Magnetic Mapping for the ^{6}HeCRES Experiment

Jessica Thwaites
Mentor: Prof. Alejandro García
2018 INT REU Program, University of Washington
(Dated: August 23, 2018)

The ultimate goal of the ^{6}He experiment is to use the weak interaction, through beta decay, to study Beyond the Standard Model Physics. By probing beta decay, we hope to investigate potential chirality-flipping interactions, which are not allowed within the Standard Model (SM). By measuring the energy spectrum of the beta decay of ^{6}He extremely accurately using the Cyclotron Radiation Emission Spectroscopy (CRES) technique developed by Project 8, we can look for these interactions that are outside the SM. In order to do so, we need an accurate representation of the magnetic field inside our magnet. This paper deals with the development of a multipole expansion model and adaptation of a NMR probe and measurement device to measure the field in our magnet.

I. MOTIVATION

The Standard Model (SM) of particle physics is the model that describes our current knowledge of the workings of nature at its most fundamental level. However, there are many theories that suggest that the SM is incomplete, and there are missing elements that are Beyond the Standard Model (BSM). The ^{6}HeCRES experiment is an experiment that is looking for chirality-flipping interactions in the beta decay spectrum of ^{6}He. The presence of such interactions would promote the existence of tensor currents, a BSM component present in the general form of the interaction Hamiltonian. This experiment is a precision measurement of the beta decay spectrum of ^{6}He. It relies on a technique for measuring the energy of the emitted electron using Cyclotron Radiation Emission Spectroscopy (CRES), a technique developed by the Project 8 collaboration. [1]

In order to use this technique, the decay volume will be placed inside a solenoidal superconducting magnet. The magnetic field of this magnet must be known very precisely in order to accurately characterize the energy of the beta particles emitted from the decay. This paper focuses on the preliminary work done to develop a magnetic multipole expansion to describe the magnet that will be used in the ^{6}He experiment. It utilizes a Nuclear Magnetic Resonance (NMR) probe to measure the strength of the magnetic field at particular coordinates within the magnet. We also modified a gadget for placing the probe and performed tests to accurately model the position of the probe. This work will allow us to model the magnetic field very accurately in the magnet.

II. THEORY

A. Helicity and Chirality

There are 2 components that are important to understanding the physics we are after. The *helicity* of a particle is the projection of its spin onto its momentum. This can be written as:

$$H = \frac{\vec{p}\cdot\vec{s}}{|\vec{p}||\vec{s}|} = \pm 1 \tag{1}$$

This quantity gives an important value for particles. For massless particles (such as photons), it is invariant. However, for particles with mass, the momentum vector is dependent upon the frame of observation and thus is not invariant. Thus we define another quantity, the *chirality*, which is invariant in all frames. One can express the chirality of a left-handed, spin 1/2 particle in terms of the two possible helicity eigenstates:

$$e^L = \sqrt{\frac{1+\frac{p}{E}}{2}}e^{H=-1} + \sqrt{\frac{1-\frac{p}{E}}{2}}e^{H=+1} \tag{2}$$

With a similar expression (but with coefficients switched) for e^R. In the limit where $m \to 0$, then by the relationship $E^2 = p^2 + m^2$ (using natural units) the coefficient for $e^{H=-1}$ approaches 1 and the coefficient for $e^{H=+1}$ approaches 0, so it can be seen that for massless particles chirality is the same as helicity.

According to the SM, chirality is conserved by the weak interaction. Additionally, in the SM there are only left-handed electrons and neutrinos that participate in beta decay. However, if there are right-handed particles in beta decay, there would be a small cross-term ($e^R e^L$) with chirality not conserved that comes from the interaction Hamiltonian. This small interference term would have the form

$$e^L e^R = \sqrt{1-(\frac{p}{E})^2} = \frac{m}{E} \tag{3}$$

So we expect the interference term to be proportional to m/E.

B. Interaction Hamiltonian

The interaction Hamiltonian allows us to model an interaction of 2 particles. For an interaction of 2 particles

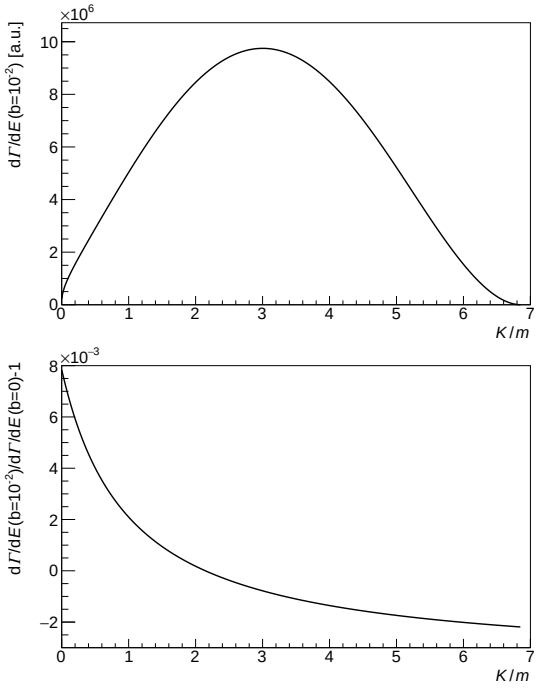


FIG. 1: Top plot: the decay spectrum of ^{6}He from the SM. Bottom plot: Fierz interference term. Both are plotted with respect to energy on the x-axis, and count rate on the y-axis.

(weak or electro-magnetic interaction), the Hamiltonian can be generated by [2]

$$H = j^1_\mu(\frac{1}{q^2+M^2})j^\mu_2 \tag{4}$$

where j_μ are bilinear currents and $\frac{1}{q^2+M^2}$ is called the propagator. q^2 represents the square of the momentum transfer, and M is the mass of the mediator for the interaction. For nuclear beta decay, $M >> q$ so the momentum dependence can usually be neglected.

In its full form, the interaction Hamiltonian for weak interaction can be written as: [3]

$$\begin{aligned} H_{int} = \sum_{i=V,A} (\bar{\psi_p} O^i \psi_n)((C_i + C_i')\bar{\psi_e^L} O_i \psi_\nu^L + \\ (C_i - C_i')\bar{\psi_e^R} O_i \psi_\nu^R) + \\ \sum_{i=S,T} (\bar{\psi_p} O^i \psi_n)((C_i + C_i')\bar{\psi_e^R} O_i \psi_\nu^L + \\ (C_i - C_i')\bar{\psi_e^L} O_i \psi_\nu^R) \end{aligned} \tag{5}$$

The terms are dependent upon the type of bilinear current. In this equation, the V-A terms (first summation) are the vector and axial-vector currents, which are allowed in the SM. The S,T terms are scalar or tensor currents. It is important to note that the S,T terms are chirality-flipping components, and so are not allowed by the SM.

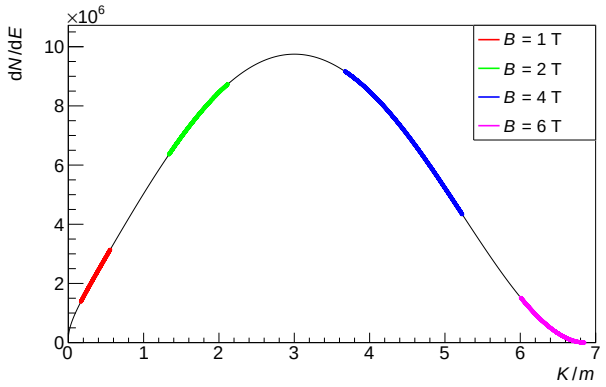


FIG. 2: Plot showing count rate versus energy for the ^{6}He beta spectrum. Colors show our 6 GHz bandwidth with different magnetic field values (red: 1T, green: 2T, blue: 4T, pink: 6T)

C. Fierz Interference

In the Interaction Hamiltonian, there are some terms that contain $\psi_e^R \psi_\nu^L$, which represent both right-and left-handed particles. These can interfere with the SM-allowed term ($\psi_e^L \psi_\nu^L$), yielding a contribution to the decay rate that is forbidden in the SM. This term is called the Fierz interference term. It causes a small distortion on the beta spectrum of ^{6}He proportional to m_e/E_e, which can be seen in Fig 1. [4]

This term can be seen in the decay rate equation:

$$dW \approx dW_0(1 + a\frac{\bar{p_e} \cdot \bar{p_\nu}}{E_e E_\nu} + b\frac{m_e}{E_e}) \tag{6}$$

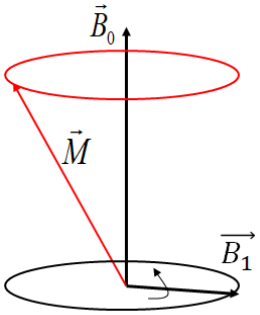
Where b is the parameter that describes the Fierz interference, which is proportional to the m_e/E_e from the chirality flipping term. This term should have a magnitude that can be described by the constants from the Hamiltonian:

$$b_{Fierz} \approx \frac{(C_T + C_T')}{C_A} \tag{7}$$

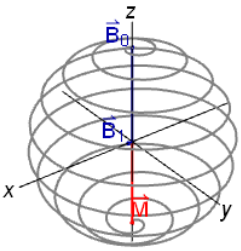
By measuring the beta spectrum very accurately, we would be able to observe this small deviation caused by the chirality-flipping interactions, which is more significant with smaller E. This would demonstrate the existence of tensor currents (C_T and C_T'), which is the BSM physics we are after.

We plan to measure the spectrum of ^{6}He very precisely, using the Cyclotron Radiation Emission Spectroscopy (CRES) method. CRES, which was developed by Project 8, uses a measurement of frequency to extract a value of E_e. As the ^{6}He decays within our magnetic field, the emitted electrons rotate about magnetic field lines, thus emitting cyclotron radiation. The frequency of this radiation takes the form

$$\omega_c = \frac{qBc^2}{E_e} \tag{8}$$



(a) Diagram of rotating B_1 and stationary B_0



(b) Diagram of precession and rotation of a particle's spin as it flips at resonance

FIG. 3: Diagrams describing magnetization and spin flip in NMR spectroscopy

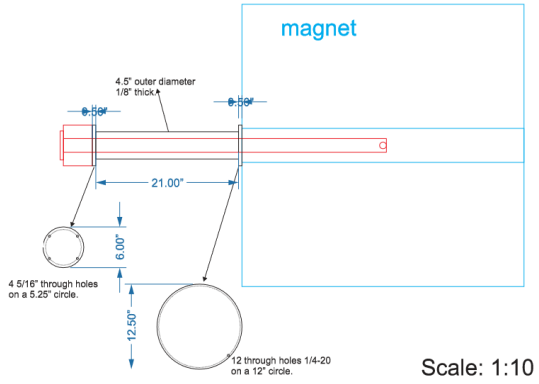


FIG. 4: Modification of our probe. The device in dark blue was added to the probe setup in order to make the device the correct length for our magnet.The light blue outline is our magnet, and the red is the probe device.

The dependence of ω_c on B means that in order to use this technique, we need an accurate determination of the magnetic field at all points in our magnet. In order to do this, we have adapted an NMR probe to take accurate magnetic field magnitude measurements.

When taking data, we will use a bandwidth of 6 GHz (18-24 GHz range) as can be seen in Fig 2. By tuning our magnet to different values, we can record data for the entire beta spectrum.

D. NMR Techniques

Our probe that will be measuring the magnetic field at different regions inside the solenoid operates on Nuclear Magnetic Resonance (NMR) techniques. [4]

NMR spectroscopy is a method of experimentally studying nuclear magnetism. Initially, a sample of particles with individual magnetic moments are placed in a static magnetic field $\vec{B_0}$, which gives the sample a net macroscopic magnetization. An additional oscillating magnetic field, $\vec{B_1}$ is applied, oriented perpendicular to the main field. This field can be pictured as a field rotating at a frequency of ω in the lab frame, as can be seen in Fig 3a. At resonance, this perturbation causes the magnetization of the sample to flip, which gives a signal that can be translated to a magnetic field strength [5].

In the lab frame, the rate of change of the magnetic moment can be modeled with the equation

$$\frac{\vec{M}}{dt} = \gamma \vec{M} \times \vec{B} \tag{9}$$

If we change to a frame that is rotating at the frequency ω, causing B_1 to be along the $\hat{x}$ axis, this can be rewritten as

$$\frac{\partial \vec{M}}{\partial t} = \gamma \vec{M} \times \vec{B_e} \tag{10}$$

with a stationary effective B-field

$$\begin{aligned} \vec{B_e} &= \vec{B} + \frac{\vec{\omega}}{\gamma} \\ &= \hat{k}(B_0 + \frac{\omega}{\gamma}) + \hat{i}(B_1) \end{aligned} \tag{11}$$

This stationary B_e field causes motion that is precession around B_e in the rotating frame. In the lab frame, this motion is precession combined with the rotation rate of the moving frame to the lab frame. This motion traces a path as can be seen in Fig 3b.

III. MAGNETIC MODEL

In order to use the CRES technique, we need to know the magnetic field of our magnet extremely accurately at all points. To do this, we modified an NMR probe that will be used to collect magnetic field values. We then developed a multipole expansion code to model the magnetic field inside our solenoid. We also used a theodolite to perform position measurements to check the accuracy of our dials and map the position of the probe. All of these aspects will be used to determine the magnetic field that our decay volume will be placed in for the experiment.

A. Modifying the Probe Apparatus

We have a probe device, which allows us to place the NMR probe at different locations inside our magnet. Our probe device has the capability to move in 3-dimensions, which generally correspond to cylindrical coordinates, and allows us to set the probe to multiple locations within the region where our decay volume will be. The first is horizontal ($\hat{z}$) motion, controlled by turning a dial on the end of the device. The second and third are two disks,

larger and smaller, which control the angular position of the probe within the magnet ($\hat{\theta}$) and the radius of the probe ($\hat{r}$).

This probe was initially designed for a different, vertical magnet, not our horizontal one. For this reason, we added an extension to the original apparatus, since the probe was initially too long for our magnet. This way, the probe would be centered at $z = 0$ horizontally. This modification can be seen in the diagram in Fig 4.

B. Spherical Harmonic Solutions

For our model we aim to find a solution to Maxwell's equations for our magnet that can be expanded in the form of a magnetic multipole. In the case where there are no enclosed currents in our region of interest, we can write **B** in the form[6]:

$$\mathbf{B} = \nabla\phi_m \tag{12}$$

Which gives Laplace's equation:

$$\nabla^2\phi_m = 0 \tag{13}$$

Solutions for ϕ_m can be written as a Taylor expansion to find all of the terms. These terms are of the form of spherical harmonics (Y_l^m)[7].

$$\phi_m = \sum_{l=0}^{\infty}\sum_{m=-l}^{l}(a_l^m r^l + b_l^m r^{-(l+1)})Y_l^m \tag{14}$$

We transformed these spherical harmonics into Cartesian form, according to the table from [8] (see Appendix A). Because our magnet is primarily a superconducting solenoid, its magnetic field is predominantly in the $\hat{z}$ direction, so these polynomials were used to expand the potential function. This gives 25 linearly independent terms to be fitted by the program.

C. Linear Least Squares

A linear least squares fit was used in order to fit each of the parameters given by the polynomial terms in order to find B_z[9]. Linear least squares utilizes linear algebra to model the relationship between an equation and a set of parameters, such that

$$y_i(x) = \sum_{k=1}^{M} a_k X_k(x) \tag{15}$$

Here, X_i are the functions of position, in this case Cartesian harmonic polynomials, y_i are the values of the magnetic field (B_z) for each point i and a_k are the parameters to be fitted. These X_i are given in [8], and can be explicitly seen in Appendix A: Table I. [9] shows this method for one dimension. However, here we make the substitution for 3 dimensions: $X_k(x) \to X_k(\vec{x})$. This can be proven to be a mathematically valid substitution, as long as the functions X_k are a linearly independent set of functions.

We can then define a function for χ^2, as

$$\chi^2 = \sum_{i=1}^{N}[\frac{y_i - \sum_{k=1}^{M} a_k X_k(\vec{x_i})}{\sigma_i}]^2 \tag{16}$$

where σ_i are the measurement error for the ith data point. When optimized, this equation will give values for the parameters a_k.

We can define a matrix A, called the design matrix, which describes the functions $X_i(\vec{x_i})$ that we want to fit to. This design matrix can be written as:

$$A_{ij} = \frac{X_j(\vec{x_i})}{\sigma_i} \tag{17}$$

In order to optimize the equation for χ^2, one can define two matrices, α and β, such that

$$\alpha_{kj} = \sum_{i=1}^{N}\frac{X_j(\vec{x_i})X_k(\vec{x_i})}{\sigma_i^2} = A^T \cdot A \tag{18}$$

$$\beta_k = \sum_{i=1}^{N}\frac{y_i X_k(\vec{x_i})}{\sigma_i^2} \tag{19}$$

Using a proof from linear algebra[9], we find that the parameters a_k and their uncertainties $\sigma^2(a_j)$ can be found by performing the matrix operations [9]:

$$a_k = [\alpha]^{-1}[\beta] \tag{20}$$

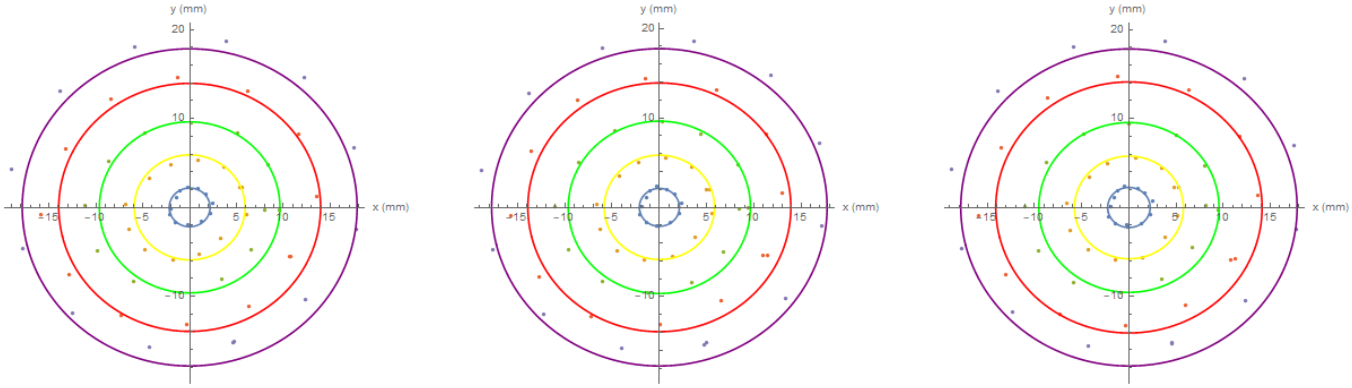
$$\sigma^2(a_j) = [\alpha]^{-1}_{jj} \tag{21}$$

The code to find these parameters for a given set of data points was written as a ROOT macro script. Using ROOT, we are able to clearly define matrices, and do matrix operations easily. The code takes an input of data as a text file with the number of data points taken, the position vector in Cartesian coordinates, and the field value B_z at that particular position. It then prints to the screen values for the coefficient for each of the polynomials a_k. The full code using this method can be seen in Appendix B.

For $[\alpha]^{-1}$, the code uses a Singular Value Decomposition method to ensure that there is no singularity within the matrix α. This ensures that the values for a_k found by the program are the true minimum, and that if one or more values are irrelevant to the fit it is driven to a very small number rather than towards canceling infinities, as often happens if a matrix is singular (or nearly singular) [9].

D. Position Mapping using a Theodolite

To ensure that the values read from the dials are accurate and the position measurements are repeatable, a



(a) $z = -1$ (b) $z = 0$ (c) $z = +1$

FIG. 5: Graphs of average position values for particular r-values. The data and their mean radius are color coded (blue: $\phi_2 = 0$; yellow: $\phi_2 = 30$, green: $\phi_2 = 60$; red: $\phi_2 = 100$; purple: $\phi_2 = 180$)

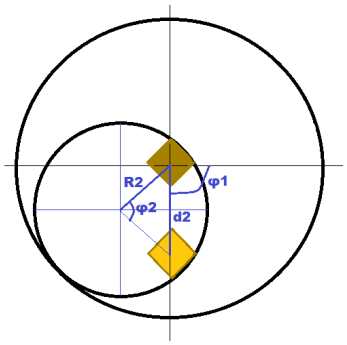


FIG. 6: Diagram of the geometry of the NMR device. The smaller circle allows the user to set the radius of the circle mapped by the device, in this case labeled d_2. The large circle sets the angular position of the probe.

theodolite was used to map the position of the probe. This device allows us to make an accurate (x, y) position measurement, and interpolate from there to map the positions of the probe. A diagram detailing the geometry of the setup can be seen in Fig 6.

If the construction of the apparatus, the relation between the angles and radii would be described by the geometry shown in Fig 6. Then, the (x, y) position coordinates of the probe should be given by:

$$\begin{aligned} x &= d_2 cos(\phi_1) \\ &= 2R_2 sin(\frac{\phi_2}{2}) cos(\phi_1) \qquad (22) \\ y &= d_2 sin(\phi_1) \\ &= 2R_2 sin(\frac{\phi_2}{2}) sin(\phi_1) \qquad (23) \end{aligned}$$

However, due to inaccuracies in the construction we decided to make to tables mapping the coordinates read by the values on the dials to the (x, y, z) coordinates that describe the probe's actual position. The full map that we created for our data sets can be seen in Appendix C: Table III. We took data for a set radius d_2, obtained by fixing ϕ_2 from $0 < \theta < 2\pi$. We did this 3 times for each set value of d_2. We then moved to the next value of d_2. For the first run, we had the probe set at $z = -2.0125$ cm, and went through 7 values of θ from 0 to 2π, incremented evenly in values of 30 degrees for ϕ_2. This was a good troubleshooting run, which gave us an idea of what the data would look like.

The next data sets that we took were for $z = -1, 0, +1$, which is where our decay volume will be, so it is the region of most interest and value to us. For these, we decreased the number of radii that we took data at, by evenly distributing the radii of the trials. We took data at $\phi_2 = 0, 30, 60, 100,$ and180 degrees. The maximum radius for d_2 is 760 mils, so these ϕ_2 values correspond to $d_2 = 0$, $197(\approx d_{2,max}/4)$, $380(d_{2,max}/2)$, $582(\approx d_{2,max}*3/4)$, and 760 mils.

From these data sets, we averaged points at the same ϕ_1 value. We calculated the average radius, maximum radius, and model radius d_2. We then calculated the difference between the mean radius and the maximum radius, and the difference between the mean radius and model radius. These values can be explicitly seen for the $z = 0$ case in Appendix D: Table II. The $r_{mean} - r_{max}$ and $r_{mean} - r_{model}$ provide us with 2 different estimates for error. Minimizing both of these errors allows us to have an accurate position measurement for the data sets.

IV. DISCUSSION

Plots showing the average radius overlaid over the average of 3 trials for each r-value can be seen in Fig 5. These plots show good agreement between the points and the average radii. However, there are some discrepancies that can be seen in these values. The first would be that the blue data points and circle correspond to setting $r = 0$ ($\phi_2 = 0$) on the device. The radius measured from our data when r is set to 0 is about 2 mm. This

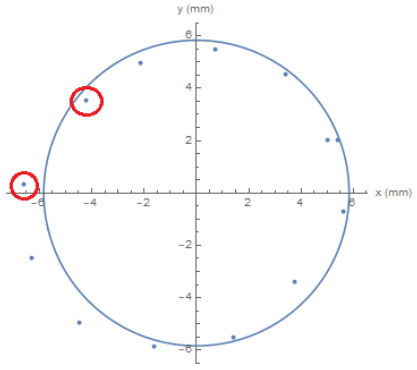


FIG. 7: Data for $z = 0$, $\phi_2 = 30$. Circled in red are the 2 points around the turning point, which could point to a jump in the data, potentially caused by hysteresis in the horizontal dial.

has the largest deviation from expected value of all of our points. However, other values for ϕ_2 were much closer to the expected values. This shows that the values that are in the middle of the radius range are more similar to the model than those at either of the extremes.

Another element that most likely contributed significantly to our error was the hysteresis of the horizontal dial. Because the θ motion of the probe inscribed a full circle, it is impossible to take all the data with motion in a single direction: there must be 2 turning points, where the direction of motion in the horizontal axis must change direction. Prior to beginning the position mapping, I performed tests to determine the amount of hysteresis in this dial. It moved an average of 37 mils without changing the position of the device. In order to mitigate this, whenever I needed to change dial direction I moved it between 200 and 300 mils in the same direction, changed direction and moved it 200-300 mils back.

Ideally, this should have minimized the hysteresis entirely. However, there is a feature in my data that can be seen near this turning point. It is shown in Fig 7 for the $z = 0$, $\phi_2 = 30$ case. Between these 2 circled points I had to change the dial direction. Because these measurements are on a circle, it is impossible to take data for all points without changing the direction of the dial at least once. For most of the trials, $\phi_2 = 0$ was at a turning point, so I was able to use only one turning point rather than 2. But this turning point may have contributed some systematic error in my data. $\phi_2 = 30$ is the most obvious jump in the data sets, and the jumps are less severe as ϕ_2 is larger.

This error is not present in the $\phi_2 = 0$ data set, since there is a separate dial that is more accurate with a range of -100 to +100 mils that could be used for this data set, since its radius was less than 100 mils. Thus even though this data set has the highest observed deviation from expected value from the model, it has the lowest uncertainty from the theodolite, because it uses a more accurate dial. All other data sets had a radius too large to use this more accurate dial measurement.

Even with this error, it is important to note that the main reason for performing these tests is to ensure that our position results are reproducible. Between trials, there was very little deviation (usually ± 20 mils at most), which leads us to believe that despite the small offset the position of the device is still reproducible. If we could remove this error, the value for $r_{max} - r_{mean}$ would be reduced significantly, which would improve our error.

It can also be seen in our graphs for each $z = -1, 0, +1$ case (Fig 5) that the positions of the average points themselves are almost identical between plots. This also points to the reproducibility of our data. The radii found for each value of ϕ_2 (refer to Appendix D Table II) are also almost identical for each value of z. This is what we want, since r and θ should be independent of z. Overall, despite the error that we found, our values agree fairly well with the model values. The data set with $\phi_2 = 60$, with a value of $d_2 = d_{2,max}/2$, had the greatest agreement with the expected values.

V. FUTURE WORK

This project is in its initial stages. [6]HeCRES is an ongoing experiment. For followup with this specific project (mapping the magnet) there is still some work that can be done. It would be beneficial when taking NMR data from the probe to completely automate the data collection process. Currently, all of the dials are still being turned by hand. It would reduce error if this process could be automated.

By the same idea, when data collection takes place, it would be helpful to directly hook the Teslameter, which is the device taking NMR readings from the probe, to a computer with a LabView program running. We attempted this, but were unsuccessful, but this would allow us to fully automate the entire data collection process, and possibly output to the format needed in ROOT more easily.

The multipole model is also still a work in progress. It does take an input of data in 3 dimensions, but only considers aspects of B_z since we expect the field to mostly be that of a solenoid. It would be ideal, however, to also consider the B_x, B_y solutions in the final version. This would allow us to incorporate all 35 linearly independent solutions outlined by [8], rather than the only 25 components that are linearly independent when only B_z is considered.

VI. ACKNOWLEDGEMENTS

This work was supported by the Institute for Nuclear Theory REU program at the University of Washington, and by grant number 1559631 from the National Science Foundation. I would like to thank the REU coordinators, Dr. Gray Rybka and Dr. Subhadeep Gupta, as well as

Cheryl McDaniel and Linda Vilett for their work making this REU possible. I would also like to acknowledge my mentor, Dr. Alejandro García, for all his guidance and help this summer. I would also like to acknowledge the help that I received from graduate students Drew Byron, Madeline Hanley, Yelena Bagdasarova and Thomas Lopez.

Appendix A: Table of Spherical Harmonics

Table I shows the Cartesian spherical harmonics as they appear in the code [8]. The terms have different orders of magnitude determined by Equation 14. The horizontal lines represent different orders of magnitude for The dipole term is r^0, the quadrupole terms are r^1, the sextupole terms are r^2, octupole terms are r^3, and decupole terms are r^4. Note that indexes (n) start at 0 to correspond to the code, since $C++$ indexes always begin at 0.

n	$P_{n,z}(x,y,z)$	name
0	1	dipole
1	$-z$	skew quad
2	x	norm quad
3	y	
4	$-2xz$	skew sext
5	$-2yz$	
6	x^2-z^2	norm sext
7	xy	
8	y^2-z^2	
9	z^3-3x^2z	skew oct
10	$-6xyz$	
11	z^3-3y^2z	norm oct
12	x^3-3xz^2	
13	$3x^2y-3yz^2$	
14	$3xy^2-3xz^2$	
15	y^3-3yz^2	
16	$4xz^3-4x^3z$	skew decu
17	$4yz^3-12x^2yz$	
18	$4xz^3-12xy^2z$	
19	$4yz^3-4y^3z$	
20	$x^4-6x^2z^2+z^4$	norm decu
21	$x^3y-3xyz^2$	
22	$3x^2y^2-3x^2z^2-3y^2z^2+z^4$	
23	xy^3-3xyz^2	
24	$y^4-6y^2z^2+z^4$	

TABLE I: Table of spherical harmonic solutions, considering only terms for B_z, with index number (for ROOT macro) and naming conventions to show orders of magnitude.

Appendix B: Code for Multipole Expansion

Below is the code written to determine the multipole moments that characterize our magnet. This code is written as a ROOT macro, and uses a Linear Least Squares method.The TMatrixD class in ROOT is used to allow us to explicitly declare matrices and use them throughout the program.The design matrix ("void designmatrix'' function) uses the Cartesian spherical harmonics from [8] found in Appendix A: Table I. The code is commented (denoted in C++ by //) to describe individual parts.

```
//Jessie Thwaites
//7/16/18
//Linear least squares fitting program for multipoles
//fits 25 linearly independent fitting parameters
//source for math: Press/Teukolsky/Vettering/Flannery, "Numerical Recipes in
FORTRAN"

#include <iostream>  //libraries
#include <math.h>
#include <fstream>
#include "TMath.h"
#include "TMatrixD.h"
#include "TDecompSVD.h"

using namespace std;  //3 functions
void designmatrix(TMatrixD& matrix, const Double_t (*data)[3],
  const Double_t *unc, int n);
void parameter_unc(TMatrixD& alpha_inv, Double_t *var, int M);
void print_values(TMatrixD& param_val, Double_t *unc, int M);

void linfit(){
  //input of data from a file: "data.txt", as written by linfitdata.C
  int N;
  int M=25;
  ifstream inp;
  inp.open("data.txt");

  Double_t pos[N][3];  //position vectors (x,y,z)
  Double_t unc[N],B[N];  //field and uncertainty values
  if(inp.is_open()){
    inp>>N;  //needs a number of points
    for(int i=0;i<N;i++)
      inp>>pos[i][0]>>pos[i][1]>>pos[i][2]>>B[i]>>unc[i];
  }else{
    cout<<"input file failed to open \n \n";
    return 1;
  }


  //b vector(def to use with matrix operations) for later use with functions
  TMatrixD b(N,1);
  for (int j=0;j<N;j++)
    TMatrixDColumn(b,0)[j]=B[j]/unc[j];

  //design matrix function - sets model form/equations
  TMatrixD Adesmatrix(N,M);
  designmatrix(Adesmatrix, pos, unc, N);

  //transpose of A for alpha,beta fxns
  TMatrixD Atransp(M,N);
  Atransp.Transpose(Adesmatrix);

  //beta matrix, length M
  TMatrixD beta(M,1);
  beta.Mult(Atransp,b);


 //alpha matrix, M by M
 TMatrixD alpha(M,M);
 alpha.Mult(Atransp,Adesmatrix);

 //use SVD to ensure matrix is properly invertible
 TDecompSVD svd(alpha);
 TMatrixD invalpha = svd.Invert();


  TMatrixD a_k(1,M);   //matrix of coefficient values
  a_k.Mult(invalpha,beta);
  Double_t param_unc[M];
  parameter_unc(invalpha, param_unc, M);

  print_values(a_k, param_unc, M);  //print function
```

```
  TMatrixD matrix(N,1);
  matrix.Mult(Adesmatrix,a_k);
  Double_t chisq=0.;
  for(int i=0;i<N;i++)
    chisq+=TMath::Power((TMatrixDColumn(matrix,0)[i]-TMatrixDColumn(b,0)[i]),2);

  cout<< "Chi-sqaured: "<<chisq<<endl<<endl;

  return;
}
void designmatrix(TMatrixD& matrix,const Double_t (*data)[3], const Double_t *unc,
int n){
 //note: in data matrix, data[][0] are x values, data[][1] y, and data[][2] z
 //dipole (l= 0), quadrupole (l= 1)
 for(int i=0;i<n;i++){
   TMatrixDColumn(matrix,0)[i]=1/unc[i];
   TMatrixDColumn(matrix,1)[i]=(-data[i][2])/unc[i];
   TMatrixDColumn(matrix,2)[i]=(data[i][0])/unc[i];
   TMatrixDColumn(matrix,3)[i]=(data[i][1])/unc[i];
 }
 //sextupole functions (l= 2)
 for(int i=0;i<n;i++){
   TMatrixDColumn(matrix,4)[i]=(-2*data[i][0]*data[i][2])/unc[i];
   TMatrixDColumn(matrix,5)[i]=(-2*data[i][2]*data[i][1])/unc[i];
   TMatrixDColumn(matrix,6)[i]=(data[i][0]*data[i][0]-data[i][2]*data[i][2])
     /unc[i];
   TMatrixDColumn(matrix,7)[i]=(data[i][0]*data[i][1])/unc[i];
   TMatrixDColumn(matrix,8)[i]=(data[i][1]*data[i][1]-data[i][2]*data[i][2])
     /unc[i];
 }
  //octupole (l= 3)
  for(int i=0;i<n;i++){
    TMatrixDColumn(matrix,9)[i]=(TMath::Power(data[i][2],3)-3*
      data[i][0]*data[i][0]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,10)[i]=(-6*data[i][0]*data[i][1]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,11)[i]=(TMath::Power(data[i][2],3)-3*data[i][1]*
      data[i][1]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,12)[i]=(TMath::Power(data[i][0],3)-3*data[i][2]*
      data[i][2]*data[i][0])/unc[i];
    TMatrixDColumn(matrix,13)[i]=(3*data[i][0]*data[i][0]*data[i][1]-3*
      data[i][1]*data[i][2]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,14)[i]=(3*data[i][0]*data[i][1]*data[i][1]-3*
      data[i][0]*data[i][2]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,15)[i]=(TMath::Power(data[i][1],3)-3*data[i][1]*
      data[i][2]*data[i][2])/unc[i];
  }
  //decupole (l= 4)
  for(int i=0;i<n;i++){
    TMatrixDColumn(matrix,16)[i]=(4*TMath::Power(data[i][2],3)*data[i][0]-
      4*TMath::Power(data[i][0],3)*data[i][2])/unc[i];
    TMatrixDColumn(matrix,17)[i]=(4*TMath::Power(data[i][2],3)*data[i][1]-
      12*data[i][0]*data[i][0]*data[i][1]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,18)[i]=(4*data[i][0]*TMath::Power(data[i][2],3)-
      12*data[i][0]*data[i][1]*data[i][1]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,19)[i]=(4*TMath::Power(data[i][2],3)*data[i][1]-
      4*TMath::Power(data[i][1],3)*data[i][2])/unc[i];
    TMatrixDColumn(matrix,20)[i]=(TMath::Power(data[i][0],4)-6*data[i][0]*
      data[i][0]*data[i][2]*data[i][2]+TMath::Power(data[i][2],4))/unc[i];
    TMatrixDColumn(matrix,21)[i]=(data[i][1]*TMath::Power(data[i][0],3)-
      3*data[i][0]*data[i][1]*data[i][2]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,22)[i]=(3*data[i][0]*data[i][0]*data[i][1]*data[i][1]-
      3*data[i][0]*data[i][0]*data[i][2]*data[i][2]-3*data[i][1]*data[i][1]*
      data[i][2]*data[i][2]+TMath::Power(data[i][2],4))/unc[i];
    TMatrixDColumn(matrix,23)[i]=(data[i][0]*TMath::Power(data[i][1],3)-
      3*data[i][0]*data[i][1]*data[i][2]*data[i][2])/unc[i];
    TMatrixDColumn(matrix,24)[i]=(TMath::Power(data[i][1],4)-6*data[i][1]*
      data[i][1]*data[i][2]*data[i][2]+TMath::Power(data[i][2],4))/unc[i];

  }
  return;
}

//defines the undertainties for parameters
void parameter_unc(TMatrixD& alpha_inv, Double_t *var, int M){
  for(int i=0;i<M;i++)
    var[i]=sqrt(TMatrixDRow(alpha_inv,i)[i]);
  return;
}

//print function, to show values
void print_values(TMatrixD& param_val, Double_t *unc, int M){
  cout<<"\nBz Parameter Values \n";
  cout<<"--------------------------------"<<endl;

  for(int i=0;i<M;i++)
    Errors[i]=TMath::Abs(TMath::Abs(TMatrixDRow(param_val,0)[i])-TMath::Abs(A[i]));

  //Double_t perc_err[M];
  //for(int i=0;i<M;i++)
  //  perc_err[i]=TMath::Abs(100*unc[i]/TMatrixDRow(param_val,0)[i]);

  cout<<"Avg Dipole: \n";
  cout<<"  p0"<<" = "<<TMatrixDRow(param_val,0)[0]<<" +/- "<<unc[0]<<endl;
  cout<<"Quadrupole: \n";
  for(int i=1; i<4;i++){
    cout<<"  p"<<i<<" = "<<TMatrixDRow(param_val,0)[i]<<" +/- "<<unc[i];
    if(i==1) cout<<"   skew quad \n";
    else if(i==2) cout<<"   norm quad \n";
    else cout<<endl;
  }
  cout<<"Sextupole: \n";
  for(int i=4; i<9;i++){
    cout<<"  p"<<i<<" = "<<TMatrixDRow(param_val,0)[i]<<" +/- "<<unc[i];
    if(i==4) cout<<"   skew sext \n";
    else if(i==6) cout<<"   norm sext \n";
    else cout<<endl;
  }
  cout<<"Octupole: \n";
  for(int i=9; i<16;i++){
    cout<<"  p"<<i<<" = "<<TMatrixDColumn(param_val,0)[i]<<" +/- "<<unc[i];
    if(i==9) cout<<"   skew oct \n";
    else if(i==13) cout<<"   norm oct \n";
    else cout<<endl;
  }
  cout<<"Decupole: \n";
  for(int i=16; i<M;i++){
    cout<<"  p"<<i<<" = "<<TMatrixDColumn(param_val,0)[i]<<" +/- "<<unc[i];
    if(i==16) cout<<"   skew dec \n";
    else if(i==20) cout<<"   norm dec \n";
    else cout<<endl;
  }
  cout<<"--------------------------------"<<endl;
  return;
}
```

Since the code was still at the testing phase, the data read in the file "data.txt" was generated by another macro with a Gaussian distributed error added. When this data was created, an uncertainty of 0.1 was assumed for each data point, and the Gaussian error was within this range given to this program. The uncertainties in parameters are given in the function void parameter_unc, and follow the derivation given in [9]. The final function in the code, void print_values, shows the values that the code found for each of the parameters.

The code prints the fitted parameters, as well as the chi-squared value for the data set. This chi-squared value seems very high, so this part of the code may need more work. The parameters a_k found in the code are entirely arbitrary and were intended to test the accuracy of the fitting function. An example of the output that the code gives when run in ROOT can be seen in Fig 8.

```
root [0] .x linfit2.C

Bz Parameter Values
--------------------------------
Avg Dipole:
  p0 = 1.99962 +/- 0.000511427
Quadrupole:
  p1 = -1.10032 +/- 0.000994423   skew quad
  p2 = 0.339895 +/- 0.000906581   norm quad
  p3 = 0.310111 +/- 0.000865493
Sextupole:
  p4 = 0.720034 +/- 0.000824298   skew sext
  p5 = -0.100499 +/- 0.000828448
  p6 = 0.429432 +/- 0.00147726   norm sext
  p7 = -0.399555 +/- 0.00145589
  p8 = 0.300368 +/- 0.00141209
Octupole:
  p9 = -0.539612 +/- 0.000959421   skew oct
  p10 = 0.30006 +/- 0.000435674
  p11 = -0.0999961 +/- 0.000945347
  p12 = 0.599341 +/- 0.00124343
  p13 = 0.53972 +/- 0.000881845   norm oct
  p14 = 0.500138 +/- 0.000885921
  p15 = -0.599441 +/- 0.00119909
Decupole:
  p16 = -0.390096 +/- 0.00102692   skew dec
  p17 = -0.560032 +/- 0.000437914
  p18 = 0.399936 +/- 0.000411512
  p19 = 0.0400499 +/- 0.00103975
  p20 = -0.320421 +/- 0.00105646   norm dec
  p21 = -0.839845 +/- 0.00382977
  p22 = -0.209545 +/- 0.00163417
  p23 = 0.300287 +/- 0.00390039
  p24 = -0.0101462 +/- 0.000969316
--------------------------------
Chi-sqaured: 68.0351
```

FIG. 8: Output from the code when run in ROOT

Appendix C: ϕ Values and (x, y, z) Coordinates

Because our apparatus may have had some small inconsistencies in its construction, we decided to create a map of the position coordinates (x, y, z) rather than reading the values directly from the dials. These values are reported as distances from the center of the device, and show the readings on the dials, and the positions these values actually correspond to. These values represent an average from 3 trials of data taken for each value of ϕ_2 (refer to geometry given in Fig 6). The full map of values can be seen on the next page, in Table III.

Appendix D: Average values and model comparison

The average values of our data show the deviation from the model that we are finding in our data. This data shows the mean radius (r_{avg}), maximum radius (r_{max}), model predicted radius (r_{model}). These were calculated using Mathematica. Table II explicitly shows the results of these calculations. All r-values are in mm (converted from mils) and all ϕ values are in degrees.

$z = 0$					
ϕ_2	r_{avg}	r_{max}	r_{model}	$r_{max} - r_{avg}$	$r_{avg} - r_{model}$
0	2.117	2.349	0.	0.269	2.117
30	5.832	6.649	4.996	0.922	0.836
60	9.635	10.911	9.652	1.451	0.017
100	13.906	15.498	14.788	1.840	0.881
180	17.738	18.957	19.304	1.522	1.566
$z = +1$					
ϕ_2	r_{avg}	r_{max}	r_{model}	$r_{max} - r_{avg}$	$r_{avg} - r_{model}$
0	2.251	2.531	0.	0.321	2.251
30	5.771	6.598	4.996	0.933	0.775
60	9.542	10.866	9.652	1.498	0.110
100	14.105	15.716	14.788	1.863	0.683
180	17.814	18.986	19.304	1.476	1.490
$z = -1$					
ϕ_2	r_{avg}	r_{max}	r_{model}	$r_{max} - r_{avg}$	$r_{avg} - r_{model}$
0	2.177	2.432	0.	0.293	2.177
30	5.909	6.851	4.996	1.052	0.913
60	9.626	11.022	9.652	1.572	0.026
100	13.959	15.626	14.788	1.917	0.828
180	17.852	19.183	19.304	1.639	1.452

TABLE II: Table of average values for radius traced out by the probe device for $z = 0, +1, -1$, as well as the value for r_{model} which is predicted by the geometry of the device.

[1] D. M. Asner, R. F. Bradley, L. de Viveiros, P. J. Doe, J. L. Fernandes, M. Fertl, E. C. Finn, J. A. Formaggio, D. Furse, A. M. Jones, J. N. Kofron, B. H. LaRoque, M. Leber, E. L. McBride, M. L. Miller, P. Mohanmurthy, B. Monreal, N. S. Oblath, R. G. H. Robertson, L. J. Rosenberg, G. Rybka, D. Rysewyk, M. G. Sternberg, J. R. Tedeschi, T. Thuemmler, B. A. VanDevender, and N. L. Woods (Project 8 Collaboration), "Single-Electron Detection and Spectroscopy via Relativistic Cyclotron Radiation," *Phys. Rev. Lett.*, vol. 114, Apr. 2015.

[2] F. Halzen and A. Martin, *Quarks and Leptons*. John Wiley & Sons, 1984.

[3] R. Hong, M. G. Sternberg, and A. Garcia, "Helicity and nuclear beta decay correlations," *Am. J. Phys.*, vol. 85, no. 45, 2017.

[4] M. Fertl, B. Graner, A. Garcia, M. Guigue, D. Hertzog, P. Kammel, A. Leredde, P. Mueller, N. S. Oblath, R. G. H. Robertson, G. Rybka, G. Savard, D. Stancil, H. E. Swanson, B. A. VanDevender, and A. Young, "Detection of cyclotron radiation applied to searches for chirality-flipping interactions," *Not yet published*, 2018.

[5] A. Abragam, *Principles of Nuclear Magnetism*. Clarendon Press, 1961.

[6] J. D. Jackson, *Classical Electrodynamics*. Hoboken, NJ: John Wiley & Sons, 3 ed., 1999.

[7] C. Boettcher, *Theory of Electric Polarization*. Elsevier Publishing Co., 1952.

[8] M. Fertl, "Trying to understand the expansion of the magnetic field in the g-2 storage ring for the BNL E821 experiment," May 2015.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN*. New York, NY: Cambridge University Press, 2 ed., 1992.

$\phi_2 = 0$				$\phi_2 = 30$				$\phi_2 = 60$				$\phi_2 = 100$				$\phi_2 = 180$			
ϕ_1	z	x	y	ϕ_1	z	x	y	ϕ_1	z	x	y	ϕ_1	z	x	y	ϕ_1	z	x	y
0	0	-0.2761	-2.1343	0	0	5.0136	2.0125	0	0	8.4468	-0.1368	0	0	11.3961	-5.4056	0	0	4.7244	-15.3833
0	1	-0.1495	-2.2590	0	1	4.7755	2.1948	0	1	8.1544	0.2475	0	1	10.8370	-5.8811	0	1	4.5346	-15.1879
0	-1	-0.0033	-2.1447	0	-1	5.3461	2.2144	0	-1	7.9802	-0.3387	0	-1	10.8377	-5.5099	0	-1	4.6511	-15.2530
30	0	1.1674	-1.71943	30	0	5.6147	-0.6969	30	0	6.7789	-4.9628	30	0	6.5193	-10.4856	30	0	-3.9497	-15.7219
30	1	1.3745	-1.8441	30	1	5.1989	-0.9378	30	1	6.6727	-4.4939	30	1	5.9602	-10.9611	30	1	-4.2623	-15.6112
30	-1	1.2244	-1.6451	30	-1	6.0530	-0.8336	30	-1	6.5324	-4.7413	30	-1	6.2699	-11.0978	30	-1	-4.2939	-15.6764
60	0	2.1200	-0.7330	60	0	3.7690	-3.4062	60	0	3.4557	-8.4341	60	0	0.0762	-13.1103	60	0	-12.2089	-11.9119
60	1	2.3228	-0.7942	60	1	3.8823	-3.3932	60	1	3.4596	-8.0499	60	1	-0.3729	-13.3317	60	1	-12.2887	-11.8012
60	-1	2.2065	-0.7350	60	-1	3.3649	-3.5430	60	-1	3.4463	-8.1280	60	-1	-0.2494	-13.1298	60	-1	-12.4515	-11.8664
90	0	2.3316	0.5073	90	0	1.4152	-5.5229	90	0	-1.2264	-9.5348	90	0	-7.2559	-12.2636	90	0	-17.6615	-4.8000
90	1	2.4837	0.6663	90	1	1.4481	-5.5945	90	1	-0.8711	-9.5739	90	1	-7.2647	-12.0617	90	1	-17.7116	-4.6046
90	-1	2.4182	0.5054	90	-1	0.9645	-5.3210	90	-1	-1.0157	-9.5673	90	-1	-7.2471	-12.1138	90	-1	-17.7813	-4.6697
120	0	1.6289	1.4048	120	0	-1.6074	-5.8615	120	0	-6.0312	-8.3494	120	0	-12.7423	-7.7763	120	0	-18.8214	4.0901
120	1	1.7301	1.5468	120	1	-1.3882	-5.9332	120	1	-5.8199	-8.4732	120	1	-13.0178	-7.5744	120	1	-18.8080	4.2854
120	-1	1.7493	1.4918	120	-1	-1.7025	-5.9136	120	-1	-5.9475	-8.2973	120	-1	-12.8732	-7.6265	120	-1	-19.0090	4.3050
150	0	0.8203	2.0187	150	0	-4.4860	-4.9302	150	0	-9.6041	-4.7934	150	0	-15.7141	-1.0030	150	0	-13.8091	12.7261
150	1	0.9216	2.0844	150	1	-4.3346	-4.7479	150	1	-9.6087	-4.8325	150	1	-15.9431	-0.8857	150	1	-13.8423	13.0061
150	-1	0.8773	2.0887	150	-1	-4.7420	-4.7283	150	-1	-9.8337	-4.9107	150	-1	-15.8577	-0.7685	150	-1	-13.8316	13.0256
180	0	-0.2338	2.3362	180	0	-6.2852	-2.4749	180	0	-11.0858	0.1172	180	0	-13.5086	6.2784	180	0	-6.0410	17.8061
180	1	-0.4881	2.4104	180	1	-6.2015	-2.5465	180	1	-11.0396	0.07815	180	1	-13.4835	6.7343	180	1	-5.8879	18.0861
180	-1	-0.2276	2.2453	180	-1	-6.4861	-2.5270	180	-1	-11.1968	0.1693	180	-1	-13.2415	6.5128	180	-1	-5.8221	18.1056
210	0	-0.9450	1.7943	210	0	-6.5773	0.3191	210	0	-8.4865	5.0279	210	0	-8.6741	11.9510	210	0	3.7084	18.5681
210	1	-1.1274	1.9236	210	1	-6.5275	0.4168	210	1	-8.7451	4.9888	210	1	-8.5728	12.3223	210	1	3.8064	18.6788
210	-1	-1.0235	1.8262	210	-1	-6.7825	0.2670	210	-1	-8.5976	5.1647	210	-1	-8.3986	12.1855	210	-1	3.9484	18.6983
240	0	-1.3302	1.1000	240	0	-4.2278	3.5365	240	0	-4.7146	8.5839	240	0	-1.4732	14.4064	240	0	11.7560	14.7581
240	1	-1.5126	1.0473	240	1	-4.1356	3.3802	240	1	-4.8081	8.2908	240	1	-1.2111	14.6929	240	1	12.1715	14.4454
240	-1	-1.3664	1.1108	240	-1	-4.2763	3.3150	240	-1	-4.7495	8.3820	240	-1	-1.2654	14.5562	240	-1	12.2415	14.5496
270	0	-1.7959	0.0374	270	0	-2.1365	4.9758	270	0	0.3103	9.5999	270	0	6.0621	13.0517	270	0	16.8995	7.2227
270	1	-1.9275	0.0863	270	1	-2.3026	4.9042	270	1	0.0391	9.3915	270	1	6.4089	13.2536	270	1	17.5182	6.9101
270	-1	-1.9718	-0.1254	270	-1	-1.9438	4.8390	270	-1	0.2289	9.4827	270	-1	6.2403	13.1168	270	-1	17.5247	6.7603
300	0	-2.0457	-0.8643	300	0	0.7125	5.4838	300	0	4.4040	8.1606	300	0	11.3580	8.1410	300	0	17.1365	-2.0906
300	1	-2.0883	-1.0694	300	1	0.7793	5.5815	300	1	5.0090	8.1215	300	1	11.7218	8.0043	300	1	17.7976	-2.4879
300	-1	-2.1538	-0.8450	300	-1	0.8545	5.3470	300	-1	5.1100	8.2973	300	-1	11.6081	8.2062	300	-1	17.7364	-2.4684
330	0	-1.3387	-1.6813	330	0	3.4007	4.5525	330	0	8.3494	4.7739	330	0	13.0133	1.5370	330	0	13.3054	-10.1339
330	1	-1.3687	-1.7933	330	1	3.5013	4.3962	330	1	8.0485	4.6502	330	1	13.6098	1.4003	330	1	12.4001	-10.4466
330	-1	-1.5103	-1.7467	330	-1	3.7035	4.5850	330	-1	8.4204	4.826	330	-1	13.5343	1.1788	330	-1	12.3219	-10.4270

TABLE III: Tables showing the dial settings (ϕ_1 and ϕ_2) and their respective (x, y, z) coordinates. Note that z is both the z-coordinate and the dial reading.