Modelling the Magnetic Vector Field of a Magnetostatic Toroidal Solenoid

IB Physics HL – Exploration May 2024

---

1 Introduction

Maxwell's equations are the most beautiful differential equations I know. The elegant simplicity of the time-variant relationship between electric and magnetic fields, the natural and unique emergence of electrostatic superposition, and their unconceivable global significance combine to solidify these equations' awe-inspiring stature. Ever since I encountered them, my interest in studying their applications was clear.

Most commonly, these foundational equations of classical electrodynamics are applied to point charges or dipoles to derive Coulomb's Law or to ideal solenoids to derive Biot-Savart's law. In a Physics class, my teacher suggested I look into toroidal electric currents which instantly sparked a curiosity in me. Toroidal paths, unlike infinite solenoids, are finite closed-loops. They are radially symmetric which I hypothesized was sure to lead to some interesting cancellation or amplification properties. The overabundance of literature studying solenoids was battled by the scarcity of studies into toroids.

Yet, toroids are vital shapes in electromagnetism. Their shape make them a highly efficient inductor commonly used to manufacture transformers. Their closed loop minimizes stray electromagnetic fields making their use in the audio industry to reduce digital distortion and filter signals crucial.

Thus, I decided to model the 3-dimensional vector magnetic field of a toroidal solenoid parameterised by its major and minor radii, and number of turns.

> **Research Question:** How do the number of turns, major and minor radii (m) of a 0 wire-width toroidal solenoid affect its magnetic vector fields when current (A) is kept constant?

2 Background

2.1 Divergence & Curl

If $\boldsymbol{F}$ describes a vector field, $\nabla \cdot \boldsymbol{F}$ describes the *divergence* of the vector field. Positive divergence indicates sources and negative indicates sinks. $\nabla \times \boldsymbol{F}$ describes the curl, amount of rotation, of the vector field. Positive curl indicates clockwise and negative indicates counter-clockwise rotation.

Divergence and curl are the language of Maxwell's equations. Visual interpretations of these operations on a vector field are shown in Figure 1 and Figure 2.

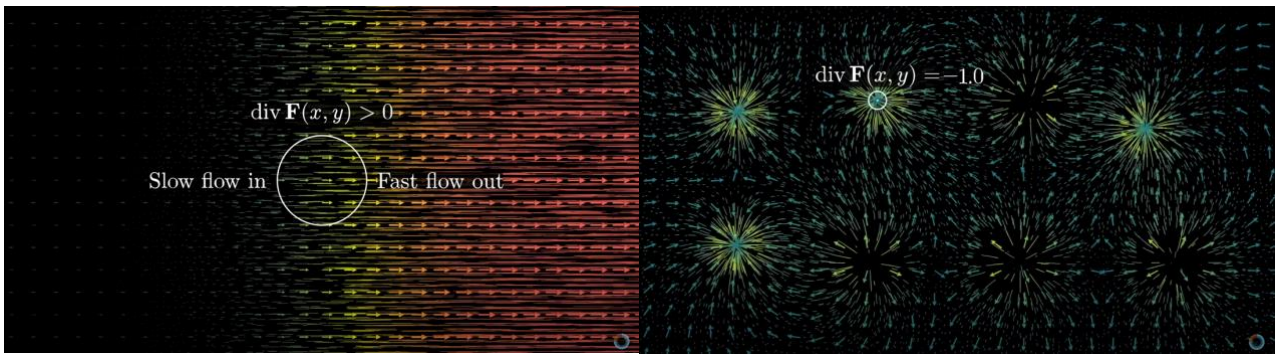


Figure 1 Divergence of a Vector Field[1] (Sanderson, 2015)

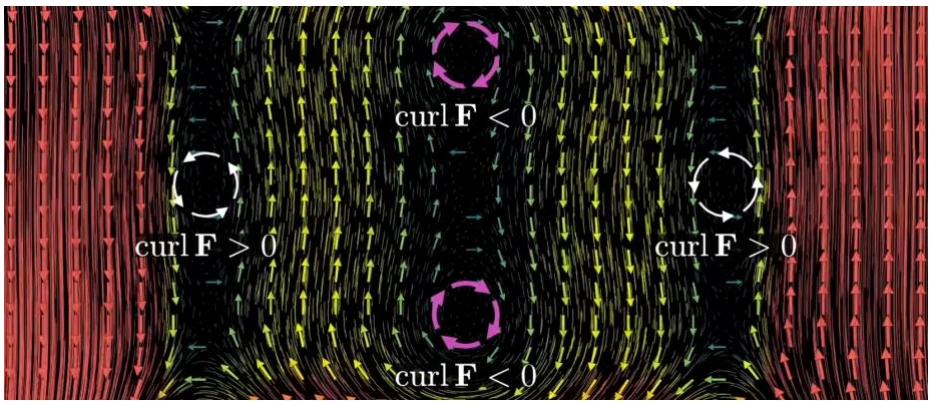


Figure 2 Curl of a Vector Field (Sanderson, 2015)

2.2 Electromagnetic Forces & Fields

> *"In modern physics, we believe that all forces are mediated by fields (not to be confused with "fields" in algebra, or agriculture" (Chua & Tong, 2015)*

In electromagnetism, there exist 2 primary fields: the electrical field $\vec{\boldsymbol{E}}$ and the magnetic field $\vec{\boldsymbol{B}}$. Electric fields are exerted by the presence of electric charge as governed by the first of Maxwell's equations: *Gauss's Law.* The charge density ρ is defined as the charge per unit volume. *Gauss's Law* states that the divergence of the electric field is proportional to the charge density:

$$\nabla \cdot \vec{\boldsymbol{E}} = \frac{\rho}{\varepsilon_0} \tag{1}$$

where ε_0 is the *permittivity of free space* hereby called the *electric constant*. For a positive charge, the electric field acts as a *source* and for a negative charge, as a *sink*. Magnetic fields are created by moving electric current and are proportional in strength to I. Biot-Savart's law calculates the magnetic field at position $\vec{\boldsymbol{x}}$ generated by a current I in a closed loop C:

$$\vec{\boldsymbol{B}} = \frac{\mu_0 I}{4\pi} \oint_C \frac{\overrightarrow{\boldsymbol{d\ell}} \times (\vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}')}{\|\vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}'\|^3} \tag{2}$$

where $\vec{\boldsymbol{x}}'$ is a point along C and $\overrightarrow{\boldsymbol{d\ell}}$ is a differential element of C.

[1] Sanderson, Grant. *Divergence and Curl: The Language of Maxwell's Equations, Fluid Flow, and More*. *YouTube*, 3Blue1Brown, 21 June 2018, https://www.youtube.com/watch?v=rB83DpBJQsE. Accessed 15 May 2023.

3 Formulation & Hypothesis

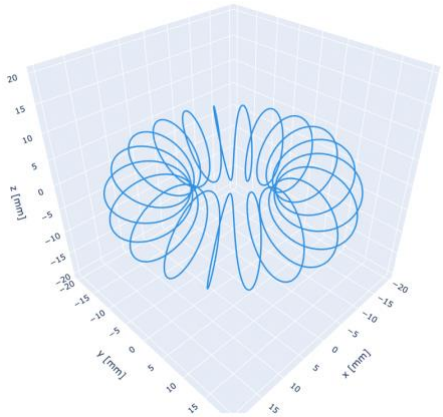


Figure 3 Parameterised toroidal solenoid with 20 turns

A toroidal solenoid with n turns, minor radius a, and major radius $a+b$ can be modelled using a 3-dimensional parametric equation for $t \in [0, 2\pi]$:

$$C_x(t) = [a + b\cos(nt)]\cos(t) \quad (3)$$

$$C_y(t) = [a + b\cos(nt)]\sin(t) \quad (4)$$

$$C_z(t) = b\sin(nt) \quad (5)$$

To limit the complexity of this study, we will consider only the magnetic field generated along the plane at $z = 0$ and the vertical line at the origin. That is, where either $z = 0$ or $x = y = 0$.

3.1 Variables

Independent: There are 4 independent variables that are used to model this exploration:

1. Number of turns in solenoid ($n \in \mathbb{Z}^+$)
2. Minor radius (in meters) of solenoid ($a > 0$)
3. Major radius (in meters) of solenoid ($a + b,\ b > 0$)
4. 3-dimensional position vector (in meters) ($\vec{\boldsymbol{x}} \in \mathbb{R}^3$)

Dependent: We wish to approximate the magnetic field located at $\vec{\boldsymbol{x}}$:

1. 3-dimensional magnetic field (in Teslas) approximation ($\vec{\boldsymbol{B}}'$: $\mathbb{R}^3 \to \mathbb{R}^3$)

Controlled:

1. **Direct current magnitude and direction:** The generated magnetic fields are directly proportional to the magnitude of the current through the solenoid and are directed towards a direction determined by the direction of the current. Therefore, the current and its direction will be controlled in this study. Time-variant currents that produce time-varying fields are not considered.

2. **Zero wire-width:** A non-zero wire width impacts the 3 model of the parameterised solenoid and makes calculating the generated field much more difficult. Thus, a theoretical zero-width solenoid will be used.

3. **Zero resistance:** Resistive effects impact the current flow through the solenoid and the fields produced. Therefore, we also assume a controlled resistance of 0.

3.2 Hypothesis

My hypothesis is that as the number of turns increases, the magnetic field will converge to a smooth and more symmetric distribution across the plane. This is because a solenoid with more turns better approximates an ideal toroid which is radially symmetric. This will also reduce the effect of local curls in the electromagnetic field created due to the spacing between wires.

I also hypothesise that the x- and y-components of the magnetic fields produced outside the rings of the toroid will approach zero as the number of turns increases the difference between the major and minor radii (b) decreases, and the minor radius a decreases. Similar to how this field cancels itself out when flowing in a ring, because it is produced in a direction that directly oppose itself, I expect a similar result in this study. As the difference between the major and minor radii decreases, the solenoid will more closely resemble a ring of current which will intensify the cancelling of the field. As the minor radius decreases, the field should cancel itself out at smaller distances too.

Pertaining to the z-component of the produced field – depending on the direction of current – the magnetic field should concentrate in intensity at the centre of the toroid and flow vertically outwards. Gauss's Law for Magnetism states that the divergence of the magnetic fields should be 0. Since there are no magnetic poles considered, the outwards flowing magnetic fields must join themselves on the other side. The magnetic fields should be most intense in the centre of the toroid, similar to conventional cylindrical models. This is because it converges at the centre and is thus amplified.

I also hypothesize that the general strength of the magnetic field produced will be proportional to the number of turns, similar to how it operates in cylindrical solenoids.

4 Methodology

4.1 Magnetic Field along Plane

Let us begin by constructing Biot-Savart's integral given in (2). The vector difference $\vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}'$ can be found component-wise and its magnitude can be found using 3-dimensional Euclidean distance:

$$\vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}' = \begin{bmatrix} x_x - x'_x \\ x_y - x'_y \\ x_z - x'_z \end{bmatrix} \qquad \|\vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}'\| = \sqrt{(x_x - x'_x)^2 + \left(x_y - x'_y\right)^2 + (x_z - x'_z)^2} \tag{6}$$

$\vec{\boldsymbol{x}}'$ is defined by equations the parameterise the solenoid (3), (4), and (5). $\overrightarrow{\boldsymbol{d\ell}}$ can be defined as the derivatives of these equations:

$$d\ell_x(t) = \frac{d}{dt} C_x = -[a + b\cos(nt)]\sin(t) - bn\cos(t)\sin(nt) \tag{7}$$

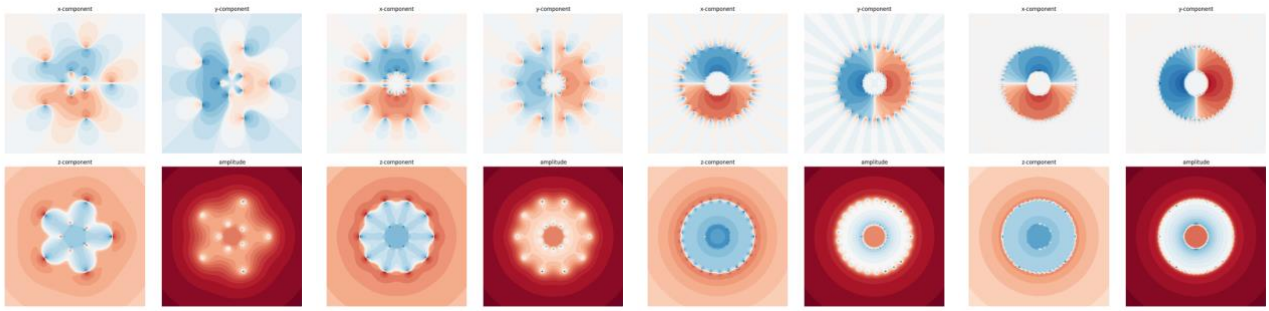
$$d\ell_y(t) = \frac{d}{dt} C_y = [a + b\cos(nt)]\cos(t) - bn\sin(t)\sin(nt) \tag{8}$$

$$d\ell_z(t) = \frac{d}{dt} C_z = bn\cos(nt) \tag{9}$$

Thus, the magnetic field at position $\vec{x}$ with a current I can be found as:

$$\vec{B} = \frac{\mu_0 I}{4\pi} \int_0^{2\pi} \frac{\begin{bmatrix} -[a + b\cos(nt)]\sin(t) - bn\cos(t)\sin(nt) \\ [a + b\cos(nt)]\cos(t) - bn\sin(t)\sin(nt) \\ bn\cos(nt) \end{bmatrix} \times \begin{bmatrix} r_x - [a + b\cos(nt)]\cos(t) \\ r_y - [a + b\cos(nt)]\sin(t) \\ r_z - b\sin(nt) \end{bmatrix}}{\left((r_x - [a + b\cos(nt)]\cos(t))^2 + \left(r_y - [a + b\cos(nt)]\sin(t)\right)^2 + (r_z - b\sin(nt))^2\right)^{3/2}} dt \quad (10)$$

Finding the anti-derivative is beyond the scope of this study. Whether it is even possible is unknown. Instead, we will computationally evaluate the integral component-wise. Figure 4 shows the results of the numerical integration, created using the Python program given in Appendix 8.2.



$n = 5$ $n = 10$ $n = 25$ $n = 50$

Figure 4 Numerical Integration of Magnetic Field along $z = 0$

4.2 Magnetic Field along Toroidal Axis

To find the magnetic field generated along the line at $x = y = 0$, i.e., the toroidal axis, we will exploit the radial symmetry of the solenoid. The reference vector where the magnetic field is being calculated $\vec{x}$ is given by:

$$\vec{x} = z\hat{e}_z \quad (11)$$

where $\hat{e}_z$ is the basis vector pointing in the z-direction and z is the magnitude of $\vec{x}$. A point on the solenoid $\vec{x}'$ can be represented in cylindrical coordinates using its radius $R(\phi)$ and z-position $C_z(\phi)$ which are functions of the angle ϕ:

$$\vec{x}' = R(\phi)\hat{e}'_\rho + C_z(\phi)\hat{e}_z \quad (12)$$

where $\hat{e}'_\rho$ is the basis vector pointing in the radial direction at $\vec{x}'$. Note that in cylindrical coordinates, the basis vectors $\hat{e}_\rho$ and $\hat{e}_\phi$ are themselves functions of the angle ϕ. Hence, we use the symbol $\hat{e}'_\rho$ instead of $\hat{e}_\rho$ to denote that the basis vector lies at some point on the solenoid. The radius function can be calculated using our parameterisation of the solenoid:

$$R(\phi) = \sqrt{C_x(\phi)^2 + C_y(\phi)^2} = \sqrt{a + b\cos(n\phi)} \quad (13)$$

Next, the difference between the reference position and the point in the solenoid is:

$$\vec{\boldsymbol{r}} = \vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}' = \big(z - C_z(\phi)\big)\hat{\boldsymbol{e}}_{\boldsymbol{z}} - R(\phi)\hat{\boldsymbol{e}}'_{\boldsymbol{\rho}} \tag{14}$$

The magnitude of $\vec{\boldsymbol{r}}$ can be easily calculated too:

$$\|\vec{\boldsymbol{r}}\| = \sqrt{\big(z - C_z(\phi)\big)^2 + R(\phi)^2} \tag{15}$$

Biot-Savart's Law also requires a $\overrightarrow{\boldsymbol{d\ell}}$ element. This can be calculated by adding up 3 terms representing the change in arc-length of the solenoid in each direction:

$$\overrightarrow{\boldsymbol{d\ell}} = R(\phi)d\phi \cdot \hat{\boldsymbol{e}}'_{\boldsymbol{\phi}} + dR \cdot \hat{\boldsymbol{e}}'_{\boldsymbol{\rho}} + dZ \cdot \hat{\boldsymbol{e}}_{\boldsymbol{z}} \tag{16}$$

From the chain rule, it follows:

$$\overrightarrow{\boldsymbol{d\ell}} = R(\phi)d\phi \cdot \hat{\boldsymbol{e}}'_{\boldsymbol{\phi}} + R'(\phi)d\phi \cdot \hat{\boldsymbol{e}}'_{\boldsymbol{\rho}} + C'_z(\phi)d\phi \cdot \hat{\boldsymbol{e}}_{\boldsymbol{z}} \tag{17}$$

Thus, the cross product $\overrightarrow{\boldsymbol{d\ell}} \times \vec{\boldsymbol{r}}$ can be represented as:

$$\overrightarrow{\boldsymbol{d\ell}} \times \vec{\boldsymbol{r}} = (z \cdot R(\phi)d\phi)\hat{\boldsymbol{e}}'_{\boldsymbol{\rho}} + \big(C'_z(\phi)R(\phi)d\phi - R'(\phi)\big(z - C_z(\phi)\big)d\phi\big)\hat{\boldsymbol{e}}'_{\boldsymbol{\phi}} - (R(\phi)^2 d\phi)\hat{\boldsymbol{e}}_{\boldsymbol{z}} \tag{18}$$

Finally, we can write Biot-Savart's integral in cylindrical coordinates as:

$$\vec{\boldsymbol{B}}(z) = \frac{\mu_0 I}{4\pi}\int_{-\pi}^{\pi} \frac{zR(\phi)\cdot\hat{\boldsymbol{e}}'_{\boldsymbol{\rho}} + \Big(C'_z(\phi)R(\phi) - R'(\phi)\big(z - C_z(\phi)\big)\Big)\cdot\hat{\boldsymbol{e}}'_{\boldsymbol{\phi}} - R(\phi)^2\cdot\hat{\boldsymbol{e}}_{\boldsymbol{z}}}{\sqrt{\big(z - C_z(\phi)\big)^2 + R(\phi)^2}^{\,3}}\, d\phi \tag{19}$$

$\hat{\boldsymbol{e}}'_{\boldsymbol{\rho}}$ is a function of ϕ and can be represented as $\hat{\boldsymbol{e}}'_{\boldsymbol{\rho}} = \cos\phi\,\hat{\boldsymbol{e}}_{\boldsymbol{x}} + \sin\phi\,\hat{\boldsymbol{e}}_{\boldsymbol{y}}$ which can be substituted into the integral:

$$\vec{\boldsymbol{B}}(z) = \frac{\mu_0 I}{4\pi}\int_{-\pi}^{\pi} \frac{zR(\phi)\cdot\big(\cos\phi\,\hat{\boldsymbol{e}}_{\boldsymbol{x}} + \sin\phi\,\hat{\boldsymbol{e}}_{\boldsymbol{y}}\big) + \cdots}{\cdots}\, d\phi \tag{20}$$

The integral $\int_{-\pi}^{\pi}\cos\phi\,\hat{\boldsymbol{e}}_{\boldsymbol{x}}\,d\phi = \int_{-\pi}^{\pi}\sin\phi\,\hat{\boldsymbol{e}}_{\boldsymbol{x}}\,d\phi = 0$ and because of the radial symmetry of the solenoid, the radius is an even function, i.e., $R(\phi) = R(-\phi)$. Therefore, the first term cancels out to zero and we are left with:

$$\vec{\boldsymbol{B}}(z) = \frac{\mu_0 I}{4\pi}\int_{-\pi}^{\pi} \frac{\Big(C'_z(\phi)R(\phi) - R'(\phi)\big(z - C_z(\phi)\big)\Big)\cdot\hat{\boldsymbol{e}}'_{\boldsymbol{\phi}} - R(\phi)^2\cdot\hat{\boldsymbol{e}}_{\boldsymbol{z}}}{\sqrt{\big(z - C_z(\phi)\big)^2 + R(\phi)^2}^{\,3}}\, d\phi \tag{21}$$

We also know that $\oint \hat{\boldsymbol{e}}'_{\boldsymbol{\phi}}\, d\phi = 0$. Since $C'_z(\phi)$ is also an even function, part of the second term also cancels out to 0:

$$\vec{\boldsymbol{B}}(z) = \frac{\mu_0 I}{4\pi} \int_{-\pi}^{\pi} \frac{R'(\phi)\big(z - C_z(\phi)\big) \cdot \hat{\boldsymbol{e}}'_{\boldsymbol{\phi}} - \cdots}{\cdots} d\phi \tag{22}$$

$C_z(\phi)$ and $R'(\phi)$ are both odd functions. Therefore, the entire second terms cancels out to 0 over $[-\pi, \pi]$. Since $\hat{\boldsymbol{e}}_{\boldsymbol{z}}$ is a constant, it can be taken outside the integrand:

$$\vec{\boldsymbol{B}}(z) = -\frac{\mu_0 I}{4\pi} \hat{\boldsymbol{e}}_{\boldsymbol{z}} \int_{-\pi}^{\pi} \frac{R(\phi)^2}{\sqrt{\big(z - C_z(\phi)\big)^2 + R(\phi)^2}^3} d\phi \tag{23}$$

Substituting the functions for our toroidal solenoid, we can obtain the final equation. This equation is independent of n as shown in Appendix **Error! Reference source not found.**. Unfortunately, finding the anti-derivative is outside of the scope of this study and we will again have to resort to numerical integration methods.

$$\vec{\boldsymbol{B}}(z) = -\frac{\mu_0 I}{4\pi} \hat{\boldsymbol{e}}_{\boldsymbol{z}} \int_{-\pi}^{\pi} \frac{a + b\cos(\phi)}{\sqrt{(z - b\sin(\phi))^2 + a + b\cos(\phi)}^3} d\phi \tag{24}$$

Plotting the equation along the central axis gives us an exaggerated bell-curve-like relationship:

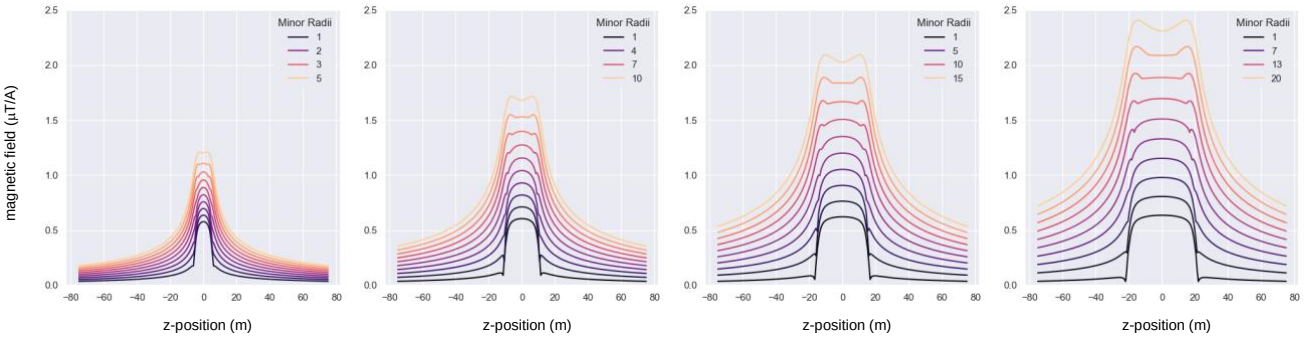


Figure 5 Numerical Integration of Magnetic Field along $x = y = 0$

These visuals support the observation that the magnetic field is strongest at the centre of the solenoid ($z = 0$) and decays as you move further away from the electric source. Additionally, larger solenoids with greater major and minor radii have stronger magnetic fields.

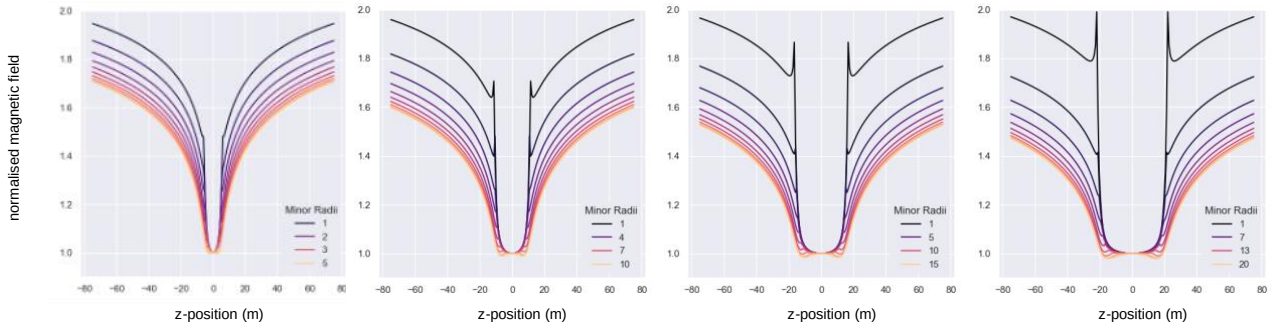
4.3 Finding an Approximating Function

Now, we wish to find a function to approximate these numerical integrals. Since the curves all resemble bell curves with a transformed x-axis, the bell curve function is a good starting point, where $B_0 = B(0)/I$ and ε is a small number added to avoid a negative square-root in (26) (taken to be 1)**:**

$$\left\|\vec{\boldsymbol{B}}(z)\right\| = B(z) \propto B_0 e^{1-f(z)^2} \tag{25}$$

Next, we can solve for $f(z)$ and arrive at an equation to transform the points in Figure 5:

$$f(z) = \sqrt{\varepsilon - \ln \frac{B(z)}{B_0}} \tag{26}$$

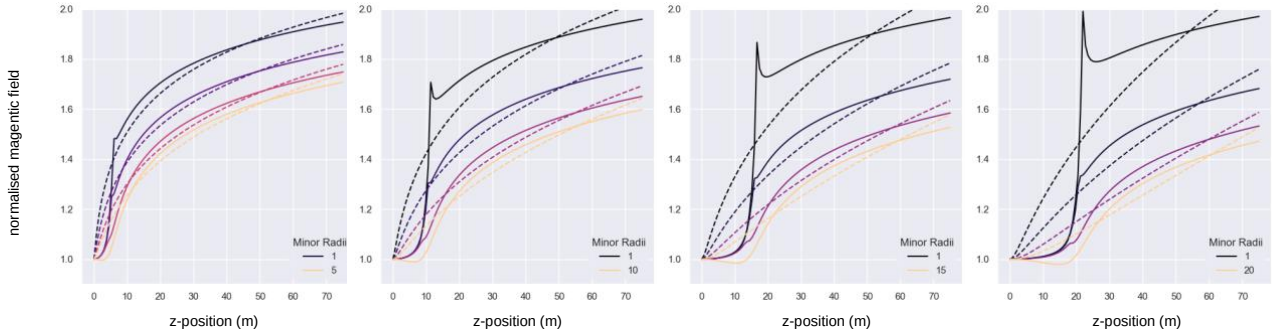


$b = 5$ $b = 10$ $b = 15$ $b = 20$

Figure 6 $f(z)$ for various Toroidal Solenoid configurations

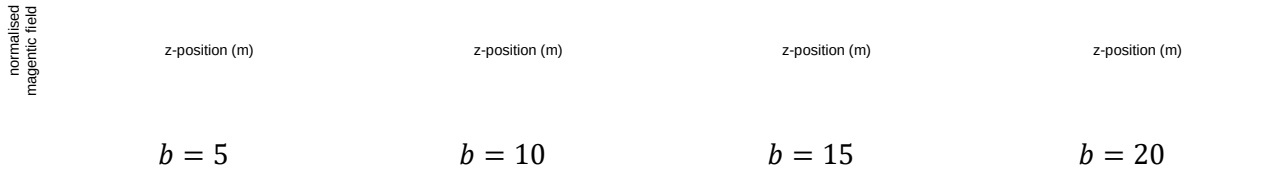
$f(z)$ is a kind of symmetric logarithmic function. We wish to optimise the parameters β_1 (coefficient of log term) and β_2 (exponent of log term) according to the following equation, to fit $f(z)$:

$$f(z) \sim f'(z) = \beta_1 \log(|z| + \varepsilon)^{\beta_2} \tag{27}$$



$b = 5$ $b = 10$ $b = 15$ $b = 20$

Figure 7 Logarithmic approximations for $f(z)$ for positive z values



$b = 5$ $b = 10$ $b = 15$ $b = 20$

Figure 7 shows the result of this optimisation computed using the Trust Region Reflective algorithm (TRF) (Yuan, 1999) (see Appendix 8.4). Clearly, this equation is not a good estimate. Plotting the residuals between $f'(z)$ and $f(z)$, we obtain the following graphs:

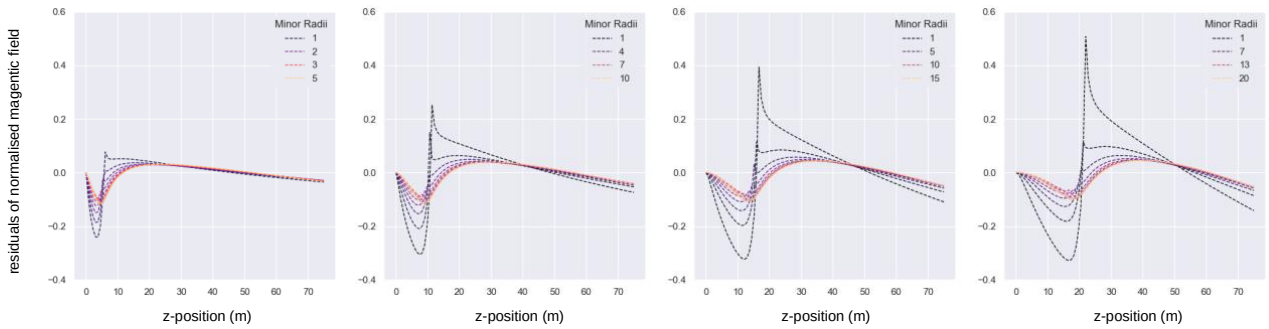


Figure 8 Residuals between $f(z)$ and a logarithmic $f'(z)$

To correct these residuals, we will employ a modified Sigmoid Linear Unit (SiLU) (Elfwing, Uchibe, & Doya, 2018) function according to the new optimisation:

$$f(z) \sim f'(z) = \beta_1 \log(|z| + \varepsilon)^{\beta_2} - \beta_3 \frac{|z|}{1 + e^{\beta_4 |z|}} \tag{28}$$

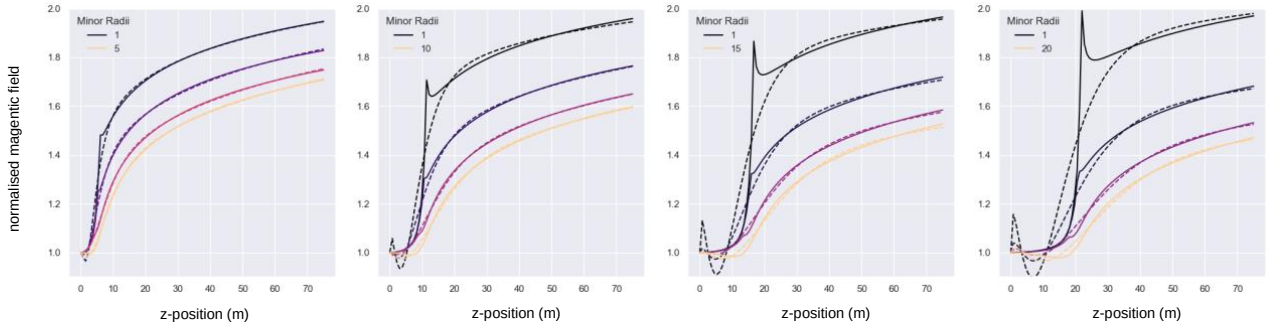


Figure 9 SiLU-corrected logarithmic approximations for $f(z)$

Substituting (28) into (25), we obtain the final approximation function. Graphical representations of the optimised parameters are given in Appendix 8.6.

$$B(z) \sim B'(z) \propto B_0 \exp\left[-\left(\beta_1 \log(|z| + \varepsilon)^{\beta_2} - \beta_3 \frac{|z|}{1 + e^{\beta_4 |z|}}\right)^2\right] \tag{29}$$

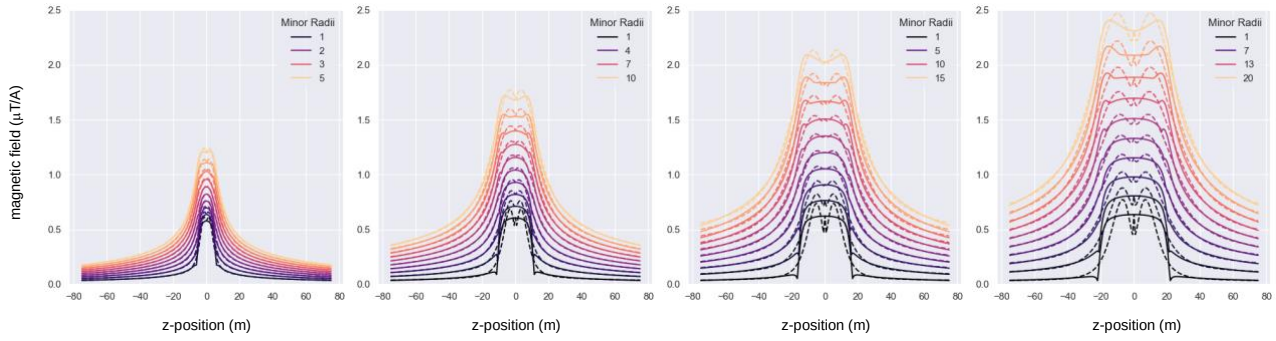


Figure 10 Final approximation of $B(z)$

5 Results & Discussion

5.1 Error Evaluation

To evaluate the central axis model, we will use the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Maximum Error metrics, evaluated for $z \in [-75, 75]$, $a \in [1, b-1] \cap \mathbb{Z}$ and $b \in [5, 10, 15, 20]$ using the following integral:

$$\text{MAE} = \frac{1}{n(a)} \sum_{a \in \mathbb{Z} \cap [1,b-1]} \int_{-75}^{75} |B(z) - B'(z)| \, dz \tag{30}$$

$$\text{MAPE} = \frac{1}{n(a)} \sum_{a \in \mathbb{Z} \cap [1,b-1]} \int_{-75}^{75} \left| \frac{B(z) - B'(z)}{B(z)} \right| dz \tag{31}$$

$$\text{Max Error} = \max_{z \in [-75,75],\ a \in \mathbb{Z} \cap [1,b-1]} |B(z) - B'(z)| \tag{32}$$

Table 1 Errors in central-axis approximation of magnetic field

b	$b = 5$	$b = 10$	$b = 15$	$b = 20$
MAE (μT/A)	0.00407	0.00822	0.01281	0.01631
MAPE (%)	0.294%	0.639%	1.004%	1.326%
Max Error (μT/A)	0.114	0.264	0.377	0.470

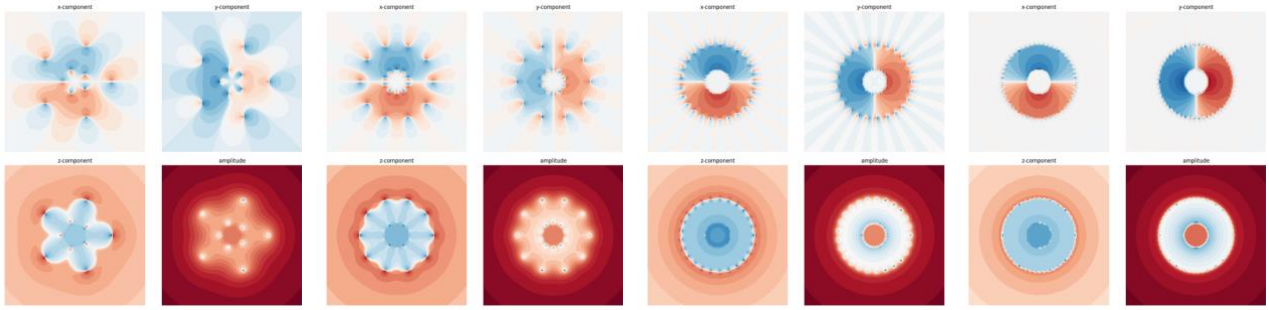
The results in Table 1 show the various metrics used to evaluate the error in the model. Generally, the central-axis approximation is accurate for smaller solenoids and loses accuracy as the major and minor radii grow. This makes sense as larger solenoids are more complicated over the $z \in [-75, 75]$ range studied and the effect of local curls in magnetic field and other changes are amplified. Of course, as we are measuring the error in micro-Teslas per Ampere, the absolute error of the approximation would grow worse as the current through the solenoid is increased, but the percentage error would stay the same.

A major strength of these results is the relatively low percentage error, being capped at 1.326% for a major radius of 20 meters. Another strength is that the mean absolute error is far smaller than the maximum error which means that, though there is a spike in error which causes a far higher max error, most deviations from the actual value are close to 0. That is, the error between the approximation and truth only spikes in some places and is nearly negligible in most others.

5.2 Cross-Sectional Magnetic Field

From these results, we can conclude qualitative attributes concerning the shape of the magnetic field and its relative strength across different sections of the toroid. First, we can see the effect of increasing the number of turns on the magnetic field as the colours get darker and more defined. The magnetic field around a toroid can be seen to have 2 sections: the inner magnetic field which

approaches 0 (white) as the number of turns increases — hypothesized to be because opposing rings of current cancel out the magnetic field inside.



$n = 5$ $n = 10$ $n = 25$ $n = 50$

Figure 11 Numerical Integration of Magnetic Field along $z = 0$ varying number of turns

The outer magnetic field increases in strength as the number of turns increases. The flower-shaped field created by local curls of current becomes overpowered by a general, circular field as the turns increase.

The x- and y-components of the field are rotationally symmetric, which follows from the toroid being symmetric around the z-axis. The magnetic field components exist in 2 opposing hemispheres (red and blue) which become more defined with more turns. This effect is likely due to the change in the direction of flowing current as it moves along a circular path around the toroid, which reverses the direction of the magnetic field after 180°.

5.3 Central-Axis Magnetic Field

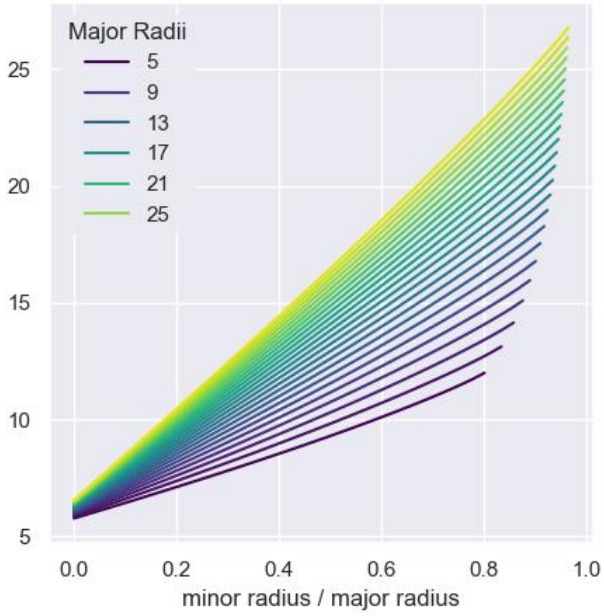
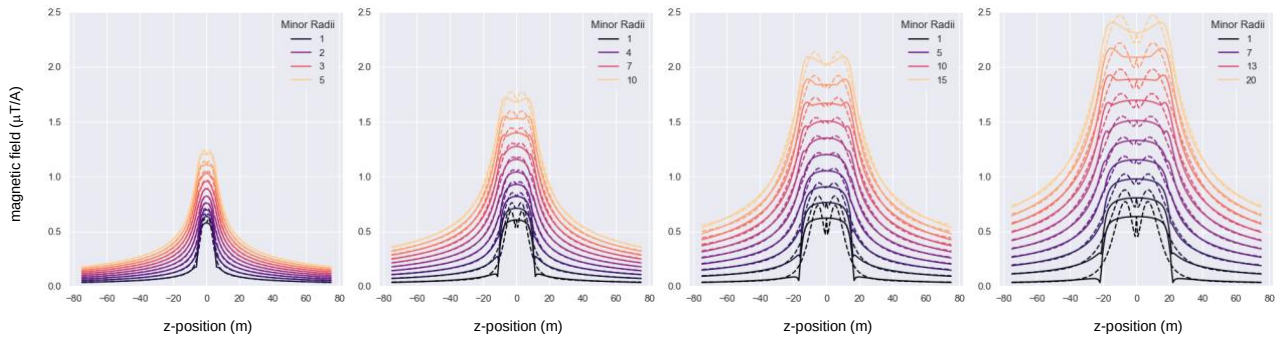


Figure 12 Peak magnetic field ($\mu T/A$) for different configurations

An interesting conclusion of this research was that the central magnetic field around a toroidal solenoid is independent of the number of turns.

Error! Reference source not found.Error! Reference source not found. shows values of the peak amplitude of the magnetic field, B_0 for different toroidal configurations. It is evident that increasing the major radius strictly increases the peak magnetic field. The x-axis represents the ratio between radii where 0 is a minor radius of 0 and 1 is a minor radius equal to the major radius. As the minor radius approaches the major radius, the magnetic field strength strictly and linearly increases. We can conclude that the magnetic field is directly proportional to the ratio between the minor and major radii and also directly proportional to the magnitude of the major radii.



$b = 5$ $b = 10$ $b = 15$ $b = 20$

Figure 13 Final approximation of $B(z)$

Figure 13 shows the magnetic field at different distances to the centre of the toroid. Similar to the magnetic field along the cross-sectional plane, there are 2 distinct sections (inner and outer) for the field along the central axis. The magnetic field is the strongest and nearly constant inside the toroid, where $z \leq$ the major radius. Outside the toroid, the field from opposing ends of the solenoid likely cancels out and exponentially tapers off to 0. The field is inversely proportional to the distance to the toroid outside the solenoid and directly proportional to the minor radius inside.

6 Conclusion

This exploration has established a method to approximate the central-axis magnetic field around a toroidal solenoid and derived qualitative results for the central cross-sectional plane of said solenoid. Using Biot-Savart's integral, a numerical evaluation of the magnetic field along the plane $x = y = 0$ was performed and using a SiLU corrected, logarithmically transformed normal curve, the field along $z = 0$ was approximated to a very low mean absolute error, mean absolute percentage error, and fairly low maximum error.

A significant strength of this investigation is its applicability to any toroidal solenoid which can be characterised by its pitch (number of turns) and radii. Thus, the results from this study are highly generalisable to many scenarios. However, its theoretical nature remains a weakness as these results have not yet been verified by experimental data. Another limitation of the approximation function created is while the B_0 parameter has an exact physical interpretation as the maximum amplitude of the magnetic field, the parameters β_n are highly abstract and do not possess any similar description. This makes understanding the correlation between the independent and dependent variable more difficult.

The results nearly fully satisfy the predictions in the hypothesis. Increasing the number of turns does produce a more symmetric, uniform, and defined magnetic field. The x- and y-components of the field do approach 0 outside the solenoid and are the strongest inside. The z-component does flow the strongest through the centre of the toroid and converges near the middle to become amplified. The only deviation from the hypothesis is that the z-component of the magnetic field inside the toroid remains nearly constant and only exponentially decreases outside, and, surprisingly, this field is independent of the number of turns in the solenoid.

In future studies, better approximation functions can be used to distinctly model the inner and outer fields, further reducing the evaluated error. The complete 3D effect of the magnetic field is yet to be

studied, and cross-sectional planes at different places in the solenoid can be modelled. The theoretical assumption of a 0-wire-width solenoid should also be broken, and the effect of common wire gauges can be modelled, though these are likely negligible with sufficiently high minor and major radii. Lastly, experimental data is vital to prove the findings of this study and datasets must be created that accurately map the magnetic field components around a toroidal solenoid.

7 Bibliography

Chua, D., & Tong, D. (2015, Lent). Retrieved April 2023, from Part IB - Electromagnetism: http://dec41.user.srcf.net/notes/IB_L/electromagnetism_trim.pdf

Sanderson, G. (2015, June 21). *Divergence and curl: The language of Maxwell's equations, fluid flow, and more.* Retrieved May 2023, from YouTube: https://www.youtube.com/watch?v=rB83DpBJQsE

Elfwing, S., Uchibe, E., & Doya, K. (2018). Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Networks, 107*, 3-11.

Yuan, Y.-x. (1999, September). A Review of Trust Region Algorithms for Optimization. *ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics*.

Harris, C., Millman, K., Walt, v. d., & al., S. e. (2020). Array programming with NumPy. *Nature, 585*, 357–362.

Virtanen, P., Gommers, R., & Oliphant, T. E. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods, 17*, 261--272.

Laslett, O., Waters, J., Fangohr, H., & Hovorka, O. (2018, January). Magpy: A C++ accelerated Python package for simulating magnetic nanoparticle stochastic dynamics.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90-95.

Waskom, M. L. (2021). Seaborn: statistical data visualization. *The Open Journal, 6*(20).

8 Appendices

8.1 Program to Visualise Toroidal Solenoid

```
import numpy as np
import magpylib as magpy
import matplotlib.pyplot as plt


MAJOR_RADIUS = 14  # mm
MINOR_RADIUS = 7  # mm
TURNS = 100
CURRENT = 100  # A
```

```
def create_toroidal_solenoid():
    ts = np.linspace(0, 2 * np.pi, 10000)

    vertices = np.c_[(MAJOR_RADIUS + MINOR_RADIUS * np.cos(ts * TURNS)) * np.cos(ts),
                     (MAJOR_RADIUS + MINOR_RADIUS * np.cos(ts * TURNS)) * np.sin(ts),
                     MINOR_RADIUS * np.sin(ts * TURNS)]
    coil = magpy.current.Line(current=CURRENT,
                              vertices=vertices)

    return coil


def show(coil):
    coil.show(backend="plotly")
```

8.2 Program to Compute Biot-Savart's Integral

```
b = 1  # mm
a = 7  # mm
n = 10


def get_integration_grid(grid_points, res):
    print("Constructing Grid...")

    grid = np.linspace(-40, 40, grid_points)
    t_range = np.arange(0, 2 * np.pi + res, res)
    x = np.tile([x for x in grid], (t_range.size, grid.size, 1))
    y = np.transpose(np.tile([y for y in grid], (t_range.size, grid.size, 1)),
                     axes=(0, 2, 1))
    z = np.zeros((t_range.size, grid.size, grid.size))
    mu = np.tile(t_range, (grid.size, grid.size, 1)).T

    return grid, t_range, x, y, z, mu


def integrand(r):
    rx, ry, rz = r

    def _integrand(t):
        cos_nt = np.cos(n * t)
        sin_nt = np.sin(n * t)
        cos_t = np.cos(t)
        sin_t = np.sin(t)

       a_plus_b_cos_nt = (a + b * cos_nt)

        ux = rx - a_plus_b_cos_nt * cos_t
        uy = ry - a_plus_b_cos_nt * sin_t
        uz = rz - b * sin_nt

        dux = - a_plus_b_cos_nt * sin_t - n * b * cos_t * sin_nt
        duy = a_plus_b_cos_nt * cos_t - n * b * sin_t * sin_nt
        duz = n * b * cos_nt

        d = np.sqrt(ux ** 2 + uy ** 2 + uz ** 2)
        d3 = d ** 3

        bx = duz * uy - duy * uz
        by = dux * uz - duz * ux
        bz = duy * ux - dux * uy

        bx /= d3
        by /= d3
        bz /= d3

        return [bx, by, bz]
```

```
    return _integrand


def plot_magnetic_field(grid_points=75, res=0.0005, clamp=0.0005,
                        kwargs={"cmap": "RdBu", "levels": 32}):

    grid, t_range, x, y, z, mu = get_integration_grid(grid_points, res)

    print("Integrating...")
    Bx, By, Bz = integrand((x, y, z))(mu)

    Bx = 2 * np.pi * np.mean(Bx, axis=0)
    By = 2 * np.pi * np.mean(By, axis=0)
    Bz = 2 * np.pi * np.mean(Bz, axis=0)

    Bx = np.clip(Bx, np.quantile(Bx, clamp), np.quantile(Bx, 1 - clamp))
    By = np.clip(By, np.quantile(By, clamp), np.quantile(By, 1 - clamp))
    Bz = np.clip(Bz, np.quantile(Bz, clamp), np.quantile(Bz, 1 - clamp))

    B = np.sqrt(Bx ** 2 + By ** 2 + Bz ** 2)
    B = np.clip(B, np.quantile(B, clamp), np.quantile(B, 1 - clamp))

    print("Plotting...")
    fig, ((ax00, ax01), (ax10, ax11)) = plt.subplots(2, 2, figsize=(10, 10),
                                                     sharex=True, sharey=True)
    ax00.axis("off")
    ax01.axis("off")
    ax10.axis("off")
    ax11.axis("off")
    fig.tight_layout(pad=1, h_pad=3, rect=(0, 0, 1, 0.95))
    ax00.contourf(grid, grid, np.cbrt(Bx), **kwargs)
    ax00.set(title="x-component", aspect=1)

    ax01.contourf(grid, grid, np.cbrt(By), **kwargs)
    ax01.set(title="y-component", aspect=1)

    ax10.contourf(grid, grid, np.cbrt(Bz), **kwargs)
    ax10.set(title="z-component", aspect=1)

    ax11.contourf(grid, grid, B, **kwargs)
    ax11.set(title="magnitude", aspect=1)

    plt.show()
```

8.3 Program to Compute Magnetic Field along Central Axis

```
def integrand(z_val, a, b=MAJOR_RADIUS):

    def _integrand(phi):
        numerator = a + b * np.cos(phi)
        denominator = a + b * np.cos(phi) + (z_val - b * np.sin(phi)) ** 2

        return numerator / np.cbrt(denominator)

    return _integrand


def plot_magnetic_field_along_center():

    plt.figure(figsize=(5, 5))

    for a in np.linspace(1, MAJOR_RADIUS, 10):
        x = np.linspace(0, 75, 100)
        y = [integrate.quad(integrand(z, a), -np.pi, np.pi, limit=5000)[0] for z in x]

        x = list(-x[::-1]) + list(x)
        y = list(y[::-1]) + list(y)
```

```
        plt.plot(x, y, c=plt.cm.magma(a / MAJOR_RADIUS - 0.1))

    plt.xlabel("z-position")

    legend = plt.legend(title="Minor Radii")
    legend.get_frame().set_linewidth(0.0)

    plt.tight_layout()
    plt.show()
```

8.4 Program to Optimise $f(z)$ using Trust Region Reflection

```
def logarithmic_fz(x, b1, b2):
    x = abs(x)
    return b1 * np.log(x + 1) ** b2 + EPSILON


def silu_logarithmic_fz(x, b1, b2, b3, b4):
    x = abs(x)
    log_factor = b1 * np.log(x + 1) ** b2
    correction_factor = b3 * x / (1 + np.exp(b4 * x))

    return log_factor + correction_factor + EPSILON

parameters = optimize.curve_fit(func, xdata=x, ydata=fz, method="trf")[0]
```

8.5 Program to Plot & Save β parameters

```
def plot_parameters(func=silu_logarithmic_fz):
    minor_radii = []
    all_parameters = []

    major_radii = range(5, 29)

    for b in major_radii:
        parameters = [(1, 1, 1, 1, 1)]
        minor_radii.append(np.linspace(1, b, 40))

        for a in minor_radii[-1]:
            z = np.linspace(0, 100, 100)
            B = np.array([integrate.quad(integrand(val, a=a, b=b),
                         -np.pi, np.pi, limit=5000)[0] for val in z])

            B0 = integrate.quad(integrand(0, a=a, b=b), -np.pi, np.pi, limit=5000)[0]
            divisor = B0 * 1.1

            B = np.sqrt(-np.log(B / divisor))
            B -= np.sqrt(-np.log(B0 / divisor))
            mask = ~np.isnan(B)

            params = optimize.curve_fit(func, p0=parameters[-1][:4], xdata=z[mask],
                                        ydata=B[mask], method="trf")[0]
            parameters.append(list(params) + [B0])
        all_parameters.append(list(zip(*parameters)))

    with open("parameters.bin", "wb") as file:
        pickle.dump(all_parameters, file)

    for param in range(5):
        plt.figure(figsize=(5, 5))
        plt.tight_layout()
```

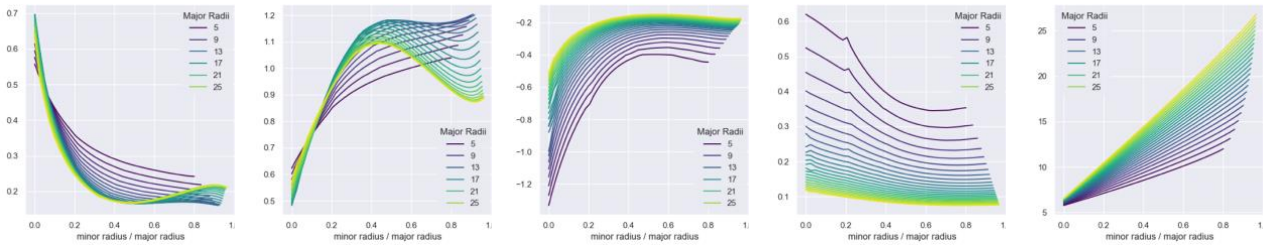
```
        plt.xlabel("minor radius / major radius")

        for i, (x, y) in enumerate(zip(minor_radii, all_parameters)):
            color = plt.cm.viridis(i / len(major_radii))
            plt.plot((x - 1) / major_radii[i], y[param][1:], c=color,
                     label=major_radii[i] if i % 4 == 0 else "")

        legend = plt.legend(title="Major Radii")
        legend.get_frame().set_linewidth(0.0)

    plt.show()
```

8.6 Graphs of β Parameters



β_1 β_2 β_3 β_4 B_0

Figure 14 Graphs of the Parameters of the SiLU-corrected logarithmic approximations for $f(z)$