 24, 33, 37, 75, 77

Bibliography

[1] C. Abel, S. Afach, N. J. Ayres, C. A. Baker, G. Ban, G. Bison, K. Bodek, V. Bondar, M. Burghoff, E. Chanel, Z. Chowdhuri, P.-J. Chiu, B. Clement, C. B. Crawford, M. Daum, S. Emmenegger, L. Ferraris-Bouchez, M. Fertl, P. Flaux, B. Franke, A. Fratangelo, P. Geltenbort, K. Green, W. C. Griffith, M. van der Grinten, Z. D. Grujić, P. G. Harris, L. Hayen, W. Heil, R. Henneck, V. Hélaine, N. Hild, Z. Hodge, M. Horras, P. Iaydjiev, S. N. Ivanov, M. Kasprzak, Y. Kermaidic, K. Kirch, A. Knecht, P. Knowles, H.-C. Koch, P. A. Koss, S. Komposch, A. Kozela, A. Kraft, J. Krempel, M. Kuźniak, B. Lauss, T. Lefort, Y. Lemière, A. Leredde, P. Mohanmurthy, A. Mtchedlishvili, M. Musgrave, O. Naviliat-Cuncic, D. Pais, F. M. Piegsa, E. Pierre, G. Pignol, C. Plonka-Spehr, P. N. Prashanth, G. Quéméner, M. Rawlik, D. Rebreyend, I. Rienäcker, D. Ries, S. Roccia, G. Rogel, D. Rozpedzik, A. Schnabel, P. Schmidt-Wellenburg, N. Severijns, D. Shiers, R. Tavakoli Dinani, J. A. Thorne, R. Virot, J. Voigt, A. Weis, E. Wursten, G. Wyszynski, J. Zejma, J. Zenner, and G. Zsigmond. Measurement of the permanent electric dipole moment of the neutron. *Phys. Rev. Lett.*, 124:081803, Feb 2020.

[2] E. S. Abers. *Quantum Mechanics.* Pearson Education, 2004.

[3] A. Aleksandrova. *MAGNETIC FIELD MONITORING IN THE SNS NEUTRON EDM EXPERIMENT.* PhD thesis, University of Kentucky, 2019.

[4] I. Altarev, E. Babcock, D. Beck, M. Burghoff, S. Chesnevskaya, T. Chupp, S. Degenkolb, I. Fan, P. Fierlinger, A. Frei, E. Gutsmiedl, S. Knappe-Grüneberg, F. Kuchler, T. Lauer, P. Link, T. Lins, M. Marino, J. McAndrew, B. Niessen, S. Paul, G. Petzoldt, U. Schläpfer, A. Schnabel, S. Sharma, J. Singh, R. Stoepler,

S. Stuiber, M. Sturm, B. Taubenheim, L. Trahms, J. Voigt, and T. Zechlau. A magnetically shielded room with ultra low residual field and gradient. *Review of Scientific Instruments*, 85(7):075106, 07 2014.

[5] N. J. Ayres, G. Ban, L. Bienstman, G. Bison, K. Bodek, V. Bondar, T. Bouillaud, E. Chanel, J. Chen, P.-J. Chiu, B. Clément, C. B. Crawford, M. Daum, B. Dechenaux, C. B. Doorenbos, S. Emmenegger, L. Ferraris-Bouchez, M. Fertl, A. Fratangelo, P. Flaux, D. Goupillière, W. C. Griffith, Z. D. Grujic, P. G. Harris, K. Kirch, P. A. Koss, J. Krempel, B. Lauss, T. Lefort, Y. Lemière, A. Leredde, M. Meier, J. Menu, D. A. Mullins, O. Naviliat-Cuncic, D. Pais, F. M. Piegsa, G. Pignol, G. Quéméner, M. Rawlik, D. Rebreyend, I. Rienäcker, D. Ries, S. Roccia, K. U. Ross, D. Rozpedzik, W. Saenz, P. Schmidt-Wellenburg, A. Schnabel, N. Severijns, B. Shen, T. Stapf, K. Svirina, R. Tavakoli Dinani, S. Touati, J. Thorne, R. Virot, J. Voigt, E. Wursten, N. Yazdandoost, J. Zejma, and G. Zsigmond. The design of the n2edm experiment: nedm collaboration. *The European Physical Journal C*, 81(6), June 2021.

[6] N. J. Ayres, G. Ban, G. Bison, K. Bodek, V. Bondar, T. Bouillaud, B. Clement, E. Chanel, P.-J. Chiu, C. B. Crawford, M. Daum, C. B. Doorenbos, S. Emmenegger, A. Fratangelo, M. Fertl, W. C. Griffith, Z. D. Grujic, P. G. Harris, K. Kirch, J. Krempel, B. Lauss, T. Lefort, O. Naviliat-Cuncic, D. Pais, F. M. Piegsa, G. Pignol, G. Rauscher, D. Rebreyend, I. Rienäcker, D. Ries, S. Roccia, D. Rozpedzik, W. Saenz-Arevalo, P. Schmidt-Wellenburg, A. Schnabel, N. Severijns, B. Shen, M. Staab, K. Svirina, R. T. Dinani, J. Thorne, N. Yazdandoost, J. Zejma, G. Zsigmond, and T. nEDM Collaboration. The very large n2EDM magnetically shielded room with an exceptional performance for fundamental physics measurements. *Review of Scientific Instruments*, 93(9):095105, 09 2022.

[7] J. Bork, H.-D. Hahlbohm, R. Klein, and A. Schnabel. The 8-layered magnetically

shielded room of the ptb: Design and construction. Proc. 12th Int. Conf. on Biomagnetism, (2001).

[8] J. Brewington. *Design of the Highly Uniform Magnetic Field and Spin-Transport Magnetic Field Coils for the Los Alamos National Lab Neutron Electric Dipole Moment Experiment*. PhD thesis, University of Kentucky, 2023.

[9] R. G. Brown. Lie (continuous) groups, Jul 2017.

[10] J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay. Evidence for the 2π decay of the k_2^0 meson. *Phys. Rev. Lett.*, 13:138–140, Jul 1964.

[11] T. Chupp and M. Ramsey-Musolf. Electric dipole moments: A global analysis. *Phys. Rev. C*, 91:035502, Mar 2015.

[12] T. E. Chupp, P. Fierlinger, M. J. Ramsey-Musolf, and J. T. Singh. Electric dipole moments of atoms, molecules, nuclei, and particles. *Reviews of Modern Physics*, 91(1), Jan. 2019.

[13] J. Engel, M. J. Ramsey-Musolf, and U. van Kolck. Electric dipole moments of nucleons, nuclei, and atoms: The standard model and beyond. *Progress in Particle and Nuclear Physics*, 71:21–74, 2013. Fundamental Symmetries in the Era of the LHC.

[14] C. Q. Geng and L. Geng. Some tests on CPT invariance. *Mod. Phys. Lett. A*, 22:651–660, 2007.

[15] H. Goldstein. *Classical Mechanics*. Addison-Wesley, 1980.

[16] D. Griffiths. *Introduction to Elementary Particles: Chapter 4*. WILEY-VCH, 2ed edition, 2008.

[17] D. He. *Investigation of Spin-Independent CP Violation in Neutron and Nuclear Radiative Decays*. PhD thesis, University of Kentucky, 2013.

[18] U. of Cambridge Department of Materials Science Metallurgy. Nodes, elements, degrees of freedom and boundary conditions. https://www.doitpoms.ac.uk/tlplib/fem/node.php#:~:text=A%20node%20is%20simply%20a,type%20of%20analysis%20being%20performed.

[19] M. E. Peskin and D. V. Schroeder. *An Introduction to Quantum Field Theory*. Westview Press, 1995. Reading, USA: Addison-Wesley (1995) 842 p.

[20] M. Pospelov and A. Ritz. Electric dipole moments as probes of new physics. *Annals of Physics*, 318(1):119–169, 2005. Special Issue.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, USA, second edition, 1992.

[22] E. M. Purcell and N. F. Ramsey. On the possibility of electric dipole moments for elementary particles and nuclei. *Phys. Rev.*, 78:807–807, Jun 1950.

[23] J. H. Smith, E. M. Purcell, and N. F. Ramsey. Experimental limit to the electric dipole moment of the neutron. *Phys. Rev.*, 108:120–122, Oct 1957.

[24] A. Tureanu. Cpt and lorentz invariance: Their relation and violation. *J. Phys.: Conf. Ser. 474 012031*, 2013.

[25] G. White. *Electroweak Baryogenesis (Second Edition)*. 2053-2563. IOP Publishing, 2022.

[26] J. Yang, X. Zhang, M. Shi, S. Yuan, L. Zhang, L. Wang, and B. Han. Experimental studies on the performance of magnetic shields under different magnetization conditions. *Journal of Physics D: Applied Physics*, 56(21):215001, apr 2023.

Vita

David Carl Bowles

Place of Birth:

- Brunswick, OH

Education:

- University of Kentucky, Lexington, KY
  Ph.D in Physics, January. 2024
- University of Kentucky, Lexington, KY
  M.A. in Physics, December. 2022
- University of Kentucky, Lexington, KY
  B.S. in Physics & Mathematics, May 2017

Professional Positions:

- University of Kentucky, Demonstrations Assistant,
  Aug. 2017 - Jan. 2020
- University of Kentucky, Research Assistant,
  Apr. 2019 - Feb. 2024

Presentations:

Oct. 2021	American Physics Society Division Nuclear Physics Meeting Division of Nuclear Physics, MIT *Communicated the instrument design for active background magnetic field canceling in a manner accessible to a wide range of experiments.*
Apr. 2021	April American Physics Society Meeting American Physics Society, Online *Presented on 3D printed coil design for an elliptical coil with arbitrary direction magnetic field. The presentation consisted of a succinct aggregation of a year's worth of data analysis, Python Coding, CAD design, and prototyping.*
Apr. 2019	Graduate Student Seminar University of Kentucky Graduate Seminar *An introductory explanation of vector proofs with Einstein summation convention and index notation. Talk focused on a mathematical discipline that is usually self taught.*
Apr. 2015	Physics Seminar Discussing MCM Results University of Kentucky Special Seminar *Explained the model developed for the Mathematical Competition in Modeling. Talk highlighted the mathematical model designed over a 3 day period and expanded on the design and functionality.*
Mar. 2015	Kentucky Mathematical Association of America Meeting Morehead State University *Presented the model developed for the Mathematical Competition in Modeling (MCM).*