

# Creating a Negative Ion Matter Interferometer

Eric N. Crook\*

*Department of Physics, Fort Hays State University, Hays, Kansas 67601, USA*

(Dated: May 3, 2016)

This paper describes the negative ion matter interferometer being constructed as part of Dr. Kayvan Aflatooni's research group and my contributions to the experiment. Particularly, my focus was on the effect of the magnetic field used for the trochoidal electron monochromator (TEM) on the beam of negative ions. Based on our experimental and theoretical studies, the experiment should be conducted without the TEM, or preferably, magnetic shielding should be used in the negative ion drift region.

## I. INTRODUCTION

A matter interferometer is a device that combines two coherent matter waves in order to extract information from the resulting interference pattern. A negative ion matter interferometer then is one that uses the matter wave of negative ions as the source. For a chlorine ion with an energy of 10 eV, the deBroglie wavelength is

$$\lambda = \frac{h}{p} \approx 0.001 \text{ nm.} \quad (1)$$

The much smaller wavelength of this source as compared to a HeNe laser, for example, of 632 nm, allows for greater precision in interferometric measurements. In addition, since the source has a mass and net charge, such an interferometer could be used to study phenomena such as gravity and electric and magnetic fields.

Our aim is to create such an interferometer. The following discusses how we plan to do so, and the contributions I have made to the experiment to-date.

### Matter Wave Mach-Zehnder

The core of the interferometer is what is known as the Mach-Zehnder interferometer for matter waves (Fig. 1). The original Mach-Zehnder is a division of amplitude light interferometer (Fig. 2). We adapt this configuration for matter waves by placing two multi-slit diffraction gratings in series with one another. The first grating produces a typical multi-slit diffraction pattern with multiple orders of peak intensity. If we consider only the zeroth and first order peaks, these will be diffracted by the second grating. If the  $m=1$  order of the original  $m=0$  wave and the  $m=0$  order of the original  $m=1$  wave are allowed to recombine, an interference pattern should be observed on the screen. The diffraction gratings therefore perform the function of the beam splitters in the light interferometer.

For our experiment, we start with only using one diffraction grating in order to observe a diffraction pattern, before we add a second grating to perform interference.

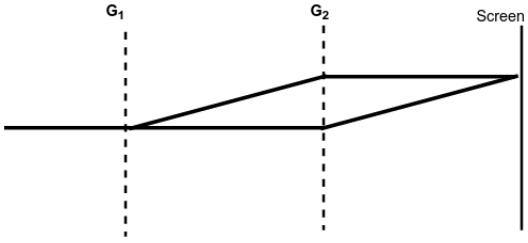


FIG. 1. Schematic of the matter wave Mach-Zehnder interferometer. A matter wave hits the multi-slit grating, G1, where the  $m = 0$  and  $m = 1$  beams interfere with each other after passing through the second grating, G2.

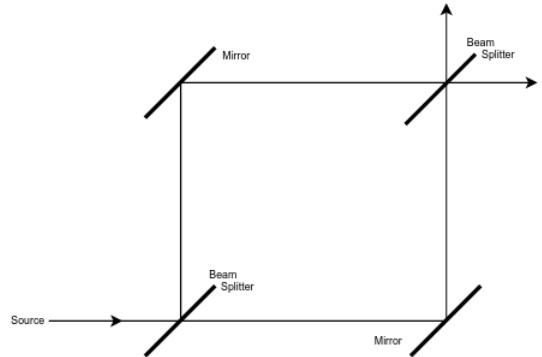


FIG. 2. Schematic of a Mach-Zehnder light interferometer.

## II. APPARATUS

The apparatus that will be used to conduct the negative ion matter interferometry experiment is composed of a vacuum chamber (Fig. 3), where the actual interferometry components are located in the left chamber arm (Fig. 4). There are of course necessary components on the exterior as well. Below is outlined the different components of the experiment and my contributions to them.

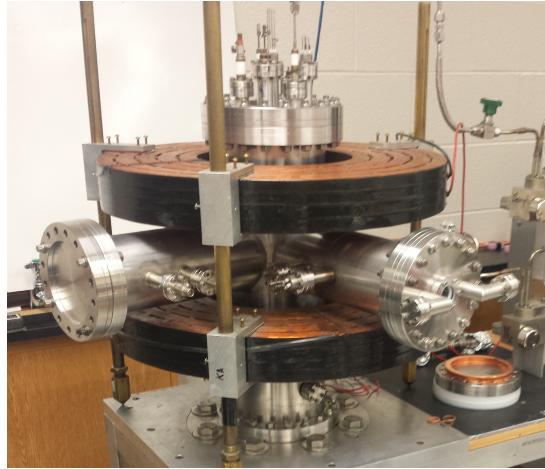


FIG. 3. Image of the apparatus, focusing on the vacuum chamber.

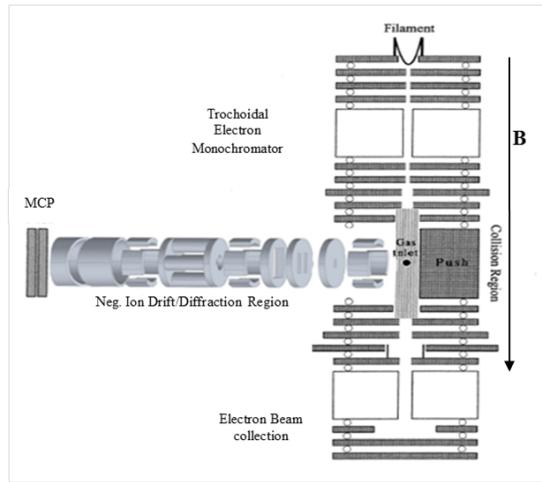


FIG. 4. Schematic of the interior of the apparatus

### Vacuum Environment

To prevent the interference of other particles and for the operation of the electron gun, we need a high-vacuum environment. This is achieved using a mechanical rotary pump, and a diffusion pump, sometimes also referred to as a momentum-transfer pump.

With the assistance of Dr. Aflatooni, I have connected the mechanical pump to the diffusion pump to provide backing for the diffusion pump, as well as roughing the system down to an appropriate pressure for the diffusion pump operation. The connections have been tested on the lower portion of the chamber with minimal loss of pressure once the pump is off. Water lines have also been hooked up to the diffusion pump for cooling and tested. The diffusion pump still needs testing, and the upper portion of the chamber cannot be tested with vacuum until the internal components are added and everything

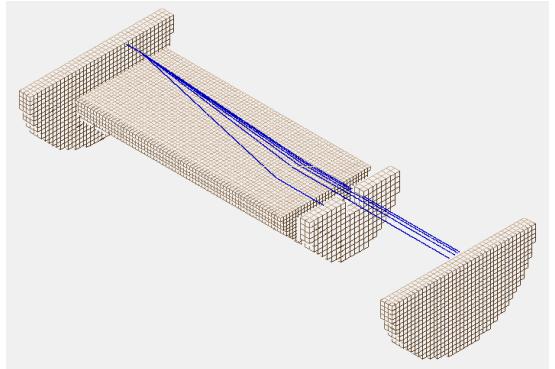


FIG. 5. Cutaway of the bottom half of the trochoidal electron monochromator in SIMION. The two large plates on the ends of the TEM are used to define a uniform axial magnetic field. The trajectory of electrons with various energies is visible in blue.

is properly sealed.

### Trochoidal Electron Monochromator

The filament produces electrons with a large distribution of energy. We use a trochoidal electron monochromator (TEM) to produce a beam of electrons with highly resolved energies. A simulation of this was created using SIMION (Fig. 5). SIMION is an ion optics software that solves Laplace's equation using the relaxation method in order to determine the potential space of a region surrounding a charged conductor. It then uses kinematics to determine the flight of charged particles through the potential space.

In the TEM there is a uniform magnetic field applied along the central axis, and an electric field between the two D-shaped plates perpendicular to the axis. This crossed magnetic and electric field causes the electrons to be bent from their initial axial trajectory. Depending on their energy, the bending will be more or less severe, where the hole at the end of the monochromator will act to filter electrons with a specific energy. Plates with varying voltages further focus and accelerate the electrons after exiting the TEM.

### Producing Negative Ions

Once we have a highly resolved beam of electrons, they are allowed to collide with chlorohydrocarbon molecules to produce  $\text{Cl}^-$  ions.

Chlorohydrocarbons are essentially hydrocarbons, such as methane, where one or more chlorine atoms have replaced hydrogen atoms. These molecules are injected as a gas through a small inlet in the collision region of the apparatus. I designed and connected the current gas

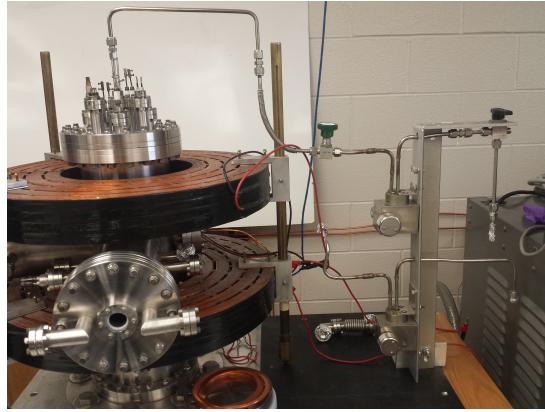
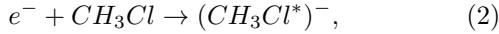


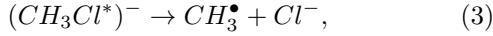
FIG. 6. Image of the gas delivery system used to inject hydrocarbons into the interaction region of the apparatus.

delivery system to be used (Fig. 6).

The actual  $\text{Cl}^-$  ions are produced through a process known as Dissociative Electron Attachment (DEA). In DEA, an electron attaches to a normally empty orbital of a molecule, such as chloromethane,  $\text{CH}_3\text{Cl}$ . An electron with the right wave can occupy the molecule's lowest unoccupied molecular orbital (LUMO) and produce a temporary negative ion [1]:



where the asterisk indicates excitation. There are multiple processes that could occur, but the actual dissociative process is the one that produces  $\text{Cl}^-$ :



where the dot indicates a free radical. The actual molecules we may use are di- or polychloroalkanes, since they have more chlorine atoms and large DEA cross sections [2].

The chlorine ions can then be accelerated toward the drift region by the “pusher”, an electric field, while the remnant of the original molecule has no net charge and is simply pumped out of the chamber.

### Drift Region

The drift region is where the beam of ions are collimated, diffracted and observed.

### Einzel Lens

An einzel lens is used to collimate the beam of chlorine ions. An einzel lens consists of three, hollow cylindrical conductors with different voltages applied to them. This creates a potential space that focuses the beam of ions. Ions further from the center of the lens converge at the

center at a later time than ions already near the center. This is important so that only one ion passes through the diffraction grating at a time, and subsequent ions pass through the same location on the grating.

### Diffraction Grating

As mentioned previously, at this stage in the experiment we will only use one diffraction grating in order to observe a diffraction pattern before we attempt interference with two gratings. The grating used is a 100 nm spacing multi-slit grating produced by MIT (Fig. 7).

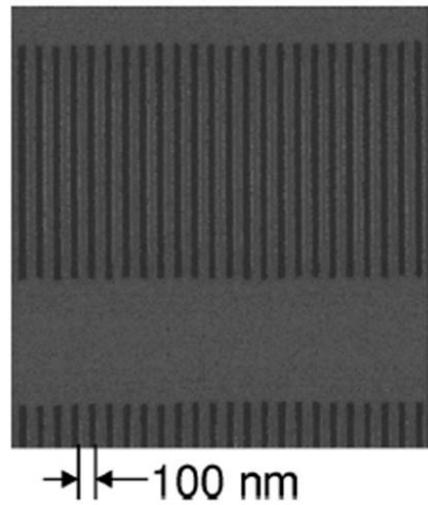


FIG. 7. Microscope view of the diffraction grating to be used in the experiment.

### Quadrupoles

A DC quadrupole is used to expand the diffraction pattern from the grating (Fig. 8). This is necessary since the angle of separation  $\theta$  between the orders  $m$  of the diffraction maxima are very small. The grating equation for a multi-slit diffraction grating with slit spacing  $d$  is

$$d \sin \theta = m\lambda. \quad (4)$$

For a typical  $\text{Cl}^-$  ion (Eq. (1)) with  $m = 1$  and  $d = 100$  nm,

$$\theta = 10 \mu\text{rad}, \quad (5)$$

an angle much too small to observe if not expanded.

The quadrupole used in the experiment was modeled using SIMION (Fig. 8) to demonstrate the expansion effect and to help predict what type of voltages will be needed to be applied to the quadrupole in order to obtain a visible diffraction pattern.

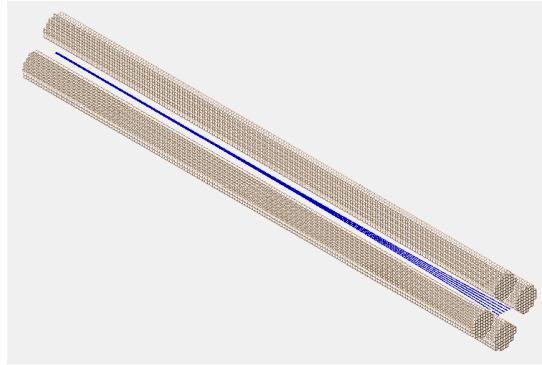


FIG. 8. SIMION model of the actual quadrupoles used in the apparatus. The expansion effect of  $\text{Cl}^-$  ions is visible in blue.

### *Observation*

The diffraction, and eventually, interference pattern will be observed on a phosphorous screen. In the future, we plan to acquire a micro-channel plate (MCP) to record the signal. We expect to observe a pattern similar to Fig. 9. This was taken from a similar experiment performed by Aflatooni, but with electrons rather than  $\text{Cl}^-$  [3].

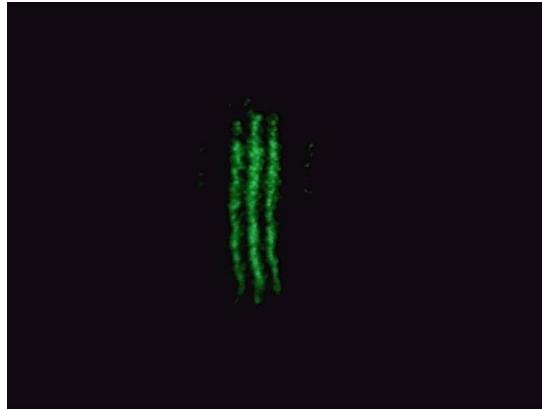


FIG. 9. Diffraction pattern of electrons from a multi-slit grating with the  $m = 0, \pm 1$  orders visible [3].

### III. MAGNETIC FIELD CONCERN

We now turn to an issue encountered with the magnetic field used for the trochoidal electron monochromator. The field is produced by two wide, thick coils of wire (Fig. 10). We expect the field along the central axis of the magnets to be uniform and aligned with the axis, similar to a Helmholtz coil. This is necessary for the operation of the trochoidal electron monochromator. However, the magnetic field in the drift region is unknown. If there is a magnetic field present, it could interfere with the ion

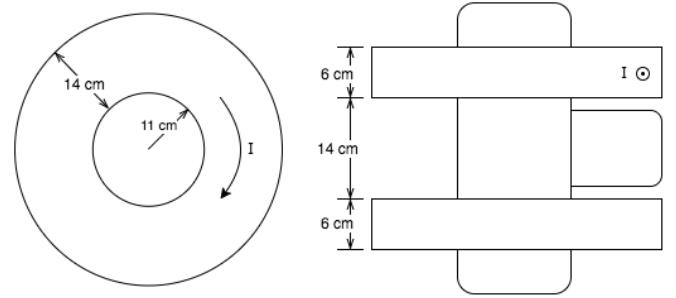


FIG. 10. Schematic of the coils used in the apparatus.

beam trajectory used for the interferometer.

The force exerted by a magnetic field  $\mathbf{B}$  on a particle with charge  $q$  and velocity  $\mathbf{v}$  is given by the Lorentz force law,

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}. \quad (6)$$

If the magnetic field is not in the same direction as the velocity, the force will cause the charged particle to bend. In the interferometer, this bending effect could divert the beam of  $\text{Cl}^-$  away from the diffraction grating and observation screen. It is therefore necessary to know what kind of magnetic field is present in the drift region of the apparatus, if any at all.

### Measuring the Magnetic Field

The magnetic field in the area near the drift region was measured for a current of 1 A in the clockwise direction as viewed from the top. Measurements were taken at points radially outward from the main chamber halfway between the top and bottom coil using a gaussmeter that measures magnetic flux density (Fig. 11). The tip of the probe was positioned in three-directions at each point so as to measure the radial, tangential and vertical components of the magnetic field, allowing magnetic field vectors to be constructed.

### *Results*

The data collected is listed in Table I, and vector plots are given in Figs. 12 and 13. The results are concerning because of the presence of  $B_y$ , or tangential component to the magnetic field (Fig. 13). Due to the azimuthal symmetry of the coils, we do not expect there to be any tangential component. To confirm this, we want to determine the magnetic field of the coils theoretically.

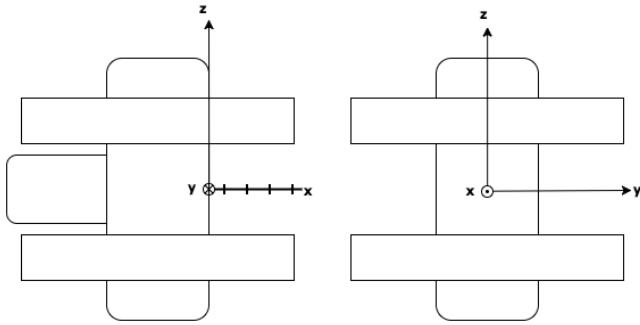


FIG. 11. Coordinate system used to measure the magnetic field. This differs from the coordinate system for the calculated magnetic field, where the origin was shifted to the center of the chamber, rather than the edge.

x (in)	y (in)	z (in)	B <sub>x</sub> (G)	B <sub>y</sub> (G)	B <sub>z</sub> (G)
1	0	0	-20	12	-104
2	0	0	-18	11	-92
3	0	0	-13	10	-74
4	0	0	-10	8	-54
5	0	0	-4	6	-32
6	0	0	-6	4	-16
7	0	0	-3	2	-5
8	0	0	-2	2	11
9	0	0	-2	2	14
10	0	0	-2	2	14

TABLE I. Measurements of the magnetic field of the coils with a 1 A current in the clockwise direction.

### Calculating the Magnetic Field

To calculate the magnetic field due to the coils used in the experiment, a program was written using C++, based on the assumption that the net field produced by the large coils of wire will be the sum of the magnetic fields produced by the individual current loops that constitute the coil. However, the magnetic field of a current loop must first be determined.

#### Magnetic field of a current loop

The magnetic field due to a circular current loop (Fig. 14) can be determined from the magnetic vector potential  $\mathbf{A}$ . The magnetic vector potential is defined such that

$$\mathbf{B} = \nabla \times \mathbf{A}. \quad (7)$$

The magnetic vector potential at a point  $\mathbf{r}$  for a volume  $\Omega$  with current density  $\mathbf{J}(\mathbf{r}')$  is given by [4]

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3 \mathbf{r}'. \quad (8)$$

Due to the azimuthal symmetry of the current loop, the magnetic field at the point  $(r, 0, z)$  will be equivalent to

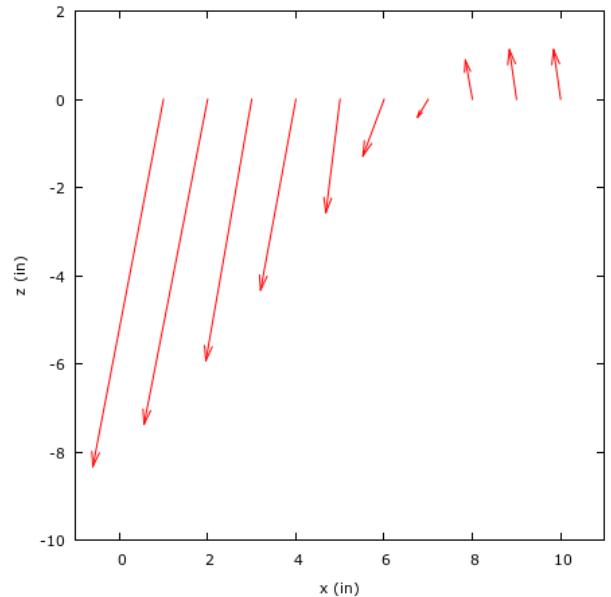


FIG. 12. Measurements for the magnetic field of the coils at points halfway between the coils in the x-z plane. The vectors' magnitude and direction are relative to each other.

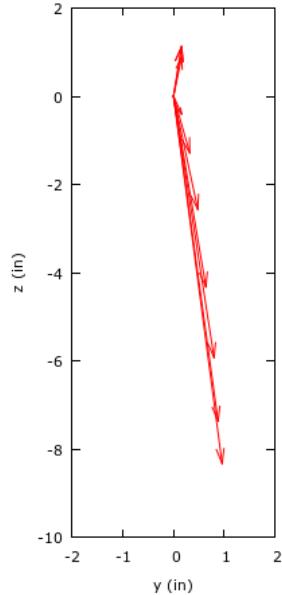


FIG. 13. Measurements for the magnetic field of the coils at points halfway between the coils in the y-z plane. The vectors' magnitude and direction are relative to each other.

the field at a point  $(r, \phi, z)$ . Therefore, the magnetic vector potential of the current loop can be determined as  $\mathbf{A}(r, 0, z)$  with no loss of generality. Defining the Cartesian position vector,  $\mathbf{r}$ , in cylindrical coordinates yields

$$\mathbf{r} = r\hat{\mathbf{x}} + z\hat{\mathbf{z}}, \quad (9)$$

and the position of the source  $\mathbf{r}'$  is given by

$$\mathbf{r}' = a \cos \phi' \hat{\mathbf{x}} + a \sin \phi' \hat{\mathbf{y}}, \quad (10)$$

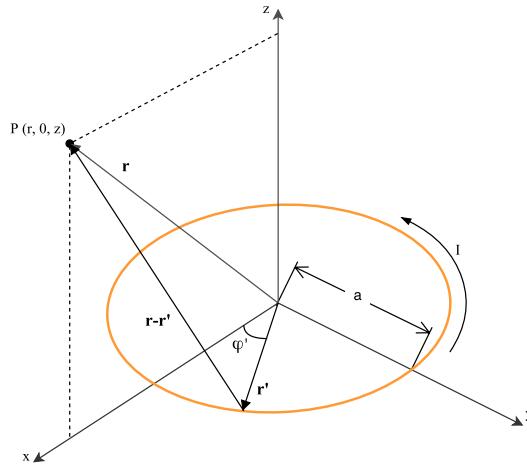


FIG. 14. Circular current loop

where  $a$  is the radius of the loop. The distance between the source and point of observation then becomes

$$|\mathbf{r} - \mathbf{r}'| = \sqrt{(r - a \cos \phi')^2 + (-a \sin \phi')^2 + z^2} \quad (11)$$

$$= \sqrt{z^2 + r^2 + a^2 - 2ra \cos \phi'}. \quad (12)$$

The current is always tangent to the circular loop in the counterclockwise direction viewed from the top:

$$\mathbf{I}(\phi') = I(-\sin \phi' \hat{\mathbf{x}} + \cos \phi' \hat{\mathbf{y}}). \quad (13)$$

Since the current loop is not a volume, but only defined for

$$\begin{cases} r' = a \\ 0 \leq \phi' < 2\pi \\ z' = 0 \end{cases},$$

the current density can be expressed as

$$\mathbf{J}(\mathbf{r}') = \delta(r' - a)\delta(z')\mathbf{I}(\phi'), \quad (14)$$

and the volume element in cylindrical coordinates is

$$d^3\mathbf{r}' = r'd\phi' dr' dz'. \quad (15)$$

The magnetic vector potential becomes

$$\mathbf{A} = \frac{\mu_o I}{4\pi} \iiint \frac{\delta(r' - a)\delta(z')\mathbf{I}(\phi')r'd\phi'dr'dz'}{\sqrt{z^2 + r^2 + r'^2 - 2rr'\cos\phi'}}. \quad (16)$$

Evaluating the integrals over  $r'$  and  $z'$  using the delta functions, and substituting the current  $\mathbf{I}$ ,

$$\mathbf{A} = \frac{\mu_o I a}{4\pi} \int_0^{2\pi} \frac{(-\sin \phi' \hat{\mathbf{x}} + \cos \phi' \hat{\mathbf{y}})d\phi'}{\sqrt{z^2 + r^2 + a^2 - 2ra \cos \phi'}}. \quad (17)$$

To determine the x-component,

$$A_x = \frac{\mu_o I}{4\pi} \int_0^{2\pi} \frac{-a \sin \phi' d\phi'}{\sqrt{z^2 + r^2 + a^2 - 2ra \cos \phi'}}, \quad (18)$$

a substitution is made to simplify the integral:

$$f(\phi') = \sqrt{z^2 + r^2 + a^2 - 2ra \cos \phi'} = \sqrt{u(\phi')} \quad (19)$$

Using the chain rule,

$$\frac{df}{d\phi'} = \frac{df}{du} \frac{du}{d\phi'} \quad (20)$$

$$= \frac{1}{2\sqrt{u}} 2ra \sin \phi' \quad (21)$$

$$= \frac{ra \sin \phi'}{\sqrt{z^2 + r^2 + a^2 - 2ra \cos \phi'}}. \quad (22)$$

This allows the x-component to be written as

$$A_x = -\frac{\mu_o I}{4\pi r} \int_0^{2\pi} \frac{df}{d\phi'} d\phi' \quad (23)$$

$$= -\frac{\mu_o I}{4\pi r} f(\phi') \Big|_0^{2\pi}. \quad (24)$$

Because the function  $f$  is periodic in  $\phi'$  with a period of  $2\pi$ , the x-component is zero:

$$A_x = 0, \quad (25)$$

and the magnetic vector potential has only a y-component.

The y-component can be simplified using a variable substitution:

$$\phi' = \pi - 2\theta, \quad (26)$$

yielding

$$A_y = \frac{\mu_o I a}{4\pi} \int_{\frac{\pi}{2}}^{-\frac{\pi}{2}} \frac{(2 \sin^2 \theta - 1)(-2d\theta)}{\sqrt{z^2 + r^2 + a^2 - 2ra(2 \sin^2 \theta - 1)}}. \quad (27)$$

This substitution is made to put the integral in terms of elliptic integrals. Aside from electromagnetism problems, elliptic integrals arise in applications such as finding the arc length of an ellipse and the motion of a pendulum with large displacements [5]. The complete elliptic integrals of the first and second kind are, respectively,

$$K(k) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \quad (28)$$

$$E(k) = \int_0^{\frac{\pi}{2}} \sqrt{1 - k^2 \sin^2 \theta} d\theta. \quad (29)$$

Simplifying Eq. (27) and switching the limits of integration,

$$A_y = \frac{\mu_o I a}{2\pi} \int_{-\pi/2}^{\pi/2} \frac{(2 \sin^2 \theta - 1)d\theta}{\sqrt{z^2 + (r + a)^2 - 4ra \sin^2 \theta}}, \quad (30)$$

which can be written as

$$A_y = \frac{\mu_o I a}{\pi} \int_0^{\pi/2} \frac{(2 \sin^2 \theta - 1) d\theta}{\sqrt{z^2 + (r+a)^2 - 4ra \sin^2 \theta}}, \quad (31)$$

since the integrand is an even function being integrated over symmetric limits. If we let

$$k^2 = \frac{4ra}{z^2 + (r+a)^2}, \quad (32)$$

Eq. 31 becomes

$$A_y = \frac{\mu_o I k}{2\pi} \sqrt{\frac{a}{r}} \int_0^{\pi/2} \frac{2 \sin^2 \theta - 1}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta. \quad (33)$$

This is further expanded to put it in the form of complete elliptic integrals:

$$A_y = \frac{\mu_o I k}{2\pi} \sqrt{\frac{a}{r}} \left( \int_0^{\frac{\pi}{2}} \frac{2 \sin^2 \theta}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta - \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta \right) \quad (34)$$

$$= \frac{\mu_o I k}{2\pi} \sqrt{\frac{a}{r}} \left( \frac{2}{k^2} \int_0^{\frac{\pi}{2}} \frac{k^2 \sin^2 \theta}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta - \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta \right), \quad (35)$$

where the first integral

$$\begin{aligned} & \int_0^{\frac{\pi}{2}} \frac{k^2 \sin^2 \theta}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta \\ &= \int_0^{\frac{\pi}{2}} \left( \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} - \frac{1 - k^2 \sin^2 \theta}{\sqrt{1 - k^2 \sin^2 \theta}} \right) d\theta \\ &= \int_0^{\frac{\pi}{2}} \left( \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} - \sqrt{1 - k^2 \sin^2 \theta} \right) d\theta, \end{aligned} \quad (36)$$

yielding

$$A_y = \frac{\mu_o I k}{2\pi} \sqrt{\frac{a}{r}} \left[ \frac{2}{k^2} \int_0^{\frac{\pi}{2}} \left( \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} - \sqrt{1 - k^2 \sin^2 \theta} \right) d\theta - \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \right]. \quad (37)$$

The y-component can now be written in terms of complete elliptic integrals (Eqs. 28 and 29):

$$A_y = \frac{\mu_o I k}{2\pi} \sqrt{\frac{a}{r}} \left( \frac{2}{k^2} [K(k) - E(k)] - K(k) \right) \quad (38)$$

$$= \frac{\mu_o I}{\pi k} \sqrt{\frac{a}{r}} \left[ \left( 1 - \frac{k^2}{2} \right) K(k) - E(k) \right]. \quad (39)$$

Since the magnetic vector potential only has a y-component at the point  $(r, 0, z)$ , it is equivalent to the cylindrical tangential component,  $\hat{\phi}$ . However, as mentioned previously, the vector potential at the point  $(r, 0, z)$  is valid for any  $\phi$  due to the azimuthal symmetry of the loop. Therefore,

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_o I}{\pi k} \sqrt{\frac{a}{r}} \left[ \left( 1 - \frac{k^2}{2} \right) K(k) - E(k) \right] \hat{\phi}. \quad (40)$$

The curl of  $\mathbf{A}$  can now be taken in cylindrical coordinates to find the magnetic field:

$$\begin{aligned} \mathbf{B} &= \nabla \times \mathbf{A} \\ &= \left( \frac{1}{r} \frac{\partial A_z}{\partial \phi} - \frac{\partial A_\phi}{\partial z} \right) \hat{r} + \left( \frac{\partial A_r}{\partial z} - \frac{\partial A_z}{\partial r} \right) \hat{\phi} \\ &\quad + \frac{1}{r} \left( \frac{\partial}{\partial r} (r A_\phi) - \frac{\partial A_r}{\partial \phi} \right) \hat{z} \\ &= -\frac{\partial A_\phi}{\partial z} \hat{r} + \frac{1}{r} \frac{\partial}{\partial r} (r A_\phi) \hat{z}. \end{aligned} \quad (41)$$

A computer algebra system was used to calculate  $B_r$  and  $B_z$  by Trebbin [6]:

$$B_r = \frac{\mu_o I}{2\pi \sqrt{z^2 + (r+a)^2}} \left( \frac{a^2 - r^2 - z^2}{z^2 + (r-a)^2} E(k) + K(k) \right) \quad (42)$$

$$B_z = \frac{\mu_o I z}{2\pi r \sqrt{z^2 + (r+a)^2}} \left( \frac{a^2 + r^2 + z^2}{z^2 + (r-a)^2} E(k) - K(k) \right). \quad (43)$$

The magnetic field of a circular current loop is now known.

### Algorithm

Using the magnetic field due to a single current loop, the magnetic field of the coils used in the experiment can be determined using a computer program. As stated previously, the algorithm is based on the superposition principle, but the field is only calculated for a fixed  $\phi$ :

$$\mathbf{B}_{net}(r, z) = \sum_{i=1}^{N_{loops}} \mathbf{B}_i, \quad (44)$$

where  $\mathbf{B}_{net}$  is the magnetic field produced by the coil composed of  $N$  current loops with magnetic field  $\mathbf{B}_i$  at the point  $(r, z)$ .

Starting with the loops in the z-direction, the magnetic field at a point  $(r, z)$  due to the first current loop at a height  $h$  above the x-y plane is equivalent to the magnetic field of the current loop laying on the x-y plane (Fig. 14) at the point  $(r, z-h)$ :

$$\mathbf{B}_1 = \mathbf{B}(r, z-h). \quad (45)$$

Similarly, for a second identical loop a distance  $d$  above the height  $h$ ,

$$\mathbf{B}_2 = \mathbf{B}(r, z - h - d). \quad (46)$$

If there are  $N_z - 1$  identical loops above the first loop, each separated by a distance  $d$ , then the net magnetic field at the point  $(r, z)$  is

$$\mathbf{B}_{net}(r, z) = \sum_{i=0}^{N_z-1} \mathbf{B}(r, z - h - i \cdot d). \quad (47)$$

To further describe the coils used in the experiment, loops with different radii must be considered. If there are  $N_r$  loops sitting in the x-y plane, each with radius  $a_j$ , the net magnetic field at a point  $(r, z)$  is

$$\mathbf{B}_{net}(r, z) = \sum_{j=1}^{N_r} \mathbf{B}(r, z, a_j), \quad (48)$$

where the dependence of a current loop's magnetic field on radius has been added. If the innermost loop of the coil has a radius of  $R_{inner}$ , then the  $j$ -th loop has a radius given by

$$a_j = R_{inner} + j \cdot d, \quad (49)$$

where  $d$  is the separation distance between successive loops. Equation (48) becomes

$$\mathbf{B}_{net}(r, z) = \sum_{j=0}^{N_r-1} \mathbf{B}(r, z, a_j). \quad (50)$$

Combining Eq. (47) and Eq. (50),

$$\mathbf{B}_{net}(r, z) = \sum_{j=0}^{N_r-1} \sum_{i=0}^{N_z-1} \mathbf{B}(r, z - h - i \cdot d, a_j). \quad (51)$$

This models the magnetic field of the top coil used in the experiment. For an identical coil at a height  $h$  below the x-y plane,

$$\mathbf{B}_{net}(r, z) = \sum_{j=0}^{N_r-1} \sum_{i=0}^{N_z-1} \mathbf{B}(r, z + h + i \cdot d, a_j). \quad (52)$$

With these two sums, the magnetic field for the coils used in the experiment can be approximated using a program written in C++.

#### C++ Code

The code written based on the algorithm outlined above is listed in the Appendix. A two-dimensional region of space is defined and divided into finite points and stored in separate  $r$  and  $z$  arrays. Two, two-dimensional

arrays are used to store the  $r$  and  $z$  components of the net magnetic field for every point defined by the  $r$  and  $z$  position arrays. Functions are defined to calculate the  $r$  and  $z$  component of the magnetic field of a single current loop at a point  $(r, z)$  with radius  $a$ , per Eqs. (42) and (43), respectively, which is necessary for the evaluation of Eqs. (51) and (52). These functions use Boost, a peer-reviewed C++ library, to compute the elliptic integrals [7].

The evaluation of the sums in Eqs. (51) and (52) is done by the `compute()` function, and the `output()` function creates a tab-spaced file for plotting the  $r$  and  $z$  components of the calculated magnetic field at every point  $(r, z)$  defined in the specified region.

It should be noted that the `calculate()` function evaluates the field at every point within the defined region, including points within the region occupied by the coils. However, the `output()` functions skips over points that lie within the coil.

#### Results

The results for the calculated magnetic field are displayed by the vector plot in Fig. 15. These values are for a 1 A current running clockwise as viewed from the top of the coils – the same current used when taking the measurements. Figure 16 shows the drift region in more detail.

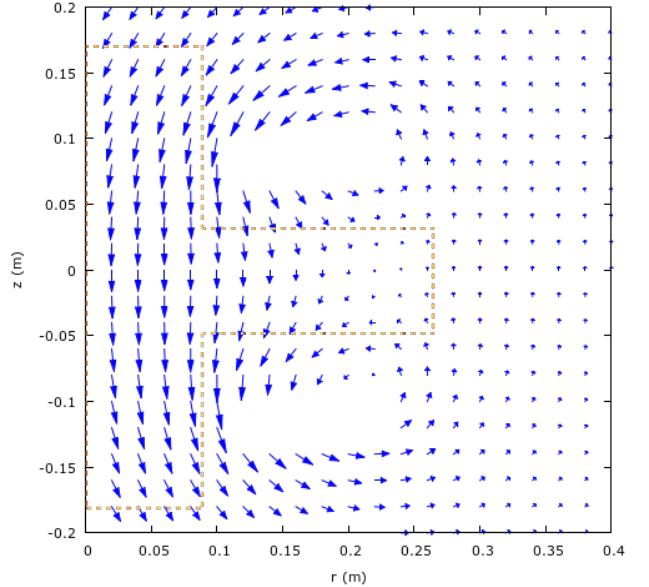


FIG. 15. Calculated magnetic field of the coils for a cross section of the apparatus. The empty space is where the coils are located, and  $r = 0$  corresponds to the central axis of the coils and main chamber.

Comparing the calculated magnetic field to the measured, the first thing to notice is that there are azimuthal

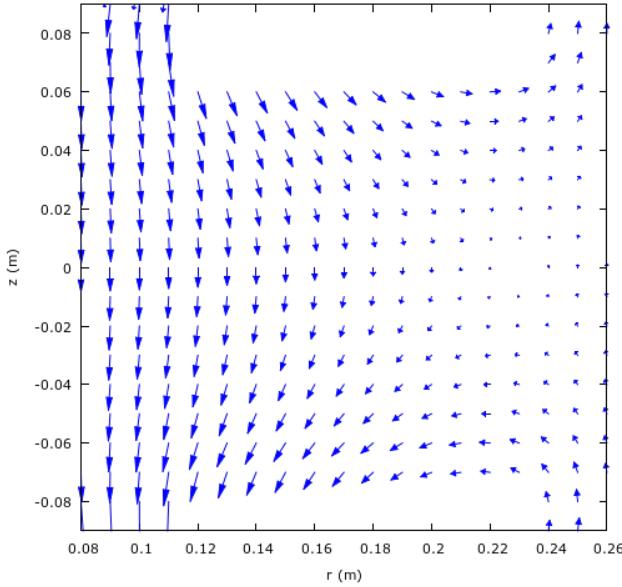


FIG. 16. Closer view of the calculated magnetic field of the coils in an area surrounding the drift region of the apparatus.  $r = 0.08$  corresponds to a distance 8 cm from the central axis of the coils and main chamber.

components to the measured magnetic field. This contradicts what we found from our derivations of a circular current loop and the calculations of the magnetic field. It is interesting, however, to note the similarity between the measurements made along the line  $z = 0$  to the calculated values right below the line  $z = 0$ . This seems to indicate that the measurements were not taken exactly at the halfway point between the coils. In addition, considering that an azimuthal component was measured, it is possible that the coils are not aligned properly, thereby eliminating the symmetry.

#### IV. CONCLUSION

We conclude that there is a magnetic field from the magnetic coils present within the drift region of the apparatus, and it will cause the interferometer's ion beam to be deflected, since it is not in the direction of the ions' velocity.

One option is to simply not use the coils. The magnetic field is only needed for the operation of the TEM to

produce a resolved beam of electrons. However, since our goal with the electrons is to produce as many  $\text{Cl}^-$  ions as possible, it is not necessary to have a narrow energy distribution. We could therefore conduct the experiment without the TEM and coils. However, even without an the coil's magnetic field, Earth's and other external magnetic fields could affect the negative ion interferometer.

Therefore, I propose the drift region of the apparatus be surrounded with a material with high permeability in order to block the magnetic field. Mu-metal is commonly used for this application. It is a soft, metal alloy composed of nickel and iron with a very high permeability [8]. In order to enclose the drift region, a cylinder with a diameter of 8 cm and a length of 25 cm is required. Having such shielding would allow us to have better control over unwanted external magnetic fields in the negative ion matter interferometer.

#### ACKNOWLEDGMENTS

I would like to thank Dr. Kayvan Aflatooni for allowing me to work in his research group, providing me with the opportunity to conduct this project, and his numerous contributions to this project. I have learned a great deal from him as a professor and research advisor. I would also like to thank the Fort Hays Physics department, my classmates, and the professors for their contribution to my education. Finally, thanks to my fiancée, family, and God.

---

\* encrook@mail.fhsu.edu

- [1] K. D. Jordan and P. D. Burrow, *Chemical Reviews* **87**, 557 (1987).
- [2] K. Aflatooni and P. D. Burrow, *The Journal of Chemical Physics* **113**, 1455 (2000).
- [3] K. Aflatooni, Personal communication.
- [4] J. D. Jackson, *Classical Electrodynamics* (Wiley, 1999).
- [5] E. W. Weisstein, "Elliptic integral," (2016), <http://mathworld.wolfram.com/EllipticIntegral.html>.
- [6] G. Trebbin, "Off Axis Magnetic Field of a Circular Current Loop," (2012), <http://www.grant-trebbin.com/2012/04/off-axis-magnetic-field-of-circular.html>.
- [7] B. Schling, *The Boost C++ Libraries* (XML Press, 2011).
- [8] M. S. Corporation, "MuMETAL; Magnetic Shielding ASTM A753," (2016), <http://www.mu-metal.com/>.

```

/* MagneticField.h
 * Author: Eric N. Crook
 * Date: April 24, 2016
 * Description: Interface for the MagneticField class.
 * Used to determine the magnetic field produced by a coil(s) of wires.
*/
#ifndef RESEARCHCODE_HEADER_H
#define RESEARCHCODE_HEADER_H

#include <iostream>
#include <cmath>
#include <fstream>

using namespace std;

class MagneticField {
public:
    MagneticField(double _rMax, int _nr, double _zMin, double _zMax, int _nz);
    //creates the 2D space (since coil of wire is symmetric, no azimuthal component)
    // we want to know mag field over

    ~MagneticField();
    //destructor

    void coils(double _innerR, double _outerR, double _width, double _dist, double _current, double
_gauge, double _separation);
    //defines the coils being modeled

    void calcOnePt(double rr, double zz);

    void calculate();
    //calculate magnetic field

    void outputField(string file_name);
    //outputs magnetic field values into a file that can be plotted

    void outputGeometry(string file_name);
    //outputs values for the geometry of the coils into a file that can be plotted

private:
    //functions:
    double calcBr(double r, double z, double a);
    //input a point (r,z) and radius a of current loop;
    // returns r-comp of mag field at that point
    //units: input--meters, output--Tesla

    double calcBz(double r, double z, double a);
    //input a point (r,z) and radius a of current loop;
    //returns z-comp of mag field at that point

    int rToIndex(double rr); //input a point r (rr) and returns the index it should be assigned to
    int zToIndex(double zz);
    int coordToIndex(double cc, double cMin, double cMax, int nc); //returns the index a point
    //cc should be assigned to based on cMin, etc.

    //variables:
    double rMax; //maximum radial distance away from the origin we want to calculate B
    //for(cylindrical coordinates)
    int nr; //number of r points
    double zMin; //minimum z
    double zMax; //maximum z
    int nz; //number of z points

    double *r; //points to an array that holds the r points
    double *z; //points to an array that holds the z points

    double **Br; //points to a two-dim array that holds the r-comp of mag field at points (r,z)
    double **Bz; // ^

```

```
double innerR; //inner radius of coils
double outerR; //outer radius of coils
double width; //width or thickness of the magnets
double dist; //distance b/w coils
double I; //current
double gauge; //diameter of wires in meters (not actually AWG)
double separation; //distance b/w wires in the coil

const double PI = 3.1459;
const double MU = 4 * PI * pow(10,-7);
};

#endif //RESEARCHCODE_HEADER_H
```

```

/* MagneticField.cpp
 * Author: Eric N. Crook
 * Date: April 24, 2016
 * Description: Implementation of the MagneticField class.
 * Used to determine the magnetic field produced by a coil(s) of wire.
*/
#include "header.h"
#include <boost/math/special_functions/ellint_1.hpp> //boost package for elliptic integrals
#include <boost/math/special_functions/ellint_2.hpp>

MagneticField::MagneticField(double _rMax, int _nr, double _zMin, double _zMax, int _nz) {
    //assign function parameters to private variables
    rMax = _rMax;
    zMin = _zMin;
    zMax = _zMax;
    nr = _nr;
    nz = _nz;

    //create two arrays to hold r and z coordinates/points
    r = new double[nr];
    z = new double[nz];

    //fill the array with the points based on the given number of points
    // we want in the given range
    double dr = rMax / (nr-1); //distance b/w r points
    double dz = (zMax - zMin) / (nz-1);

    for (int i = 0; i < nr; i++)
        r[i] = i * dr;

    for (int j = 0; j < nz; j++)
        z[j] = j * dz + zMin;

    //create the Br and Bz arrays
    Br = new double* [nr];

    for (int i = 0; i < nr; i++) {
        Br[i] = new double[nz];
    }

    Bz = new double* [nr];

    for (int i = 0; i < nr; i++)
        Bz[i] = new double[nz];

    //initialize Br and Bz elements to zero
    for (int i = 0; i < nr; i++) {
        for (int j = 0; j < nz; j++) {
            Br[i][j] = 0.0;
            Bz[i][j] = 0.0;
        }
    }
}

MagneticField::~MagneticField() {
    delete [] r; //destroys (freeing up heap/freestore) array r was pointing to
    r = nullptr; //r points to null pointer

    delete [] z;
    z = nullptr;

    for (int i = 0; i < nr; i++) {
        delete [] Br[i]; //delete the array that the ith element of Br points to
        delete [] Bz[i];
    }

    delete [] Br;
    delete [] Bz;
}

```

```

Br = nullptr;
Bz = nullptr;
}

void MagneticField::coils(double _innerR, double _outerR, double _width, double _dist, double _current,
double _gauge, double _separation) {
    //Define the Coil
    //assign parameters to private members:
    innerR = _innerR;
    outerR = _outerR;
    width = _width;
    dist = _dist;
    I = _current;
    gauge = _gauge;
    separation = _separation;
}

void MagneticField::calcOnePt(double rr, double zz) {
    double d = gauge + separation; //distance b/w the wire loops
    int nlr = int ((outerR - innerR)/d); //number of loops in the coil (radial direction)
    int nlz = int (width/d); //number of loops in the coil's width (z direction)
    double a; //radius loops

    int ri = rToIndex(rr);
    int zi = zToIndex(zz);

    for (int p = 0; p < nlr; p++) { //sum over all the loops in the radial direction
        a = innerR + p * d; //radius of successive loops

        for (int q = 0; q < nlz+1; q++) { //sum over all the loops in the z-direction
            if (a != r[ri]) { //prevents elliptic function from throwing error when k = 1
                //top coil
                Br[ri][zi] += calcBr(rr, zz - dist/2 - q * d, a);
                Bz[ri][zi] += calcBz(rr, zz - dist/2 - q * d, a);
                //bottom coil
                Br[ri][zi] += calcBr(rr, zz + dist/2 + q * d, a);
                Bz[ri][zi] += calcBz(rr, zz + dist/2 + q * d, a);
            }
        }
    }

    cout << Br[ri][zi] << "\t" << Bz[ri][zi] << endl;
}

void MagneticField::calculate() {
    double d = gauge + separation; //distance b/w the wire loops
    int nlr = int ((outerR - innerR)/d); //number of loops in the coil (radial direction)
    int nlz = int (width/d); //number of loops in the coil's width (z direction)

    double h = dist / 2; //distance from origin (halfway between top and bottom coil)
                           // to the first current loop
    double a; //radius of loops
    for (int i = 0; i < nr; i++) {
        for (int j = 0; j < nz; j++) {
            for (int p = 0; p < nlr; p++) { //sum over all the loops in the radial direction
                a = innerR + p * d; //radius of successive loops

                for (int q = 0; q < nlz; q++) { //sum over all the loops in the z-direction
                    if (a != r[i]) { //prevents elliptic func. from throwing error when k = 1
                        //top coil
                        Br[i][j] += calcBr(r[i], z[j] - h - q * d, a);
                        Bz[i][j] += calcBz(r[i], z[j] - h - q * d, a);
                        //bottom coil
                        Br[i][j] += calcBr(r[i], z[j] + h + q * d, a);
                        Bz[i][j] += calcBz(r[i], z[j] + h + q * d, a);
                    }
                }
            }
        }
    }
}

```

```

}

void MagneticField::outputField(string file_name) {
    ofstream out;
    out.open(file_name);

    out << "#r\tx\tyBr\tBz\n";

    for (int i = 0; i < nr; i++) {
        for (int j = 0; j < nz; j++) {
            //print B where the coil is NOT located
            if (!((i > rToIndex(innerR) && i < rToIndex(outerR) && j > zToIndex(dist/2) && j < zToIndex(dist/2 + width))
                || (i > rToIndex(innerR) && i < rToIndex(outerR) && j < zToIndex(-dist/2) && j > zToIndex(-dist/2 - width))) ) {
                out << r[i] << "\t" << z[j] << "\t" << Br[i][j] << "\t" << Bz[i][j] << 0 << endl;
            }
            else
                out << r[i] << "\t" << z[j] << "\t" << 0 << "\t" << 0 << "\t" << endl;
        }
    }
    out.close();
}

void MagneticField::outputGeometry(string file_name) {
    ofstream out;
    out.open(file_name);

    out << "#r\tx\tycolor (1 - coils, 0 - elsewhere)\n";

    //geometry of coil; outputs so gnuplot can color it in
    for (int i = 0; i < nr; i++) {
        for (int j = 0; j < nz; j++) {
            if (!((i > rToIndex(innerR) && i < rToIndex(outerR) && j > zToIndex(dist/2) && j < zToIndex(dist/2 + width)))
                || (i > rToIndex(innerR) && i < rToIndex(outerR) && j < zToIndex(-dist/2) && j > zToIndex(-dist/2 - width))) ) {
                out << r[i] << "\t" << z[j] << "\t" << 0 << endl;
            }
            else
                out << r[i] << "\t" << z[j] << "\t" << 1 << endl;
        }
    }
    out.close();
}

double MagneticField::calcBr(double r, double z, double a) {
    using namespace boost::math;
    double k_2 = (4 * r * a) / (z * z + (a + r) * (a + r)); //k^2
    return (MU * I * z)/(2 * PI * r * sqrt(z*z + (a + r)*(a + r))) * ((a*a + z*z + r*r)/(z*z + (r - a)*(r - a)) * ellint_2(k_2) - ellint_1(k_2));
}

double MagneticField::calcBz(double r, double z, double a) {
    using namespace boost::math;
    double k_2 = (4 * r * a) / (z * z + (a + r) * (a + r)); //k^2
    return (MU * I)/(2 * PI * sqrt(z*z + (a + r)*(a + r))) * ((a*a - z*z - r*r)/(z*z + (r - a)*(r - a)) * ellint_2(k_2) + ellint_1(k_2));
}

int MagneticField::rToIndex(double rr) {
    return coordToIndex(rr, 0, rMax, nr);
}

int MagneticField::zToIndex(double zz) {
    return coordToIndex(zz, zMin, zMax, nz);
}

int MagneticField::coordToIndex(double cc, double cMin, double cMax, int nc) {
    int i = int ((cc - cMin)*(nc - 1)/(cMax - cMin));
    return i; }
```

```
/* main.cpp
 * Author: Eric N. Crook
 * Date: April 24, 2016
 * Description: This application uses the MagneticField class in order to calculate
 * and output the magnetic field produced by the coils currently used in the
 * negative ion matter interferometry experiment.
 */

#include <iostream>
#include "MagneticField.h"

using namespace std;

int main() {
    //Actual lab coil: (don't forget, units are in meters!)
    MagneticField LabMagnets = MagneticField(.4, 21, -0.2, 0.2, 21);
    LabMagnets.coils(0.1113, 0.2483, 0.0635, 0.1349, -1, 0.0020525, 0);
    LabMagnets.calculate();
    LabMagnets.outputField("ResearchCodeMagneticField-ZoomedOut.data");
    LabMagnets.outputGeometry("geometry.data");

    // "zoomed in" view of the drift region
    MagneticField DriftRegion = MagneticField(.4, 41, -0.2, 0.2, 41);
    DriftRegion.coils(0.1113, 0.2483, 0.0635, 0.1349, -1, 0.0020525, 0);
    DriftRegion.calculate();
    DriftRegion.outputField("ResearchCodeMagneticField-Zoomed-In.data");

    //used compare the measurements taken with the gaussmeter to the programs calculations
    DriftRegion.calcOnePt(0.0254, 0);
    DriftRegion.calcOnePt(0.0508, 0);
    DriftRegion.calcOnePt(0.0762, 0);
    DriftRegion.calcOnePt(0.1016, 0);
    DriftRegion.calcOnePt(0.1270, 0);
    DriftRegion.calcOnePt(0.1524, 0);
    DriftRegion.calcOnePt(0.1778, 0);
    DriftRegion.calcOnePt(0.2032, 0);
    DriftRegion.calcOnePt(0.2286, 0);
    DriftRegion.calcOnePt(0.2540, 0);

    return 0;
}
```