

**MASTER**

**Designing technical solutions to magnetic configuration problems**  
**CIRCUS**

van der Voort, Peter M.Q.

*Award date:*  
2021

[Link to publication](#)

**Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Department of Applied Physics  
Science and Technology of Nuclear Fusion

**DESIGNING TECHNICAL SOLUTIONS TO  
MAGNETIC CONFIGURATION PROBLEMS  
CIRCUS**

M.Sc. Graduation Project Report

Author:  
P.M.Q. van der Voort

TUe supervisors:  
R.J.E. Jaspers  
N. Maassen

Eindhoven, July, 2021

## Abstract

CIRCUS is a concept hybrid tokamak stellarator device, which makes use of interlocked toroidal magnetic coils. In this master thesis a magnetic architecture where CIRCUS can confine a plasma is studied and a final design is presented. A numerical model has been developed, applying the Biot-Savart law to simulate the magnetic field in a three dimensional volume.

The study resulted in various key findings. CIRCUS is not able to operate in a purely stellarator like mode, without an additional poloidal field the magnetic field is too sensitive to perturbations. In this context it is concluded that a wire through the center axis of the plasma is the easiest method to achieve the required poloidal magnetic field.

Mechanical influences on the magnetic field have been investigated as well. The temperature increase during operation of the coils limits the operating time of CIRCUS. Further, the magnetic forces on the coils are not strong enough to influence the magnetic architecture significantly.

## List of Symbols

Symbols	Description	Units
$a$	Minor radius	[ $m$ ]
$A$	Surface area	[ $m^2$ ]
$b$	Thickness	[ $m$ ]
$B$	Magnetic field	[ $T$ ]
$B_t$	Toroidal magnetic field	[ $T$ ]
$B_p$	Poloidal magnetic field	[ $T$ ]
$cp$	Heat capacity	[ $JK^{-1}$ ]
$D$	Diameter	[ $m$ ]
$e$	Elementary charge	[ $C$ ]
$E$	Energy	[ $J$ ]
$E_m$	Elasticity modulus	[ $GPa$ ]
$F$	Force	[ $N$ ]
$h$	Height	[ $m$ ]
$k$	Boltzmann's constant	[ $m^2kgs^{-2}K^{-1}$ ]
$I$	Current	[ $A$ ]
$j$	Current density	[ $Am^{-2}$ ]
$l$	Length element	[ $m$ ]
$L$	Length	[ $m$ ]
$L_p$	Inductance	[ $H$ ]
$m$	Mass	[ $kg$ ]
$n$	Plasma particle density	[ $m^{-3}$ ]
$N_{turns}$	Windings on coil	[–]
$N_{coils}$	Number of coils	[–]
$T$	Temperature	[ $K$ ]
$T_e$	Electron temperature	[ $ev$ ]
$p$	Pressure	[ $Pa$ ]
$P$	Power	[ $watt$ ]
$q$	Heat transfer	[ $W$ ]
$r$	Radius	[ $m$ ]
$R$	Resistance	[ $\Omega$ ]
$R_{coil}$	Radius from center coil to center CIRCUS	[ $m$ ]
$R_0$	Major radius	[ $m$ ]
$V$	Volt	[ $V$ ]
$x$	Coil x location	[ $m$ ]
$y$	Coil y location	[ $m$ ]
$z$	Coil z location	[ $m$ ]
$Z_{coil}$	Height coil from center CIRCUS	[ $m$ ]



<b>Symbols</b>	<b>Description</b>	<b>Units</b>
$\alpha$	Coil angle	[°]
$\alpha_{cu}$	Temperature coefficient	[-]
$\epsilon_0$	vacuum permitivity	$[Nm^2C^{-2}]$
$\lambda$	Decay length	[m]
$\mu_0$	Magnetic permeability of vacuum	$[Hm^{-1}]$
$\nu$	Spitzer resisivity	$[\Omega m]$
$\rho$	Resistivity	$[\Omega m]$
$\rho_m$	Density	$[kgm^{-2}]$
$\sigma$	Stefan-Boltzmann constant	$[W/m^2K^4]$
$\sigma_c$	Cross-section charge exchange	$[m^2]$
$\sigma_i$	Cross-section ion collision	$[m^2]$
$\sigma_t$	Tensile stress	[Pa]
$\sigma_y$	Yield tensile strength	[MPa]
$\tau_E$	Energy confinement time	$[keVm^{-3}]$
$\Phi$	Magnetic flux	[Vs]
$\omega_{ce}$	Electron cyclotron frequency	$[s^{-1}]$

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Reasons for fusion energy . . . . .	1
1.2	Fusion energy . . . . .	1
1.3	Hybrid concept device CIRCUS . . . . .	2
1.4	Research problem . . . . .	4
1.5	Assumptions . . . . .	5
<b>2</b>	<b>CIRCUS magnetic field model</b>	<b>6</b>
2.1	Coil architecture . . . . .	6
2.2	Developing magnetic field simulation . . . . .	7
2.3	Validating developed magnetic field model . . . . .	8
<b>3</b>	<b>CIRCUS magnetic field</b>	<b>12</b>
3.1	Determining toroidal magnetic field . . . . .	12
3.2	Determining vertical magnetic field . . . . .	16
3.3	Quadropole coils contribution . . . . .	22
3.4	Complete magnetic field . . . . .	24
3.5	Particle gyration . . . . .	25
3.6	Sensitivity coil parameters . . . . .	26
3.7	Conclusions . . . . .	28
<b>4</b>	<b>CIRCUS poloidal magnetic field</b>	<b>29</b>
4.1	MATLAB simulation increasing poloidal field . . . . .	29
4.2	Sensitivity investigation of magnetic field . . . . .	31
4.3	Conclusions . . . . .	33
<b>5</b>	<b>Poloidal magnetic field generation methods</b>	<b>34</b>
5.1	Neutral beam current drive . . . . .	34
5.2	Ohmic current drive . . . . .	36
5.3	Electron cyclotron current drive . . . . .	38
5.4	Current wire . . . . .	39
5.5	Conclusions . . . . .	41
<b>6</b>	<b>Disturbances magnetic field</b>	<b>42</b>
6.1	Disturbance magnetic field due to heat . . . . .	42
6.2	Disturbance magnetic field due to magnetic forces on coils and wire . . . . .	45
6.3	Conclusions . . . . .	51
<b>7</b>	<b>Final conclusions and outlook</b>	<b>53</b>
7.1	Final conclusions . . . . .	53
7.2	Outlook . . . . .	54
	<b>References</b>	<b>56</b>

<b>Appendices</b>	<b>58</b>
<b>A Locations six toroidal field coils CIRCUS</b>	<b>58</b>
<b>B Coil parameters CIRCUS proposed by Volpe</b>	<b>58</b>
<b>C MATLAB-code Quéval</b>	<b>59</b>
<b>D MATLAB-code</b>	<b>65</b>

# 1 Introduction

## 1.1 Reasons for fusion energy

With increasing population and rising living standards the world's energy demand is ever-growing (Everett et al., 2011). Most energy is generated using fossil fuels, like oil and gas, which is unsustainable and at the same time it is changing the climate due to amongst others  $CO_2$  emissions. These emissions have a catastrophic effect on earth and human civilization if a temperature increase of more than  $2^\circ C$  is reached (Schleussner et al., 2016). A somewhat lesser-known drawback of fossil fuels is the waste dumped into the air everyone breathes. Coal ash for example has particles of lead, mercury, aluminum and many more toxic materials (Earthjustice, 2017), causing a lot of deaths every year (Ritchie, 2020).

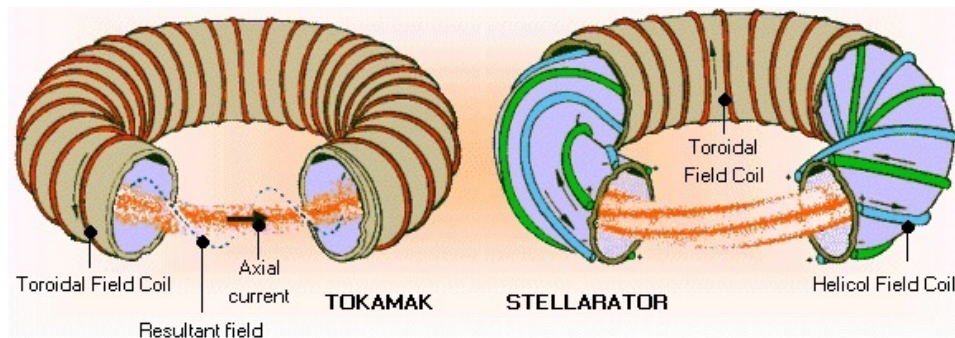
Cleaner, non-fossil energy resources, such as wind and the sun, are not likely to be a realistic solution to this problem on the long term as they produce way less energy per square meter ( $W/m^2$ ) resulting in large parts of countries necessary to deploy these technologies (Department of Energy United States of America, 2015). Besides, wind and solar are unreliable and when there is no wind or sun other energy sources must take over, which is most often gas.

In view of the above fusion energy makes an attractive alternative for fulfilling the ever-growing energy demand as its energy produced per square meter is multiples higher than other non-fossil energy sources (Van Zalk & Behrens, 2018). In addition, the waste product of the fusion process is 'clean' and useful for industry, while a small amount of radio active material can be stored in a safe location. This in contrast with the environment polluting fossil energy waste. Fusion energy however is not a solution in the short term, a lot of research still has to be done to make a functional power plant.

## 1.2 Fusion energy

Energy is released when two atomic nuclei fuse into one. This energy is captured and converted into electricity. During this fusion process the atomic particles lose a small part of their mass transferring into energy, which can be calculated using Einstein's famous equation:  $E = mc^2$ . The sun performs this process on a large scale continuously, where in the core hydrogen particles are compressed by an enormous gravity. Lacking such gravitational forces on earth, different schemes have been thought of to let fusion occur. One of these is magnetic confinement.

There are two main concepts for magnetic confinement, both of which are similar. One is the tokamak while the other is called the stellarator.



**Figure 1.1:** Schematic overview tokamak and stellarator concepts (Euratom-CEA, 2016).

To confine the plasma both technologies use a helical magnetic field, but the two devices differ in the way they generate this helical magnetic field. In a tokamak this is generated by external toroidal field coils and a plasma current, which generates a magnetic field in the poloidal direction, leading to the helical field. In a stellarator this helical field is completely generated by external coils. Both devices have advantages and disadvantages.

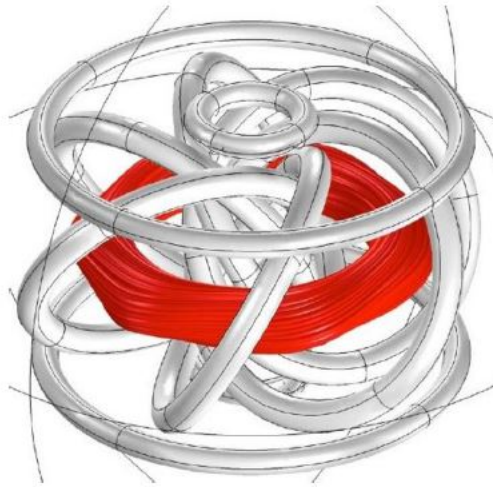
A tokamak is less complex and is further in development than a stellarator at this point. But the high plasma current can cause instabilities in the plasma. If a disruption occurs in a large tokamak the amount of energy released is very high and can damage the machine. The tokamak also needs additional devices to induce the plasma current, which is most often a central solenoid. This makes the tokamak an inherent pulsed device.

A stellarator requires no plasma current and can therefore in theory operate steady-state. Since there is no plasma current, disruptions do not have a damaging effect on the device. The fabrication of the stellarator coils however are very complex, which makes it hard to design and fabricate. This also makes it more expensive.

### 1.3 Hybrid concept device CIRCUS

Since the end of 2019 the group of Science and Technology of Nuclear Fusion of the Eindhoven University of Technology (TUE) is in possession of a small scale hybrid tokamak stellarator, called CIRCUS, which stands for CIRCULAR coil Stellarator, of which a graphical representation is displayed in figure 1.2.

The main objective of the device is to research an alternative hybrid tokamak stellarator concept, which makes use of easier to fabricate planar coils and are therefore less expensive. These coils are interlinked and on an angle, which may help reduce the required plasma current and therefore reduce common instabilities seen in tokamaks.



**Figure 1.2:** Graphical representation of CIRCUS, 6 tilted TF coils (Volpe et al., 2014).

In the figure above the coils are shown in gray, while the simulated flux lines are in red. The coils are made from multiple loops of copper wires, which are insulated with a two-component epoxy. The plasma is intended to be confined in the red area with a major radius of around  $R_0 = 0.16[m]$  and a minor radius of about  $a = 0.05[m]$ .

A construction of aluminium has been built to keep the coils in place. The apparatus is placed inside a vacuum chamber in which the required low particle density to create a plasma can be reached.

CIRCUS makes use of a tilted interlocked circular magnets system. These magnets are simpler circular coils to construct than conventional stellarator designs. Having the coils interlinked instead of separated reduces the aspect ratio of the plasma. The aspect ratio is the ratio between the small radius and the large radius of the plasma torus. Reducing this ratio makes a tokamak plasma more stable, in particular kink instabilities are strongly suppressed (Bruhns et al., 1987). Because of the tilt of the toroidal field coils each coil generates a magnetic field that has a vertical and a toroidal part. The vertical part of the magnetic field must be counterbalanced with additional vertical field coils. This subject will be further discussed in section 3.2.

Changing the angle of the toroidal field coils changes the contribution to the horizontal and vertical field of the magnets. Different angles give different magnetic architectures. For this study an angle of 45 degrees is assumed.

As stated, the reason for a device such as CIRCUS is to reduce some of the problems tokamaks and stellarators individually have. The high plasma current of a tokamak makes it sensitive to disruptions and instabilities in the plasma. The pulse length of a tokamak also depends on the maximum flux swing of the central solenoid and the amount of plasma current that needs to be generated by this process. The maximal flux swing is constrained by the material properties of the

central solenoid.

For a tokamak stellarator hybrid device the scheme to increase pulse length is by decreasing the required plasma current. This can be achieved by introducing the stellarator concept of externally generating a poloidal magnetic field with magnets. The lower required current allows for alternative methods to generate a plasma current, which will be introduced later on in this study.

## **1.4 Research problem**

For CIRCUS to achieve the objectives stated in the previous section, the device must be able to confine a plasma.

In this master thesis an investigation will be performed on the question if the magnetic configuration of CIRCUS is capable to confine a plasma. In this context the limitations of the magnetic configuration such as the maximum plasma volume, sensitivity to construction misalignments, and the mechanical influences of temperature and magnetic forces on the coils will be investigated too.

### **1.4.1 Research problem breakdown**

In order to achieve the above stated goals, the research question can be broken down into the following components:

- There is currently no reliable method to determine the magnetic field architecture. Therefore, a code will have to be developed to accomplish this task reliably and which is able to simulate the magnetic field in a three dimensional environment. This code will assist the investigation of the magnetic field and Lorentz forces on the coils.
- The appropriate currents for the coils will have to be found for the magnetic flux surfaces to be closed. Accordingly, the size of the vertical field and the influence of the quadrupole coils will be examined. Additionally, the magnetic field has certain prerequisites. A commercially available microwave must be able to use electron cyclotron resonance at the center line of the plasma. Furthermore, the field must be stable even if the current or location of the coils deviate slightly from their intended values and finally the plasma volume must be large enough to confine the gyrating particles.
- The effect of a potential plasma current in CIRCUS on the stability and the volume of the plasma will have to be investigated.
- In addition a feasible method to generate the additional poloidal magnetic field has to be found. In this regard several methods to generate a plasma current will be explored.

- The final issue is which disturbances can be expected with the current CIRCUS setup. Two mechanical factors may influence the magnetic architecture. One factor is heating of the coils and the other is deformations due to the magnetic forces. The effect of these disturbances on CIRCUS will have to be examined.

## 1.5 Assumptions

Some assumptions about the plasma and the device will be adopted in order to perform particular calculations. The physical units chosen are on the low side compared with previous small scale experiments (Leiser, 1991). These values however are expected to be reachable in this device. Nitrogen is used because it is an available and safe gas. Deuterium and Tritium are not used, because the goal of the experiment is not for fusion to occur.

	Value / Unit
Electron temperature ( $T_e$ )	1[eV]
Ion temperature ( $T_i$ )	1[eV]
Plasma density ( $n_{20}$ )	0.001[ $10^{20}m^{-3}$ ]
Gas feed	$N_2$
Toroidal coil angle	45°

**Table 1.1:** Table of assumptions about CIRCUS.

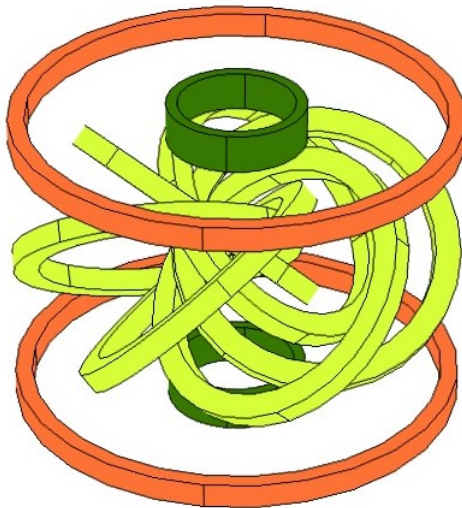


## 2 CIRCUS magnetic field model

From the research breakdown we learned that a method is needed to investigate reliably the magnetic field. Therefore an appropriate simulation model in MATLAB will be developed in this chapter. To this end the coil architecture will have to be determined first.

### 2.1 Coil architecture

CIRCUS has five tilted interlocked circular Toroidal Field (TF) coils. These coils have a radius of  $15.4[cm]$  and 69 windings each. They are placed on an angle of  $45^\circ$  and in such a manner that the center of the TF coils are  $9.24[cm]$  from the center line of the device. The coils are indicated in yellow in figure 2.1.



**Figure 2.1:** Graphical representation of CIRCUS.

The Vertical Field (VF) coils are shown in orange and the Quadrupole Field (QF) coils are indicated in green. The VF coils generate the vertical magnetic field that counters the vertical part of the magnetic field generated by the TF coils. The QF coils are mainly there to shape the vertical field. The VF and QF coils have a radius of  $25.25[cm]$  and  $7.48[cm]$ , respectively. These coils are placed respectively  $15.4[cm]$  and  $15[cm]$  above and below the vertical center axis. The VF coil has 54 windings, while the QF coil has 56 windings. The locations of all coils are summarised in table 2.1.

	x [cm]	y [cm]	z [cm]	Radius [cm]	Angle [°]	Winding's [-]
TF1	2.86	8.79	0	15.4	45	69
TF2	-7.48	5.43	0	15.4	45	69
TF3	-7.48	-5.43	0	15.4	45	69
TF4	2.86	-8.79	0	15.4	45	69
TF5	9.24	0	0	15.4	45	69
VF1	0	0	15.4	25.25	0	54
VF2	0	0	-15.4	25.25	0	54
QF1	0	0	15	7.48	0	56
QF2	0	0	-15	7.48	0	56

**Table 2.1:** Table of coil locations in centimeters with center of device as origin.

## 2.2 Developing magnetic field simulation

With the exact coil locations it is possible to create a model for simulating the magnetic field of CIRCUS. This code will be based on an open source magnetic field code (Quéval, 2015), which is explained below and can be found in the appendix C.

### 2.2.1 Base model

The open source base code makes use of the Biot-Savart Law, equation 2.1, to calculate the magnetic field in a two dimensional numeric plane. This formula is applied to calculate the magnetic field strength in each point created in the numeric plane.

$$d\vec{B} = \frac{\mu_0 I d\vec{L} \times \vec{1}_r}{4\pi r^2} \quad (2.1)$$

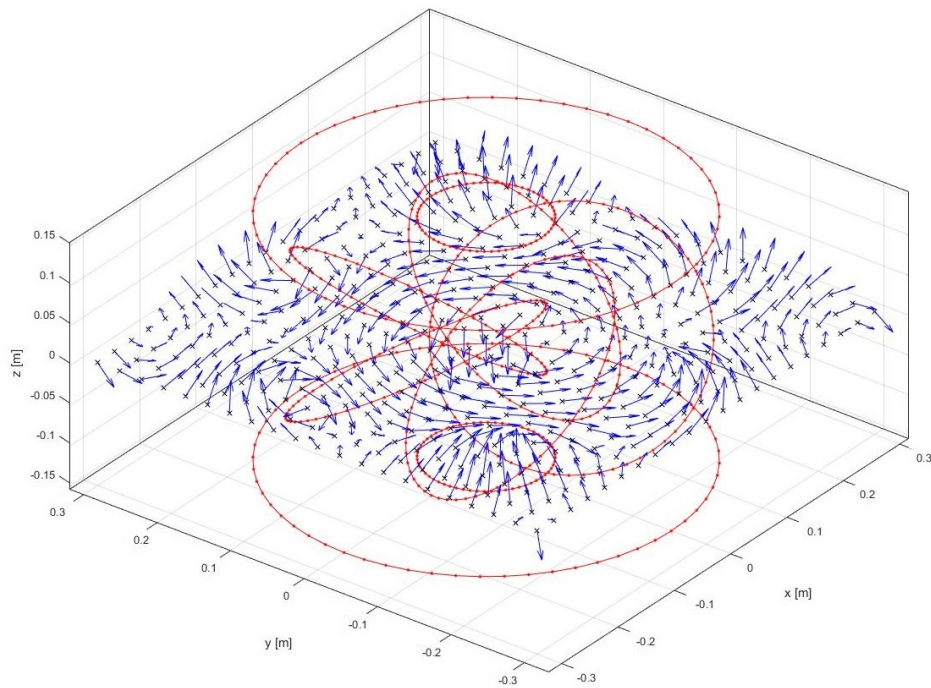
In this equation the derivative of the magnetic field,  $d\vec{B}$ , is calculated. In the formula is  $d\vec{L}$  the length of the conductor, carrying the electrical current  $I$  in [A]. Further,  $r$  in [m] is the distance from the calculated point to the element with the direction vector  $\vec{1}_r$ . The code makes use of this formula and calculates in an efficient way, which makes the model relatively fast.

### 2.2.2 Developing magnetic model for CIRCUS

The base model in the previous subsection to calculate a magnetic field has been developed further to suit our purpose. The model is now specifically made to calculate the magnetic field of CIRCUS. The base code only makes use of a two dimensional numerical plane where the magnetic field is calculated. This has been changed to a three dimensional volume. Besides, the amount of integration points can now be controlled: for different figures different integration point locations and densities can be used. In appendix D this MATLAB-code is included.

In this model the circular coils of CIRCUS have been added according to table 2.1. Here it is assumed that the coils can be approximated by infinitely thin elements. Each coil is divided into 600 smaller elements. Testing shows that increasing the amount of elements does not change the results significantly.

Figure 2.2 is an isometric view of a graph created with this MATLAB-code. This figure shows only a two dimensional numerical field in the x-y plane, in which the magnetic field is calculated.



**Figure 2.2:** *Isometric view of MATLAB-code with magnetic field vectors along x-y plane.*

In the figure above the red lines represent the coils. Here the points, where the magnetic field has been calculated, are shown on the z-is-zero-plane. For the rest of the calculations the whole space where the plasma is expected, is filled with these integration points with a space of  $0.0037[m]$  between them. Increasing the amount of numerical points in the volume does not change the results significantly. The blue vectors originating in the integration points indicate the direction and strength of the magnetic field.

### 2.3 Validating developed magnetic field model

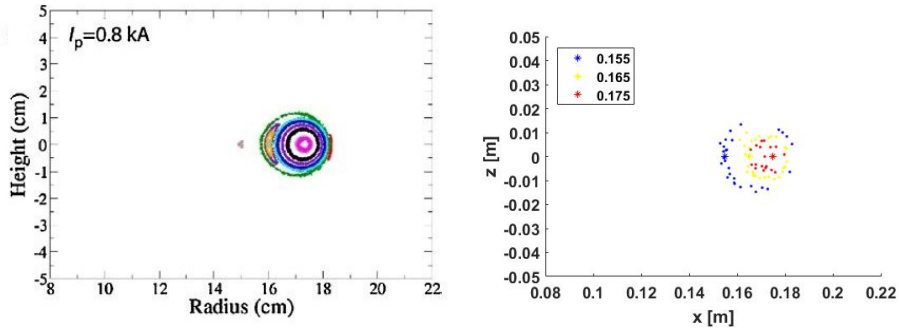
In order to validate the developed code, simulations from a paper on a six toroidal coil version of CIRCUS, (Volpe et al., 2014), will be compared to the model described in this chapter by applying the same coil currents and locations as proposed

in the paper. The coil currents are presented in table 2.2 and the locations of the coils can be found in appendix A. The results will be compared to see if there is a deviation between them. If both simulations show the same results, then it can be assumed that the developed MATLAB-code is correct.

Coil	$R[cm]$	$Z_{coil}[cm]$	$R_{coil}[cm]$	$I[A]$	$N_{turns}[-]$	$R[m\Omega]$	$V[V]$
TF	16	0	10.7	75	69	297	22.3
VF	25.25	15.4	0	75	54	177	13.3
QF	7.25	15	0	75	56	95	7.1

**Table 2.2:** Proposed coil settings for 6 TF version CIRCUS (Volpe et al., 2014)

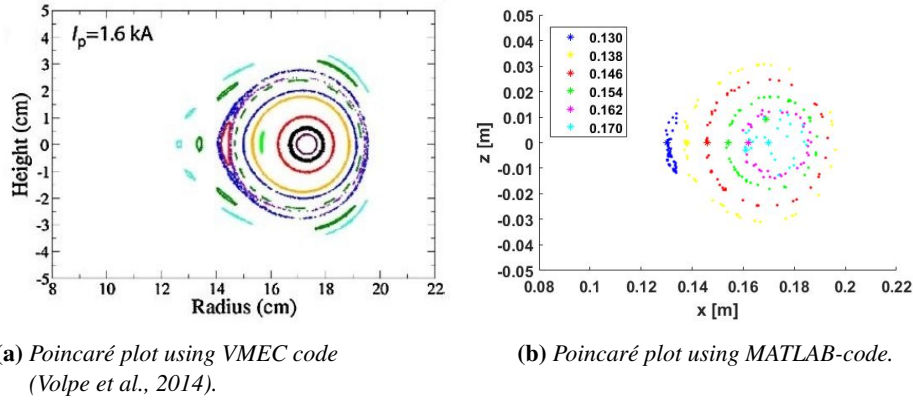
On basis of the values in table 2.2 Poincaré plots have been created previously (Volpe et al., 2014) using a Variational Moments Equilibrium Code (VMEC). In these simulations the plasma current is increased from 800[A] to 1600[A] and 2500[A]. They are displayed in figures 2.3a, 2.4a and 2.5a, respectively.



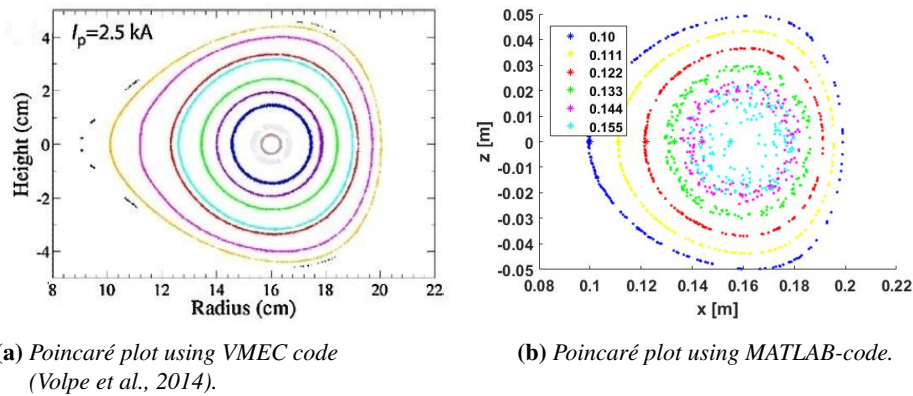
(a) Poincaré plot using VMEC code (Volpe et al., 2014).

(b) Poincaré plot using MATLAB-code.

**Figure 2.3:** Comparison six TF coils CIRCUS MATLAB-simulation versus VMEC at  $I_p = 0.8[kA]$ .



**Figure 2.4:** Comparison six TF coils CIRCUS MATLAB-simulation versus VMEC at  $I_p = 1.6[kA]$ .



**Figure 2.5:** Comparison six TF coils CIRCUS MATLAB-simulation versus VMEC at  $I_p = 2.5[kA]$ .

In the figures 2.3b, 2.4b and 2.5b Poincaré plots using the MATLAB-code are shown with the same initial values as Volpe. These figures have been put side to side with the Poincaré plots of the VMEC code. The legends in the figures display the x-coordinates of the starting locations of the flux lines. The y and z-coordinates are both zero in all points.

These figures show that when the plasma current is increased, the volume increases as well. It can also be noticed that the major radius decreases with an increasing current.

In the figures 2.3a and 2.3b the major and minor radius,  $R_0 = 0.174[m]$  and  $a = 0.01[m]$ , respectively, of both Poincaré plots are similar. Closer to the center of the plasma a difference in accuracy is noticeable. The reason for this may be the numerical nature of the MATLAB-code. When simulating the flux lines a time

step of  $0.001[s]$  is used. Closer to the generated poloidal current the magnetic field is higher and therefore the step the flux line makes in one time step is larger. This creates higher inaccuracies at higher speeds. In order to increase the accuracy the time step may be reduced, but this increases the computation time significantly.

When comparing figures 2.4a and 2.4b the same major radius and minor radius is found for this plasma current, namely  $R_0 = 0.174[m]$  and  $a = 0.03[m]$ . Both figures also show the formation of an island around  $x = 0.13$ .

In figures 2.5a and 2.5b the Poincaré plots also show similar flux surfaces with a major and minor radius of  $R_0 = 0.16[m]$  and  $a = 0.05[m]$  respectively. Similar flux surfaces in the MATLAB-code plot and VMEC plot can be found. The center of the MATLAB-code plot however becomes less accurate in the center compared to the previous figures. A reason for this could be the higher plasma current which creates a higher magnetic field. This causes that the error, due to the time step used in the streamline, is becoming larger.

The similarities in results, such as the major and minor radius, of the VMEC and the MATLAB-code simulations confirm that the MATLAB simulation is a good approach. The differences can be explained by the lower numerical integration resolution the MATLAB-code uses. The MATLAB-code, however, is considerably faster.

### 3 CIRCUS magnetic field

In this chapter it will be investigated if it is possible to confine a plasma in CIRCUS using only the coils. To achieve this an investigation will be performed if the flux lines form closed flux surfaces. Plasma effects such as pressure, drifts and collisions have not been taken into account.

The magnetic field generated by CIRCUS will have to satisfy the conditions given in the research breakdown. Therefore, first the appropriate currents for the coils have to be calculated and simulated.

In this chapter no plasma current or additional poloidal field is assumed, which implies that CIRCUS will have to operate in a stellarator like mode. This is executed by first determining the toroidal field and subsequently the corresponding vertical field, using the MATLAB-code created in the previous chapter. Additionally the contribution of the quadrupole coils will be investigated. In these studies the coil architecture described in error 2.1 will be used.

#### 3.1 Determining toroidal magnetic field

A toroidal magnetic field in a magnetic confinement device, such as CIRCUS, creates the field that confines the plasma and gives it stability. Increasing the strength of the toroidal field allows for a higher plasma pressure, which results in a higher rate of the fusion process. The toroidal magnetic field for CIRCUS however is limited by two factors.

The first is heat. In order to increase the toroidal magnetic field more current has to run through the copper coils, which generates heat. Too much heat will cause that the device will not work as intended. This will be further studied in chapter 6.1.

The second, more crucial, factor for the toroidal magnetic field is that along the magnetic axis a magnetic field strength of  $0.0875[T]$  is required. This value corresponds with the electron cyclotron resonance frequency associated with commercially available devices, which is  $2.45[GHz]$ . An electron cyclotron device such as this one will be used to heat and start up the plasma. This subject will be further discussed in section 5.3.

##### 3.1.1 Calculating toroidal magnetic field

The strength of the toroidal magnetic field can be calculated using Ampère's law, 3.1.

$$\nabla \times B = \mu_0 j \quad (3.1)$$

In the equation above  $B$  stands for the magnetic flux density in *Tesla*, while  $\mu_0$  is the magnetic permeability of vacuum, while  $j$  stands for the current density in  $A/m^2$ .

$$\oint_S v \cdot dx = \int \int_S (\nabla \times v) d^2x \quad (3.2)$$

Integrating equation 3.1 over the surface and applying Stokes Theorem, equation 3.2, we get equation 3.3. This formula can be used, since the total geometry of the coils is comparable to a toroidal solenoid. This calculation will be an approximation, because the interlinking geometry of the coils change the precise outcome.

$$\oint B \cdot dl = B(2\pi R_0) = \mu_0 \cdot N \cdot I_w \quad (3.3)$$

In equation 3.3 is  $R$  the radius of the magnetic field path. The quantity  $I_w$  is the current in one winding of the coil in  $[A]$ , while  $N$  stands for the total number of turns the wire makes around the magnetic path, which can be divided into the number of coils and the number of windings per coil. Equation 3.4 is the final resulting equation with which the toroidal magnetic field can be calculated.

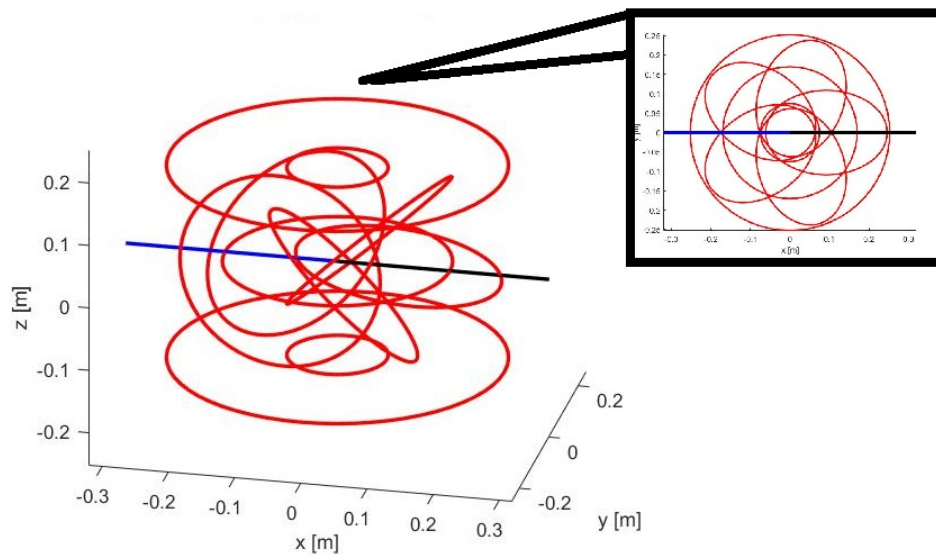
$$B_t = \frac{N_{coils} \cdot N_{turns} \cdot \mu_0 \cdot I_{TF}}{2 \cdot \pi \cdot R_0} \quad (3.4)$$

In this equation  $N_{coils}$  stands for the number of toroidal field coils in CIRCUS.  $N_{turns}$  is the amount of turns the wire makes in one coil, which is 69 for the TF coils, as shown in table 2.1.  $I_{TF}$  is the current in the wire of the coil.  $R_0$  is the major radius of the plasma, which is the distance from the center of the plasma to the center of the torus, equalling to  $0.16[m]$ . Filling in these values for the 5 TF coil CIRCUS gives us a TF coil wire current value of  $202.9[A]$ . This value will be rounded to  $200[A]$  for the rest of the report.

### 3.1.2 Simulating toroidal magnetic field

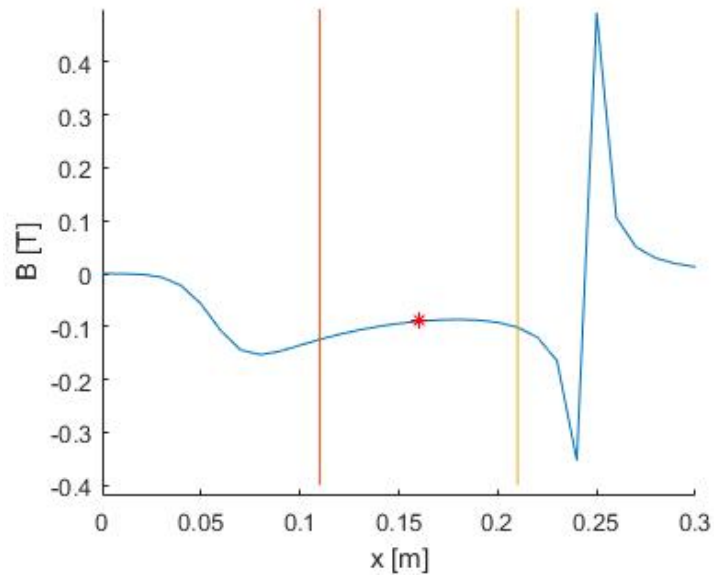
To confirm the findings above the previously created code of subsection 2.2.2 will be used. In this model the current value for the toroidal field coils of  $200[A]$ , gained from formula 3.4, will be filled in. This means that a total current of  $13.8[kA]$  is used for the generation of the magnetic field in each TF coil, since the coils have 69 windings. The resulting strength of the toroidal magnetic field is shown in a radially dependant plot along the x-axis. The axis, along which the toroidal magnetic field is simulated, is also presented in figure 3.1.





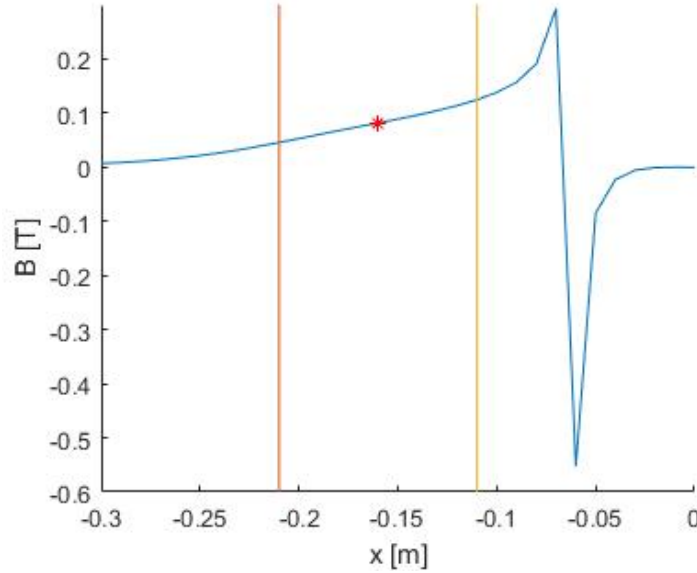
**Figure 3.1:** *Isometric and top view of CIRCUS with the negative and positive x-axis shown in blue and black respectively.*

Figure 3.1 is an isometric and top view of CIRCUS where the coils are indicated in red. As can be observed the black line, which is the positive x-axis, goes straight through the center of a toroidal field coil, while the blue line, the negative x-axis, goes right in between two coils. A toroidal magnetic field plot will go along the black line and the other along the blue line. Figure 3.2 is the plot of the toroidal magnetic field strength along the positive x-axis, straight through a TF coil.



**Figure 3.2:** The blue line indicates the radially dependant toroidal magnetic field strength on the positive x-axis. The orange and yellow lines indicate the area in which the plasma is expected (MATLAB-code simulation).

In this figure the red point is on the major radius of the device. Here a toroidal magnetic field value of  $-0.09[T]$  can be concluded. This value is negative since the toroidal field has an clockwise direction as seen in figure 3.1. Figure 3.3 is a plot of the toroidal magnetic field strength along the negative x-axis, which goes straight between two TF coils.



**Figure 3.3:** The blue line indicates the radially dependant toroidal magnetic field strength on the negative  $x$ -axis. The orange and yellow lines indicate the area in which the plasma is expected (MATLAB-code simulation).

In figure 3.3 a toroidal magnetic field value of  $0.082[T]$  can be found at the major radius. This value is positive because the toroidal magnetic field is clockwise, as mentioned before.

In conclusion the MATLAB-code shows that along the major radius of the plasma, which is  $0.16[m]$ , a maximal toroidal magnetic field strength of  $0.09[T]$  and a minimal of  $0.082[T]$  is reached, using a TF current of  $200[A]$ . This implies that the value of  $0.0875[T]$  is somewhere along the major radius, as initially required, meaning that the simulation results comply with what was calculated using equation 3.4. This current value for the TF current allows for electron cyclotron resonance near the core of the plasma.

In the next section the vertical magnetic field will be studied.

### 3.2 Determining vertical magnetic field

The vertical magnetic field is generated by a set of four coils: an inner and an outer on both, the top and the bottom of the device, which is shown in figure 2.1. The vertical field coils are the main generators of the vertical field, while the quadrupole field coils, whose effect will be studied in section 3.3, shape the field.

The main reason for the vertical field in CIRCUS is to counterbalance the vertical field the toroidal field coils produce. The angle of the TF coils creates a magnetic field that contributes to the toroidal field and to the vertical magnetic field, where the strength is dependant on the angle of the coils, which is  $\alpha = 45^\circ$ .

### 3.2.1 Calculating vertical magnetic field

Below a back-of-an-envelope calculation will be performed for determining the expected vertical field along the major radius.

The magnetic field in the center of a coil can be calculated applying formula 3.5. Here  $B$  is the magnetic field at the center of the loop.  $I$  is the current in [A], while  $N_{turns}$  is the amount of turns in the coil.  $\mu_0$  is the magnetic permeability in vacuum, while  $R_{coil}$  is the radius of the coil.

$$B = \frac{I \cdot N_{turns} \cdot \mu_0}{2 \cdot R_{coil}} \quad (3.5)$$

In CIRCUS the coils are placed on an angle, causing that only a part of the generated magnetic field contributes to the vertical field. This is taken into account in formula 3.6 by the term  $\alpha$ , which is the angle of attack.

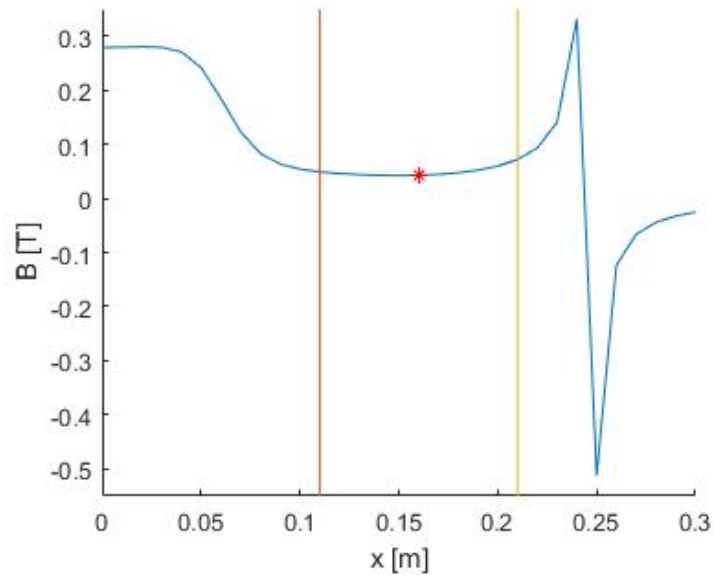
$$B_z = \cos(\alpha) \cdot \frac{I \cdot N_{turns} \cdot \mu_0}{2 \cdot R_{coil}} \quad (3.6)$$

For the calculation of the vertical field generated by the TF coils a current of 200[A] is used, as gained in the previous section. The coil constants such as the radius can be found in table 2.1. As indicated before an angle of attack of  $45^\circ$  has been applied. This calculation results in a vertical magnetic field generated by the TF coils of 0.0398[T]. In reality however the coil center is not exactly on the major radius. Additionally the coil architecture of CIRCUS makes the expected vertical field not constant along the central axis of the plasma.

### 3.2.2 Simulating vertical magnetic field

In order to have no vertical field along the center axis of the plasma, the vertical contribution of the magnetic field generated by the TF coils will be simulated using the MATLAB-code.

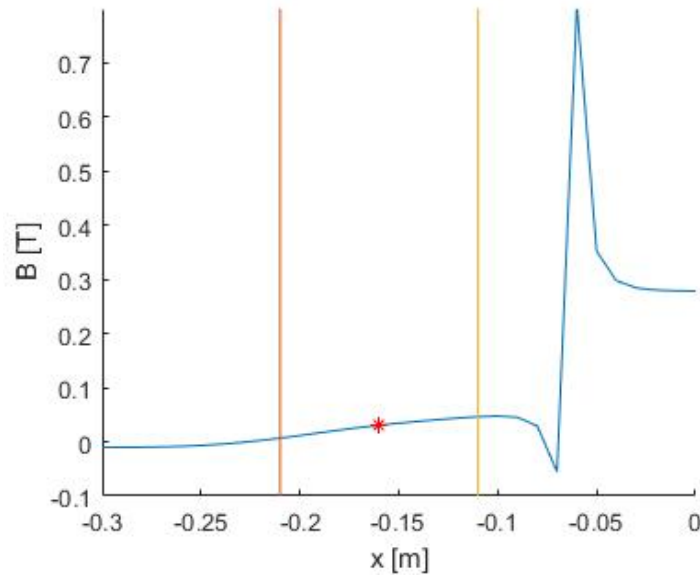
The vertical field along the positive x-axis, which goes right through the center of a TF coil, is illustrated in figure 3.4.



**Figure 3.4:** The blue line indicates the radially dependant vertical magnetic field strength on the positive  $x$ -axis, with no current for VF and QF. The orange and yellow lines indicate the area in which the plasma is expected (MATLAB-code simulation).

The figure above demonstrates that the vertical field is positive, as it has an upward direction, due to the angle of attack of the coils. In this figure there is no current in the VF and QF coil. The value of the vertical field on the major radius is  $0.043[T]$ . This is just a bit higher than the calculated  $0.0398[T]$  before, which is as expected, since the major radius is not in the center of the coil but close to the side.

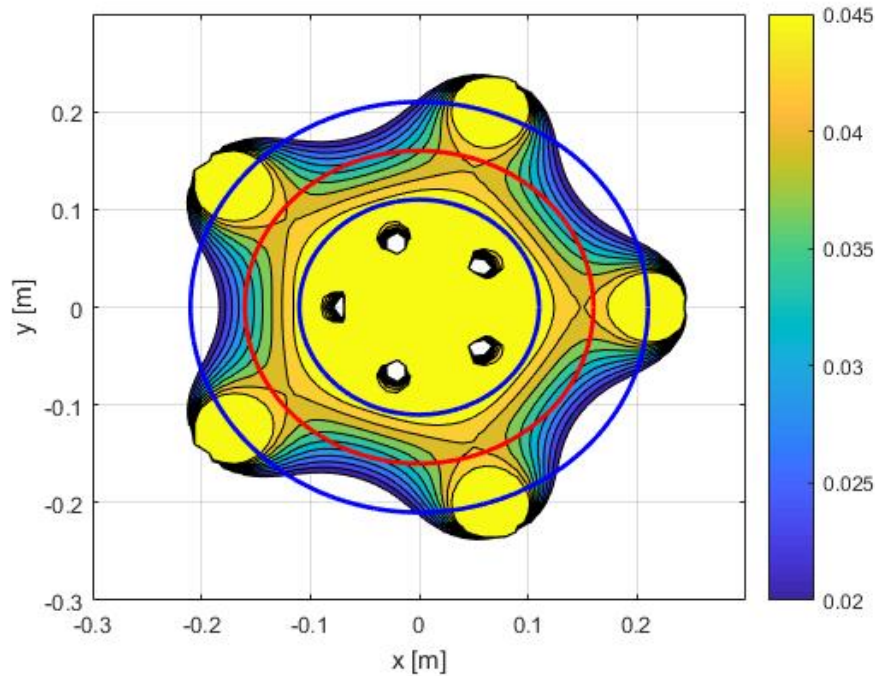
Figure 3.5 is the radially dependant magnetic field plot along the negative  $x$ -axis, which goes right between two TF coils.



**Figure 3.5:** The blue line indicates the radially dependant vertical magnetic field strength on the negative  $x$ -axis, with no current for VF and QF. The orange and yellow lines indicate the area in which the plasma is expected (MATLAB-code simulation).

In the figure above the value of the vertical field on the major radius is  $0.031[T]$ . This is lower than the previous value, since this part is further from the center of the TF coil.

From this it can be concluded that the vertical contribution of the TF coils fluctuates along the central axis of the plasma. This is further highlighted in figure 3.6, which is a top view of a MATLAB contour simulation.

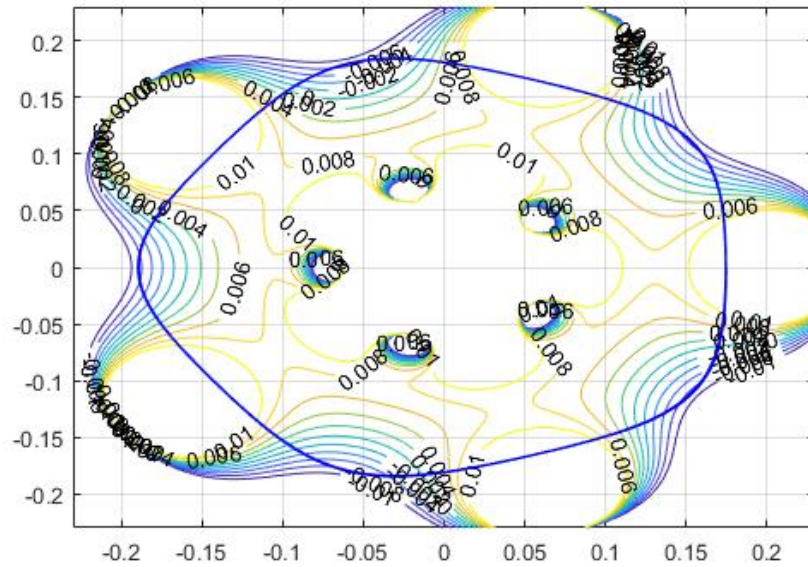


**Figure 3.6:** Top view of a contour plot of the vertical contribution of the toroidal magnetic field in [T] (MATLAB-code simulation).

In the figure above the vertical field generated by the TF coils is shown by the contour plot. The red circle is the major radius, while the blue circles are the inner and outer radius of the plasma. As can be noticed the vertical field reaches values between  $0.043[T]$  and  $0.031[T]$  on the major radius, as shown earlier in figure 3.4 and 3.5. Using this contour plot the shape of the vertical field can be investigated.

### 3.2.3 Reducing the vertical magnetic field

The objective is to reduce the vertical field contribution along the center line of the plasma to zero. The center line of the plasma however changes radially. A streamline along the center of the plasma is drawn in figure 3.6 in blue.

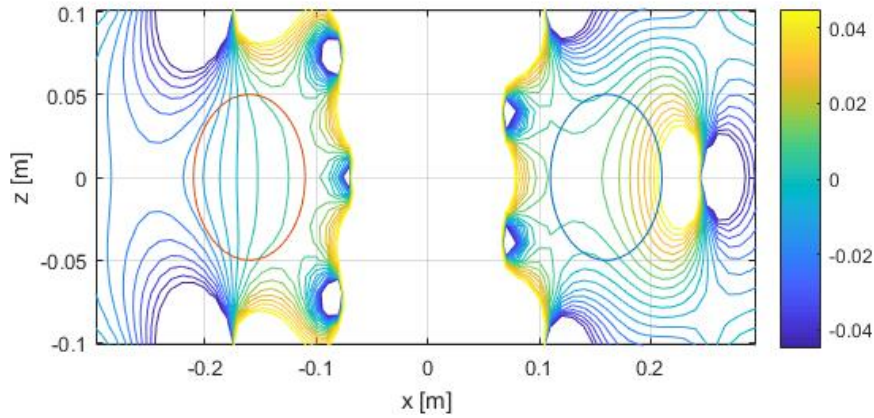


**Figure 3.7:** Top view of the vertical field generated by the TF, VF and QF coils in [T] and streamline plot of plasma center (MATLAB-code simulation).

Simulations show that when a current of  $205[A]$  is used in the VF and QF coils the total vertical magnetic field along the center line of the plasma, indicated above in blue, is reduced to zero. In the figure above a contour plot of the generated vertical field by these new values for the VF and QF coils has been overlapped.

Figure 3.8 shows a contour plot of the magnetic field in the z-x plane. In this figure circles with a radius of  $0.05[m]$  have been drawn at a major radius of  $R_0 = 0.16[m]$ . Here you can see that using a current of  $205[A]$  in both coils results in vertical contour lines for the vertical field, indicating that the vertical field is stable in the z-direction.

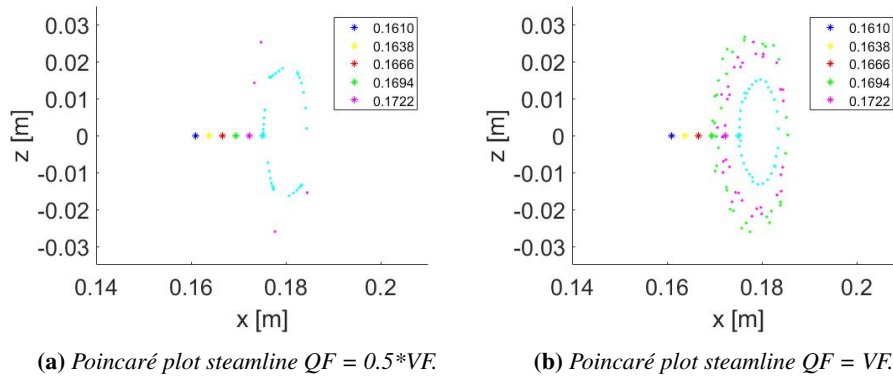




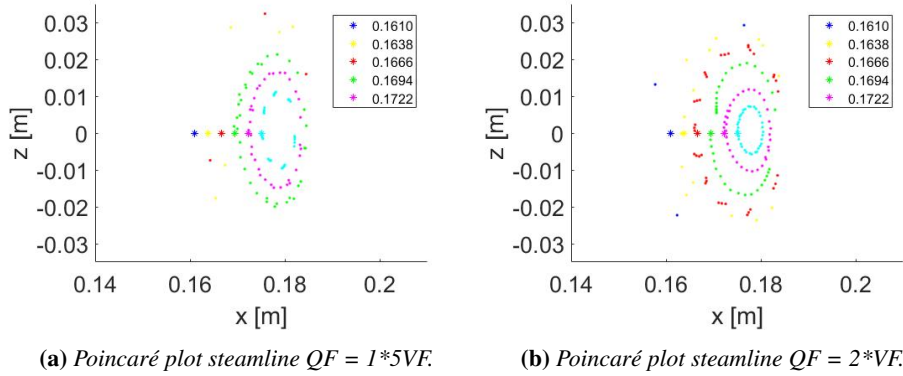
**Figure 3.8:** Side view in  $x$ - $z$  plane of MATLAB simulation of vertical field magnetic field generated by TF, VF and QF coils in [T] (MATLAB-code simulation).

### 3.3 Quadropole coils contribution

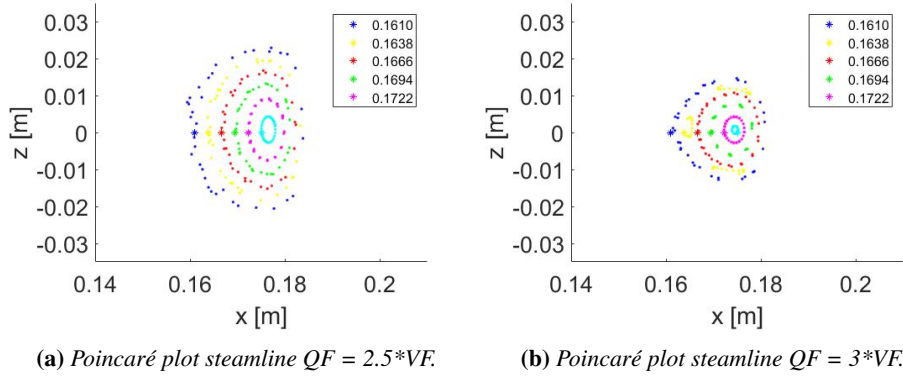
In this section the contribution of the quadropole coils to the plasma's shape will be investigated. In order to achieve this the current in the QF coils will be varied in four steps from 0 to 1.5 times the size of the current in the VF coil. In the previous section the same current has been used for the QF coil as for the VF coil. Now for each scenario a Poincaré plot of the flux lines through the positive  $x$ - $z$  plane will be created.



**Figure 3.9:** Comparison QF contribution using Poincaré.



**Figure 3.10:** Comparison  $QF$  contribution using Poincaré.



**Figure 3.11:** Comparison  $QF$  contribution using Poincaré.

In the figures above the origin points of the streamlines are indicated by star shaped marks. These are the same in all figures. It appears that a coil current of 2.5 times the current used in the VF coil gives the best results, because increasing or decreasing the current in the QF coils from this point decreases the size of the last closed flux surface.

An earlier performed study in optimising coil currents confirms this result. In this study a ratio of  $\frac{I_{QF}}{I_{TF}} = 2.72$  was determined to be the optimal solution for a maximum volume (Smits, 2021). This is close to the results gained from figure 3.11.

From this figure it can be concluded that the QF coils contribute to the stability and volume of the magnetic confinement. This new value for the QF coils is used for the rest of this study.

At the moment there is another code to compute the flux lines of CIRCUS called CIRCA, which makes use of Python. This code can show thinner flux lines, however it is not as fast.

### 3.4 Complete magnetic field

The desired values for the current in the coils to operate CIRCUS with five coils have been put in table 3.1. Here the location and the resistance of the coils are displayed too.

Coil	$R[cm]$	$Z_{coil}[cm]$	$R_{coil}[cm]$	$I[A]$	$N_{coils}[-]$	$N_{turns}[-]$	$R[m\Omega]$	$V[V]$
TF	15.4	0	9.24	200	5	69	297	59.4
VF	25.25	15.4	0	205	2	54	177	23
QF	7.48	15	0	513	2	56	95	12.4

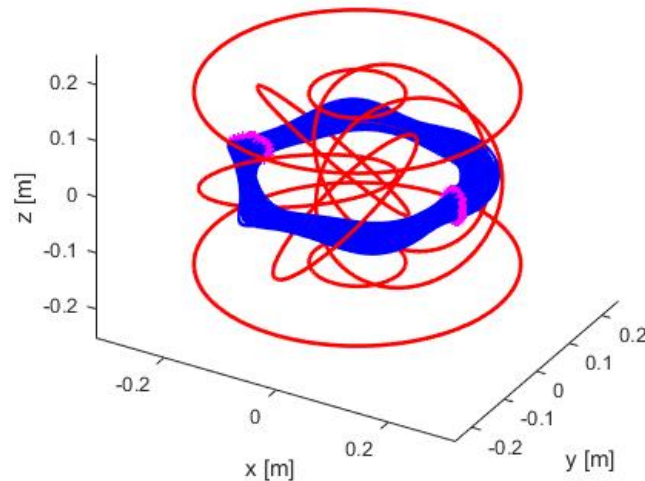
**Table 3.1:** Calculated coil settings for 5 TF coil CIRCUS.

Table 3.1 leads to the ratios between the coil currents displayed in equations 3.7 and 3.8.

$$\frac{I_{VF}}{I_{TF}} = 1.025 \quad (3.7)$$

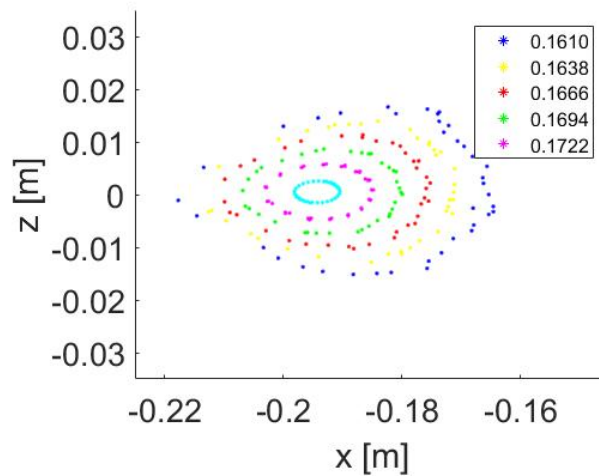
$$\frac{I_{QF}}{I_{TF}} = 2.565 \quad (3.8)$$

In the table above the values for the resistance are measured on the device itself. By applying these values a magnetic field can be simulated in MATLAB with closed flux surfaces. This is illustrated in figure 3.12.



**Figure 3.12:** Isometric view of magnetic field line simulation MATLAB-code.

The figure above is a three dimensional streamline plot of the magnetic field in CIRCUS. In this representation the red lines indicate the coils, while the blue lines are the streamlines. On the points where the streamline crosses the z-x plane purple marks have been placed. As can be noticed using a current of  $200[A]$  in the toroidal field coil and a current of  $205[A]$  in the vertical and  $513[A]$  in the quadrupole field coils gives the maximum volume and a magnetic field with closed flux surfaces. In order to investigate the magnetic field further the Poincaré plots of figure 3.11a and figure 3.13 can be used.



**Figure 3.13:** Poincaré plot of streamline on the negative x-z plane (MATLAB-code simulation).

The last closed flux surface is determined by varying the origin points and observing when the flux surface is no longer closed. The figure shows that with the applied values of table 3.1 the major radius for CIRCUS is not the expected  $16[cm]$  but  $19.5[cm]$  on the negative x-z plane. This value varies when looking at the plasma radially, as can be seen when comparing figures 3.11a and 3.13. On the positive x-z plane a minor radius of  $17.5[cm]$  is observed. Therefore, an average major radius of  $18.5[cm]$  is taken for the rest of the report. Further, it can be noticed that the minor radius also changes radially and that it varies between  $2[cm]$  and  $2.5[cm]$ . In this figure different flux surfaces are also clearly identified. The next section investigates if this plasma volume is large enough to confine the electrons and ions.

### 3.5 Particle gyration

If a charged particle moves in a magnetic field it will experience a Lorentz force, causing that the particle starts to gyrate. In this section the gyration radius of the electron will be calculated. If this radius is larger than the minor radius of the plasma it can not be contained.

The gyration radius can be calculated using equation 3.9.

$$r_{gyro} = \frac{mv_{\perp}}{|q|B} = \frac{\sqrt{2mkT}}{qB} \quad (3.9)$$

In this equation  $r_{gyro}$  is the gyroradius,  $m$  is the mass of the particle and  $v_{\perp}$  is the speed of the particle perpendicular to the magnetic field.  $B$  stands for the strength of the magnetic field and  $q$  is the electric charge of the particle. The speed perpendicular to the magnetic field,  $v_{\perp}$ , can be calculated using the electron temperature in Kelvin,  $T$ , and the Boltzmann's constant,  $k$ . Their values are listed in table 3.2, applying the assumptions of table 1.1.

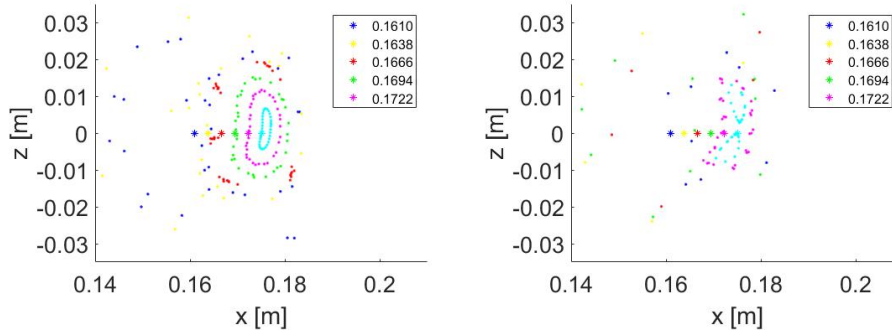
	$m[kg]$	$q[C]$	$B[T]$	$T[eV]$	$r_{gyro}[mm]$
Electron ( $e$ )	$9.11 * 10^{-31}$	$-1.6 * 10^{-19}$	0.0875	1[eV]	$3.85 \cdot 10^{-2}$
Deuterium ion ( $D^+$ )	$3.321 * 10^{-27}$	$1.6 * 10^{-19}$	0.0875	1[eV]	2.3
Tritium ion ( $T^+$ )	$4.982 * 10^{-27}$	$1.6 * 10^{-19}$	0.0875	1[eV]	2.9
Nitrogen ion ( $N^+$ )	$2.324 * 10^{-26}$	$1.6 * 10^{-19}$	0.0875	1[eV]	6.2
Nitrogen ion ( $N_2^+$ )	$4.648 * 10^{-26}$	$1.6 * 10^{-19}$	0.0875	1[eV]	8.7

**Table 3.2:** Gyration radius of ions and electrons along major radius.

With the gyration radius shown in the table above, the electron stays inside the plasma. The Nitrogen ions used in this experiment however do not, since the gyration radius is too close to that of the minor radius of the plasma. The next section will investigate the sensitivity of CIRCUS to misalignments in current and coil locations.

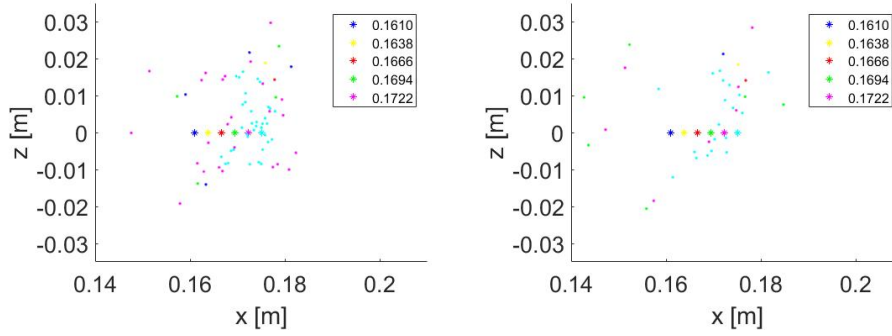
### 3.6 Sensitivity coil parameters

As stated in the research problem small misalignments or current differences in the coils can create large irregularities in the magnetic field lines, which are undesirable. To simulate such misalignments a location and current error in the coils will be introduced in the MATLAB code. In the next figures Poincaré plots of the streamlines are shown with an increasing error.



(a) Error of 0.2% in current and coil locations. (b) Error of 0.5% in current and coil locations.

**Figure 3.14:** Comparison sensitivity of magnetic field using Poincaré plots.



(a) Error of 0.7% in current and coil locations. (b) Error of 1% in current and coil locations.

**Figure 3.15:** Comparison sensitivity of magnetic field using Poincaré plots.

From figures 3.14 and 3.15 we learn that increasing the error decreases the confined volume. At an error of 1% the desired confinement is lost, which is compatible with an earlier study about misalignment sensitivity of CIRCUS (van Rooij et al., 2017).

A few sample misalignment values for the simulation performed in figure 3.15b are shown in table 3.3.

Coil parameter	Value	Error (1%)
$x_{TF}$	92.4 [mm]	0.92 [mm]
Coil angle	180°	1.8°
$I_{TF}$	200 [A]	2 [A]
$I_{VF}$	205 [A]	2.05 [A]
$I_{QF}$	513 [A]	5.13 [A]

**Table 3.3:** Errors of 1% for sensitivity simulation.

In this table  $x_{TF}$  represents the x-coordinate of the center of a TF coil. A location and current error of 1% is not an unlikely scenario, as it was not possible to reach high precision during the construction of CIRCUS. This means that a location error for the coils of around 1 millimeter and an angle error for these coils of around  $2^\circ$  can be expected. Therefore, it can be concluded that CIRCUS can not confine a plasma if it solely makes use of the existing coil architecture, since the magnetic architecture is too sensitive to disturbances.

### 3.7 Conclusions

In this chapter the magnetic field required to confine a plasma in CIRCUS with only the coils has been investigated. Back-of-the-envelope calculations and MATLAB-code simulations are used to determine which values for the currents are required to achieve this. For the TF coils this is  $200[A]$ , for the VF  $205[A]$  and QF coils  $513[A]$ . At these values the center axis of the plasma has a toroidal magnetic field strength of  $0.0875[T]$ , while the total vertical field along this axis is  $0[T]$ . The current in the coils can be changed; as long as the ratio between them stays the same, the resulting Poincaré plots will stay the same too.

The ratios for the current between the coils, equations 3.7 and 3.8, are similar to what was proposed in a former study (Smits, 2021). The results from Volpe are also similar, except for the QF current. In this study the calculated current values are higher than the proposed values in Volpe's paper (Volpe et al., 2014).

A Poincaré plot of the flux lines intersecting the x-z plane (figures 3.11a and 3.13) leads to the conclusion that the average major radius of the plasma is  $18.5[cm]$  and the minor radius is around  $2 - 2.5[cm]$  for the proposed coil currents. This is also different from the values of Volpe (Volpe et al., 2014), which can be found in the appendix B. Volpe proposes a major radius of  $16.0[cm]$ . The reason for this difference could be the high plasma current,  $I_p = 2.5[kA]$ , he assumes in his calculations.

The minor radius of the plasma is large enough to confine electrons but not large enough to confine the ions. Increasing the plasma volume by increasing the current could be a way to rectify this.

Figure 3.14 shows that it is possible to have closed flux surfaces if the device is constructed perfectly. But increasing the location errors of the coils and the current decreases the confined area. When an error of 1 percent is reached, all confinement is lost. A 1 percent error in location and angle of the coils is not unlikely. This implies that CIRCUS cannot operate purely as a stellarator. Adding an additional poloidal magnetic field may remedy this.

## 4 CIRCUS poloidal magnetic field

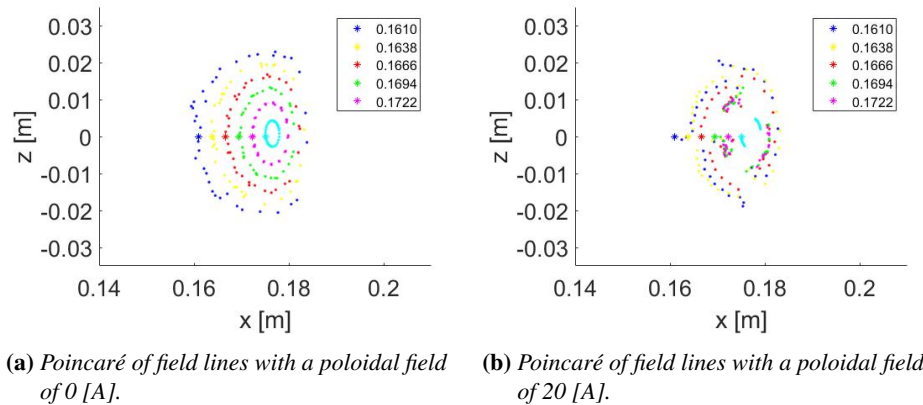
The previous chapter concluded that it is not possible to operate CIRCUS purely as a stellarator, where only the magnetic fields of the coils are enough to shape the plasma, unless an extremely high accuracy in construction is reached. Subsequently, CIRCUS must be operated as a hybrid tokamak-stellarator.

Therefore in this chapter we first look at the size of an additional generated poloidal field. Next the expectation that increasing the poloidal field increases the plasma volume will be investigated. A larger plasma volume may allow for the ions to be confined. Further a new sensitivity analysis will be performed to find out if the additional poloidal current increases the stability of the magnetic architecture of CIRCUS.

### 4.1 MATLAB simulation increasing poloidal field

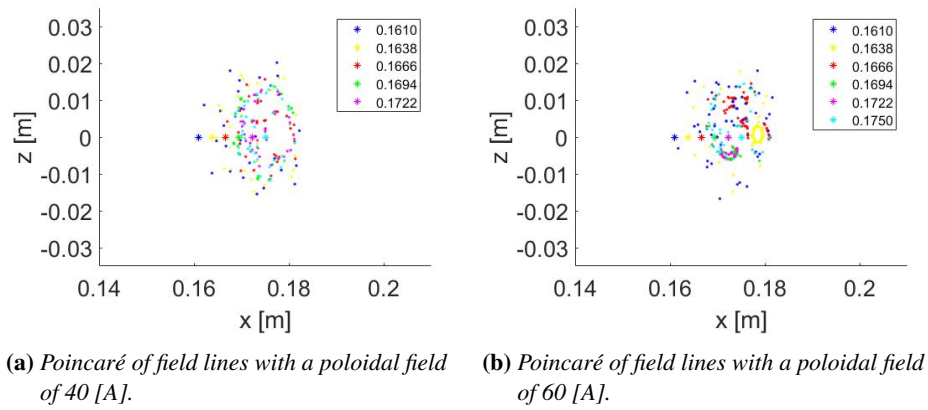
For studying the magnetic field architecture of CIRCUS the MATLAB code described in chapter 2 will be used. Also the coil currents and locations included in table 3.1 will be applied. This results in Poincaré plots of the magnetic field lines enabling a look at the flux surfaces and plasma volume.

In the previous chapter it was concluded that the average major radius is at  $18.5[cm]$  when there is no plasma current. Along this radius a plasma current will be generated in the simulations below. The current generating the poloidal magnetic field will be increased step by step from  $0[A]$  to  $140[A]$ . This is the highest current at which this major radius can be used. For higher currents the major radius must be decreased. The findings of this analysis are displayed in the following figures.

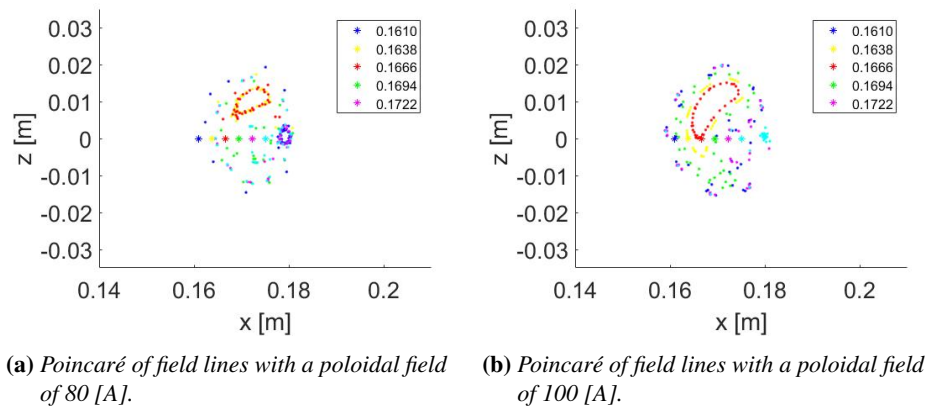


**Figure 4.1:** MATLAB-code simulation five coils.

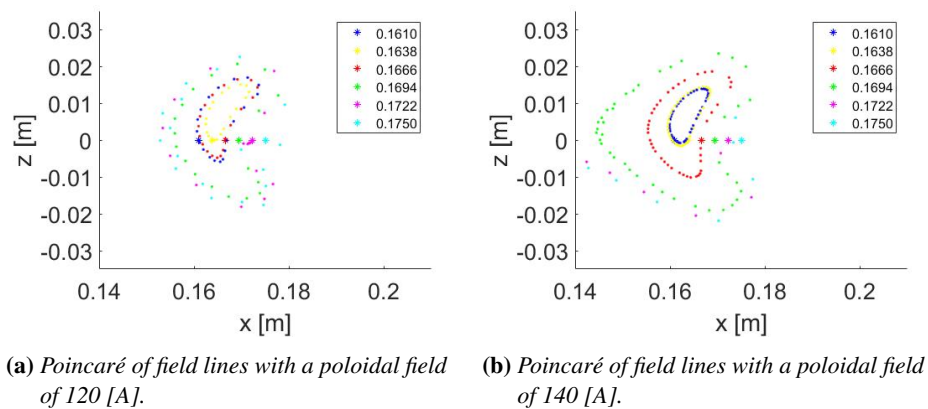




**Figure 4.2:** MATLAB-code simulation five coils.



**Figure 4.3:** MATLAB-code simulation five coils.



**Figure 4.4:** MATLAB-code simulation five coils.

The figures above suggest that the initial flux surfaces are lost when a current is introduced. The reason for this is that the simulated current does not go straight through the center of the plasma in all points. As concluded in the previous chapter the major radius of the plasma changes radially due to the coil architecture. At currents below 80[A] this causes undesirable results, since the plasma current is not high enough. In figures 4.4a and 4.4b it can be seen that there are multiple flux surfaces, but the center is not at  $R_0 = 18.5[cm]$ . Some flux lines however may concentrate around the generated current at the radius of 18.5[cm] such as figures 4.3a and 4.3b suggest.

A current of 140[A] gives the best results, since the minor radius is largest for this result. At this current the minor radius however has not increased enough to confine the ions also. An even larger plasma volume is required for this.

To achieve this the poloidal current must be increased, which in turn requires a decreased major radius. When high currents such as 800[A], 1600[A] and 2500[A] are simulated, similar results to the figures 2.3b, 2.4b and 2.5b are gained, respectively. Using these results a rough relation between the plasma current and the major radius can be estimated, at which results may be gained. This is shown in equation 4.1.

$$R_0 \approx 0.185 - I_p \cdot 10^{-5} \quad \text{for } 0 < I_p < 2500 \quad (4.1)$$

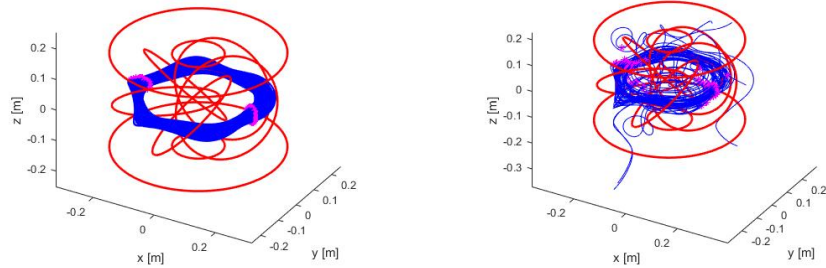
These higher currents generate larger plasma volumes, which may confine the ions. The assumed major radius however will have to be changed. For the rest of the report the major radius will be kept at  $R_0 = 0.185$ , because this was calculated to be the average major radius when there was no current. This allows for low plasma currents, which are easier to generate.

For the calculations in the rest of the report a current of 100[A] will be used. This is the first current at which a clear closed flux surface is observed. The question now before us is whether a current of 100[A] is enough to sufficiently decrease the sensitivity of CIRCUS.

## 4.2 Sensitivity investigation of magnetic field

Section 3.6 concluded that if an error in the coils was introduced the field lines follow no longer the intended path. In this section the MATLAB-code will be applied to simulate the magnetic field of CIRCUS with five toroidal field coils with a poloidal magnetic field. The effect of the additional poloidal field on the sensitivity of the device can then be compared. Here three dimensional plots of the magnetic field lines will be used

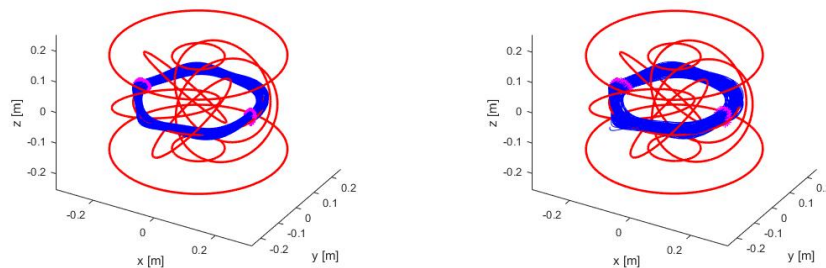
Figure 4.5a displays a MATLAB simulation where there is no plasma current and no construction error, demonstrating that the field lines follow the intended path. Now if an error in the construction of 1%, according to table 3.3, is introduced the field lines are no longer closed. This can be observed in figure 4.5b.



(a) Isometric view of magnetic field line simulation, with a plasma current of 0 [A] and no error in construction and coil current. (b) Isometric view of magnetic field line simulation, with a plasma current of 0 [A] and a 1% error in construction and coil current.

**Figure 4.5:** Magnetic field line simulation MATLAB-code.

Figure 4.5 shows that small misalignments in the coils have a drastic effect on the magnetic architecture. The introduction of an additional poloidal field should increase the stability of the magnetic architecture. Therefore, in the next simulation the selected plasma current of 100 Ampère's is used. Figure 4.6a is the field line simulation without an error, while in figure 4.6b an error of 1% in the location and current of the coils has been introduced.



(a) Isometric view of magnetic field line simulation, with a plasma current of 100 [A] and no error in construction and coil current. (b) Isometric view of magnetic field line simulation, with a plasma current of 100 [A] and a 1% error in construction and coil current.

**Figure 4.6:** Magnetic field line simulation MATLAB-code.

Figure 4.6b demonstrates that the field lines are now closed again. However, due to the misalignments the plasma is no longer symmetrical. This leads to the conclusion that increasing the poloidal magnetic field by introducing a current will decrease the misalignment sensitivity and close opened field lines. This is confirmed by the results of an earlier study (De Rooij, 2019). For the rest of this study a plasma current of 100 [A] is assumed.

### 4.3 Conclusions

Introducing a plasma current along a circular radius disturbs the existing shape of the flux surfaces. The reason for this is that the center of the plasma does not follow a perfectly circular path, but the path shown in figure 3.7.

Increasing the plasma current increases the plasma volume and decreases the minor radius. A plasma current of  $100[A]$  has been chosen since it is the first where clear flux surfaces are visible and it is the easiest to generate.

When  $100[A]$  is used to generate a poloidal magnetic field the minor radius is around  $a = 0.02[m]$  and the major radius is  $R_0 = 0.185[m]$ . This minor radius is large enough for gyrating electrons, but not ions to be confined in the plasma with the existing magnetic field strength.

A plasma current of  $100[A]$  is enough to generate the desired magnetic architecture where all field lines are closed and the sensitivity of the device is reduced sufficiently.

## 5 Poloidal magnetic field generation methods

As found out in the previous chapter CIRCUS must operate in a hybrid tokamak-stellarator mode, where an additional poloidal magnetic field exists.

A plasma current is the usual way for tokamaks to generate such a field. There are multiple ways to induce a plasma current and in this report three methods will be investigated. A fourth method to be studied generates a poloidal magnetic field directly.

The first technique is Neutral beam current drive (NBCD). Thereafter the possibility for CIRCUS to use a central solenoid will be examined. The application of Electron cyclotron current drive (ECCD) will also be investigated and finally generating a poloidal magnetic field by use of a wire is discussed.

NBCD and ECCD inject directly power into the plasma, which results in a plasma current. These two techniques will be investigated in sections 5.1 and 5.3 respectively. For this experiment a [1kW] microwave is available for ECCD.

Another option is a central solenoid, which induces a plasma current by changing the magnetic flux. The current drive is therefore limited by the maximal flux swing. A disadvantage to practising this method is that the device can no longer operate continuously, which will be discussed further in section 5.2.

Instead of using the plasma as a carrier a last option for generating a poloidal magnetic field is a circular wire mounted in the center of the device. This wire can be used to drive current through the center and achieve the sensitivity reduction of the plasma, which is desired. This option is possible since CIRCUS makes use of a low temperature plasma. This method will be further discussed in section 5.4.

As determined in section 4.1 and 4.2 a minimal poloidal current drive of 100[A] is required to reduce the sensitivity. The current drive method for CIRCUS will be selected based on this requirement.

### 5.1 Neutral beam current drive

Neutral beam current drive works by injecting neutral particles into the plasma. These particles have a high energy, which will be deposited in the plasma. This can serve several purposes: NBCD can be used to fuel the plasma, heat the plasma and perform current drive. After the high energy neutral particle has entered the plasma it will ionize. To achieve the most efficient current drive using NBCD, the high-mass ions will have to collide with the electrons and deposit their energy. Therefore, the beam must be injected tangent to the plasma in the direction the current is wished to run (Japan Atomic Energy, 2000).

For NBCD to function the injected neutral particles have to be ionized in the plasma. There are three processes where ionization of a neutral particle in a plasma can occur. The first is charge exchange and the second is when the neutral particle has a collision with an ion in the plasma. The third method is when the neutral particle is ionized by an electron. However, the chance for this to occur is so small that it can be neglected.

Applying equation 5.1 the decay length of a neutral particle can be calculated in a plasma (Freidberg, 2007).

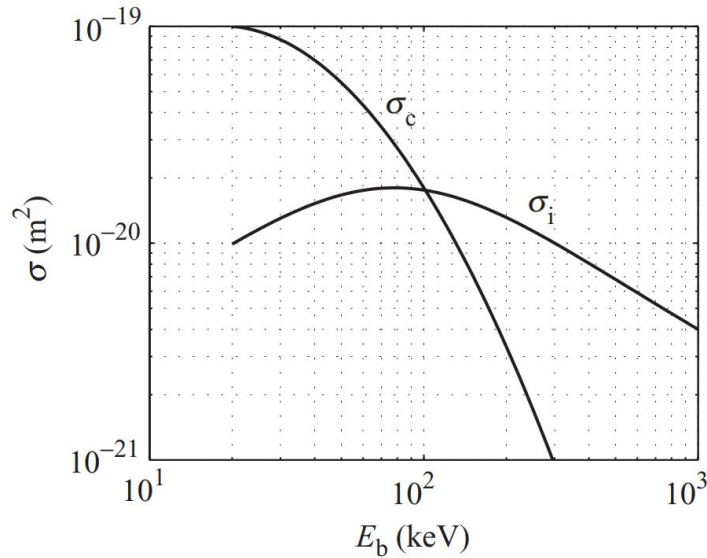
$$\lambda = \frac{1}{n_p(\sigma_c + \sigma_i)} \quad (5.1)$$

In this equation  $\lambda$  is the decay length of the neutral particle and  $n_p$  is the plasma density. The cross-section for the charge exchange and ion collision of the neutral particle are  $\sigma_c$  and  $\sigma_i$  respectively. For the neutral beam to drive current in the center of the plasma the decay length of the neutral particle must be approximately equal to the minor radius of the plasma,  $\lambda \cong a$ , which is around 1[cm] according to section 4.1. For the neutral beam not to shine through the plasma  $\lambda$  must be smaller than the plasma diameter, which is  $2a$ . The values derived from the calculation have been filled in into table 5.1. The plasma density assumption was made in section 1.5.

	$\lambda[m]$	$n_e[10^{20}m^{-3}]$	$\sigma[m^2]$
Value	0.01	0.001	$10^{-15}$

**Table 5.1:** Neutral beam current drive, ionization cross-section calculation.

Table 5.1 shows the minimal required cross-section for ionization. Now figure 5.1 can be used to find out what beam energy,  $E_b$ , is required to achieve this reaction cross-section.



**Figure 5.1:** Cross-sections for charge exchange and ionization (Wesson, 2004).

From the figure above it can be concluded that the cross-section for charge

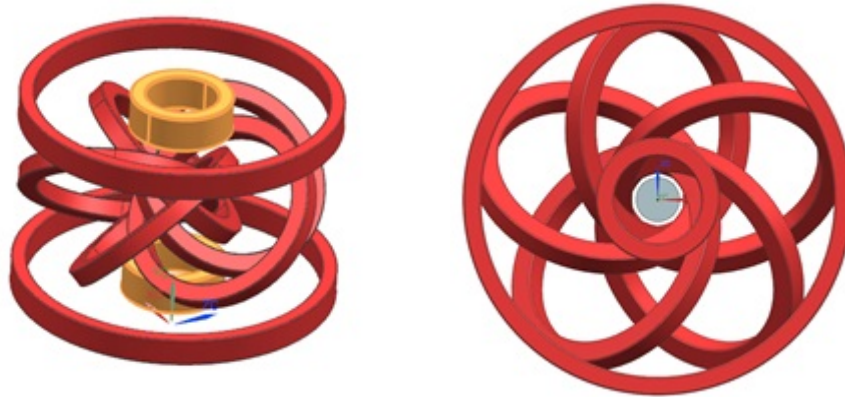
exchange and ion collisions is too low for any beam energy. This means that neutral beam current drive can not be used to drive a current in CIRCUS.

## 5.2 Ohmic current drive

Inductive current generation can be seen as a transformer, which makes use of Lenz's law, stating that a change in magnetic flux gives an induction current that opposes the change in flux. Lenz's law is a consequence of Maxwell's fourth law. The amount of current in the plasma generated due to the change in magnetic flux can be calculated using formula 5.2.

$$V = \frac{-d\Phi_B}{dt} \quad (5.2)$$

Ohmic current drive applies Lenz's law, equation 5.2. By creating a flux swing in the center of the torus a current will be induced in the plasma around it. The direction of the flux swing determines the direction of the current generation. A central solenoid can be placed in the center of the device to generate this flux swing. The available space in the center of the device however, is limited as can be observed in figure 5.2.



(a) Isometric view cad model CIRCUS Magnets. (b) Top view cad model CIRCUS Magnets.

**Figure 5.2:** Cad models CIRCUS.

A cylinder with a maximum radius of  $30[mm]$  would fit in the center of the device. This area could theoretically be used to place a central solenoid. Suppose a superconducting material such as Niobium Tin,  $Nb_3Sn$ , is used to create this coil with a wire diameter  $3mm$ . This wire would have a maximal current of  $500[A]$  at a temperature of  $9^\circ[K]$  (Superconwire, 2021). With these presumptions equation 5.3 can be applied to calculate the generated magnetic field at the axis of the solenoid.

$$B = \frac{\mu \cdot N_{turns} \cdot I}{l} \quad (5.3)$$

In this equation  $N_{turns}$  is the amount of turns of the wire, which is 10, since the wire is  $3[mm]$  thick and in the center of the device the radius of free space is  $30[mm]$ .  $I$  is the current, which is  $500[A]$  and  $l$  is the height of the coil. This is equivalent with the width of the wire,  $3[mm]$ . This results in a magnetic field of  $B = 2.1[T]$ , enabling to calculate the amount of plasma current generated by this coil. For the calculation we use a plasma current of  $I_{plasma} = 100[A]$ , as described in the previous chapter. Now to use Lenz's law of equation 5.2 to calculate the pulse length the voltage of the plasma is required. Here Ohm's law can be used. This equation is displayed below.

$$I_p = V_L / R_p \quad (5.4)$$

Here  $R_p$  is the resistance and  $V$  is the loop voltage of the plasma. The resistance of the plasma can be determined by equation 5.5, for a toroidally shaped plasma.

$$R_p = \nu \cdot \frac{2 \cdot \pi \cdot R_0}{\pi \cdot a^2} \quad (5.5)$$

In this equation  $R_0$  and  $a$  are the major and minor radius of the plasma respectively, while  $\nu$  is the Spitzer resistivity of the plasma in  $[\Omega m]$ . The Spitzer resistivity can be calculated using formula 5.6.

$$\nu = 1.65 \cdot 10^{-9} \cdot \ln \Lambda \cdot T e^{(3/2)} \quad (5.6)$$

Next, using the resistance and the plasma current the plasma voltage can be calculated with equation 5.4. Now all values are known to calculate the pulse length. Equation 5.2 is rewritten into equation 5.7 to calculate the pulse length.

$$\frac{\delta \psi}{\delta t} = V_L = \frac{\delta B_0}{\delta t} \cdot A \quad (5.7)$$

In this equation  $A$  stands for the surface encircled by the solenoid and  $\frac{\delta B_0}{\delta t}$  stands for the magnetic flux. All values have been calculated and put in table 5.2, some assumed values about the plasma from table 1.1 are required as input for these calculations.

	$I_p[A]$	$B[T]$	$\ln \Lambda[-]$	$\nu[\Omega m]$	$R_p[\Omega]$	$V_L[V]$	$A[m^2]$	$t[ms]$
Value	100	2.1	17	$8.87 \cdot 10^{-4}$	0.11	11	0.003	0.5

**Table 5.2:** Ohmic current drive calculation values.

The table above shows that a plasma current of  $100[A]$  can only be generated for a length of  $t = 0.5[ms]$ , which is relatively short. This plasma current however must first be induced. The required magnetic flux,  $\Delta \Psi_{CIRCUS}$ , to reach  $100[A]$  in the plasma can be calculated using equation 5.8.

$$\Delta \Psi_{CIRCUS} = L_p \cdot I_p \quad (5.8)$$



In this equation  $L_p$  stands for the inductance of the plasma in  $[H]$ , which can be calculated with equation 5.9.

$$L_p = \mu_0 \cdot R_0 \cdot \ln\left(\frac{8 \cdot R_0}{a} - 1.5\right) \quad (5.9)$$

In this equation  $R_0$  and  $a$  stand for the major and minor radius, respectively. Equation 5.10 can be used to calculate the total magnetic flux of the central solenoid.

$$\Delta\Psi_{total} = 2 \cdot B \cdot A_{core} \quad (5.10)$$

The total magnetic flux calculated in the equation above must be larger than the required magnetic flux to generate a current of  $100[A]$  in the plasma. The values and results for the equations above have been included in table 5.3.

	$I_p[A]$	$B [T]$	$R_0[m]$	$a[m]$	$A_{core}[m^2]$	$\Delta\Psi_{CIRCUS}[\mu Wb]$	$\Delta\Psi_{total}[\mu Wb]$
Value	100	2.1	0.185	0.02	0.003	0.35	6

**Table 5.3:** Inductance ohmic current drive calculation values.

In the table above it can be seen that  $\Delta\Psi_{CIRCUS}$  is lower than  $\Delta\Psi_{total}$ . This leads to the conclusion that it is possible to reach a plasma current of  $100[A]$ , but the pulse length calculated in table 5.2 will be further reduced.

A superconducting central solenoid is not an ideal solution for CIRCUS since the internal construction of the device will have to be significantly altered, as the center is used to attach the coils. Additionally, the solenoid must maintain a temperature of  $9^\circ[K]$  to stay superconducting. Therefore, this method is not suited to generate the appropriate poloidal magnetic field for CIRCUS.

### 5.3 Electron cyclotron current drive

One of the ways in which a plasma current can be induced in CIRCUS is by a microwave, which produces Electron cyclotron current drive. ECCD works by injecting resonant electro-magnetic waves into the plasma, which will be absorbed. If the waves are launched tangentially into the plasma they are mainly absorbed by the electrons that move in the desired direction to generate a plasma current. ECCD is available for CIRCUS since it is necessary to create the breakdown for the plasma. An  $1[kW]$  microwave is available for the experiment and operates at a frequency of  $2.45[GHz]$ . As stated above, for the ECCD to function the electro-magnetic waves must be absorbed into the plasma. If the waves will be absorbed depends on the cut-off frequency of the plasma, calculated with equation 5.11. This frequency must be lower than what the microwave uses.

$$\omega_{ce} = 9 \cdot 10^3 \sqrt{\frac{n_e}{cm^{-3}}} [Hz] \quad (5.11)$$

When the assumed plasma density of table 1.1 is used, a cut-off frequency of  $2.85[GHz]$  is calculated. This is above the frequency the microwave uses. The density inside CIRCUS however can be lowered so that ECCD is possible.

It is unclear if ECCD is able to generate a plasma current of  $100[A]$ , since certain values required to calculate the generated current with equation 5.12 are unknown.

$$\Delta J = -e \left( \frac{v_{||2}}{\nu_2 \delta t} - \frac{v_{||1}}{\nu_1 \delta t} \right) \quad (5.12)$$

An experiment will have to be performed in the future to measure the generated plasma current by this ECCD device.

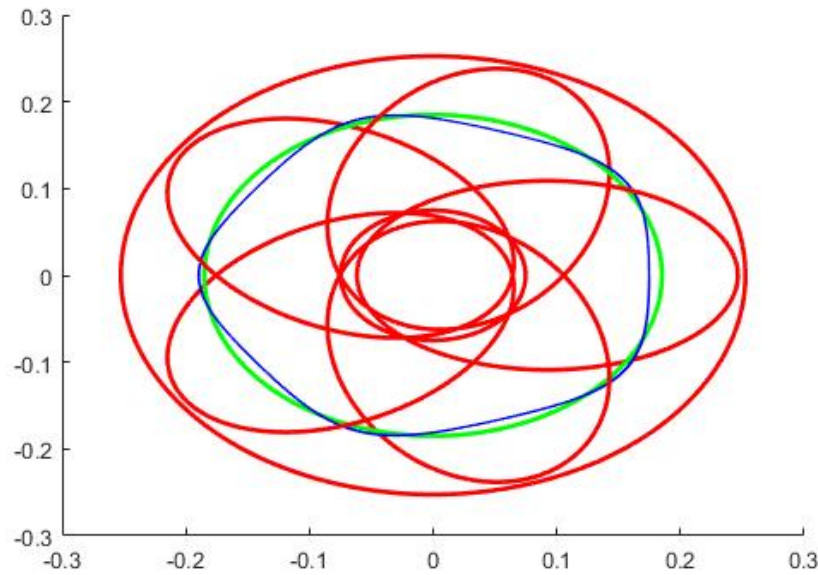
## 5.4 Current wire

The last option for generating a toroidal magnetic field to repair the field lines is by running a current through a wire. This wire must run through the center of the plasma. An advantage of this system is that you can directly influence the magnetic field, while other current drive methods are dependant on indirect plasma mechanisms. This makes testing the magnetic field more straightforward.

With formula 5.13 we can calculate the magnetic field that is generated by the wire, which is the generated poloidal field.

$$B = \frac{\mu_0 I}{2\pi r} \quad (5.13)$$

For the wire to work the center line of the plasma must be investigated. Since there are five tilted coils, it is possible that the magnetic axis of the CIRCUS device is not circular, but slightly in the shape of a pentagram. To investigate this issue figure 5.3 is created. This figure is a top view of CIRCUS, where the red lines represent the coils. The blue line indicates the two dimensional streamline in the x-y plane, which starts at the center of the plasma. The wire is indicated in green, at a distance of  $18.5[cm]$  from the center.



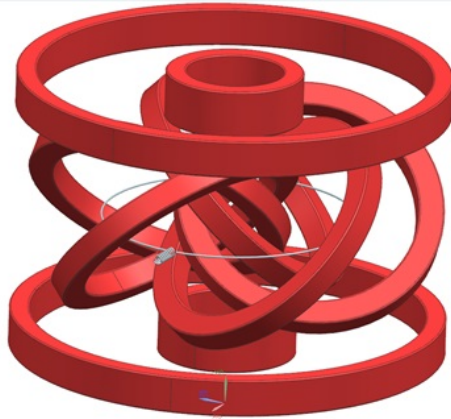
**Figure 5.3:** 2D streamline through center of plasma in a top view of Circus.

As can be observed in this figure the center of the plasma does not strictly follow the circular shape of the wire, since it has a slight pentagonal shape. The difference however is not very large and the wire stays inside the simulated plasma, as was shown in chapter 4. It can therefore be concluded that a circular shape for the wire is acceptable. To improve the performance the wire should follow the path indicated by the blue line in the figure above.

A disadvantage to a wire in the center of the plasma is that fusion conditions, where a high enough density and temperature are reached, will not be possible. The reason for this is that there is now a wire in the center instead of plasma. However, this is not a big issue since the main goal for CIRCUS is to investigate the magnetic field.

Figure 5.4 is a schematic drawing of the wire in the center of the plasma. The wire entering and exiting the center of the plasma must twist around, as can be seen in figure 5.4. This way a separate magnetic field will not be generated which would interrupt the existing magnetic configuration.

The required thickness and the expected magnetic force on this wire, as well as the temperature increase due to operation, will be investigated in the next chapter.



**Figure 5.4:** *Cad model CIRCUS coils with wire.*

## 5.5 Conclusions

The investigations in this chapter show that neutral beam current drive is not able to reach a plasma current of  $100[A]$ . Therefore, this method is not usable for CIRCUS. The investigation in electron cyclotron current drive showed that it is possible to penetrate the plasma if the density is lowered. It is however unclear if it could generate the required plasma current. An experiment is required to determine this possibility.

Applying ohmic current drive, by placing a super conductor in the center of the device, is not desirable either. The operating time is then relatively short,  $0.5[ms]$ , and it requires significant change in the construction of the device, because the solenoid has to be placed in the center. A temperature of  $9[K]$  must be maintained in the super conductor as well.

A simpler solution is installing a wire to generate the poloidal magnetic field directly. This technique is the most desirable method, since the generation of a poloidal field can be controlled directly. This allows for experiments at multiple settings. Therefore, the wire is selected to generate the poloidal field in the rest of this study.

This wire however will heat up due to the current inside. Accordingly, the wire must be thick enough so it does not generate too much heat.

## 6 Disturbances magnetic field

As stated in the research problem, possible mechanical disturbances on the magnetic field will be studied in this report. In this context the following effects will be investigated:

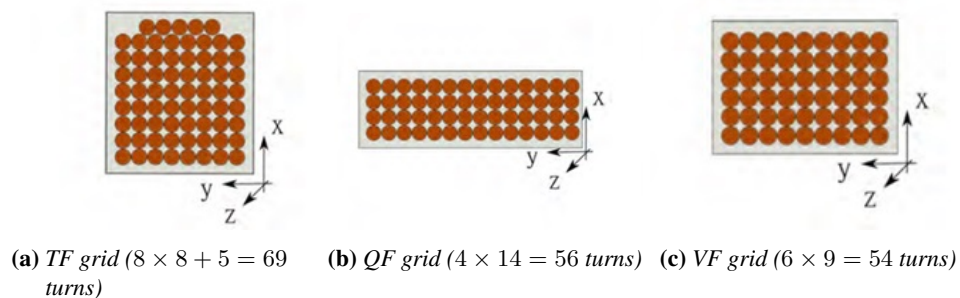
- How does the temperature increase affect the magnetism of the coils and what is the maximum temperature?
- What are the magnetic forces on the coils and does it disturb the magnetic configuration?
- How does heat and magnetic force affect the wire that generates the poloidal magnetic field?

### 6.1 Disturbance magnetic field due to heat

When a current is running through a wire the temperature of the wire increases, implying that the coils will have to be analyzed. The copper wires used to wind the coils have a diameter of  $3.264[mm]$ . The coils are insulated with a two-component epoxy called EPOX-TEK TK7110, of which the maximal operating temperature is  $80^{\circ}[C]$ . This means that the operation of CIRCUS must be stopped before the coils reach this temperature.

The coils are placed in a vacuum with little conduction. Previously a water cooled conduction mechanism existed on top of the vacuum vessel, which has been removed. In this section will be determined if this cooling mechanism is strictly necessary to operate CIRCUS. The heat that is generated by the wire that generates the poloidal magnetic field will be investigated too.

In figure 6.1a, 6.1b and 6.1c the sections of the toroidal field (TF) coils, quadrupole field (QF) coils and vertical field (VF) coils are shown, respectively.



**Figure 6.1:** Cut-through sections of coils (Maragkoudakis, 2019).

The generated magnetic field may change over time due to heating, if the power supply can not increase in power to keep a steady current in the coils. It is there-

fore assumed that the supplied current is constant. The coils however do have a temperature limit as is established earlier.

In order to calculate the temperature increase in a wire, which is dependant on the current, material and radius of the wire, formula 6.1 can be used.

$$dT = \frac{E(t)}{m \cdot cp} = \frac{I^2 \cdot R \cdot dt}{A \cdot l \cdot \rho_m \cdot cp_{cu}} \quad (6.1)$$

The resistance of a material is dependant on the temperature. Equation 6.2 shows how this can be calculated.

$$R = R_{ref}(1 + \alpha_{cu}(T - T_{ref})) \quad (6.2)$$

$R_{ref}$  is the reference resistance for copper at a temperature of  $293^\circ[K]$ . This can be found for the various coils in table 3.1. This value is measured experimentally on the coils. The resistance of the poloidal wire can be calculated using the equation below.

$$R = \frac{\rho \cdot l}{A} \quad (6.3)$$

In this equation  $R$  is the resistance in  $[\Omega]$  and  $A$  is the cross section area of the wire.  $\rho$  is the resistivity which is  $1.724 \cdot 10^{-8}[\Omega m]$ . The term  $l$  stands for the length, which can be calculated using equation 6.4.

$$l = 2 \cdot \pi \cdot r_{wire} \cdot N \quad (6.4)$$

In equation 6.4 the circumference of the wire is calculated and multiplied by the number of windings the coils has. This is the major radius of the device  $R_0 = 0.185[m]$  and 1 winding, resulting in a wire length of  $1[m]$ . Using this result and equation 6.3 the initial resistance of the wire at room temperature can be calculated. Now the temperature increase for the coils and the wire can be determined. Table 6.1 is a table where the temperature increase is calculated for standardised copper wire diameters.

Diameter wire [mm]	3.264	4.115	5.189	6.544	8.251	10.4
dT/dt [K/s]	0.71	0.28	0.11	0.044	0.018	0.007

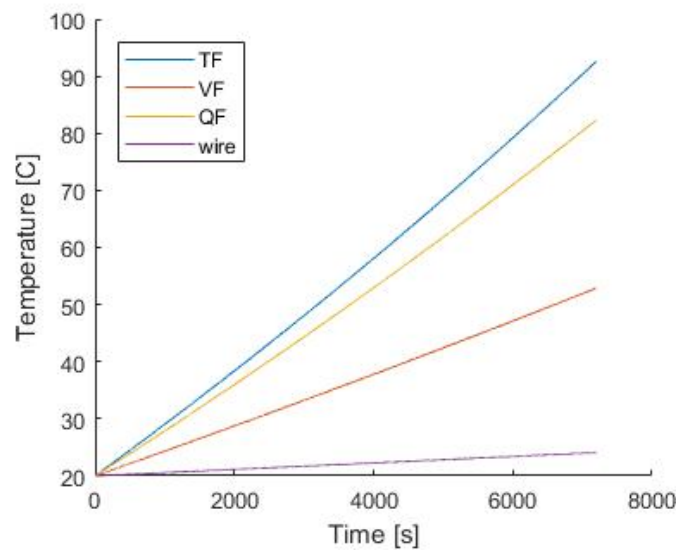
**Table 6.1:** Table of temperature increase per second for standard copper wire diameters.

The table above shows that the temperature increase over time quadratically decreases by increasing radius. A thicker wire is also more useful as it is less likely to deform due to its weight or magnetic forces. A wire diameter of  $3.264[mm]$  has been selected since the temperature increase is still relatively low and it is the same size as used for the other coils. The temperature of the wire is expected to stay far below the melting point of copper, which is  $1358^\circ[K]$ . Now the temperature increase of the coils and wire over time can be determined from which the values in table 6.2 will be used.

	I [A]	R <sub>ref</sub> [ $\Omega$ ]	A [ $m^2$ ]	$\rho_{cu}$ [ $kg/m^3$ ]	$c_{p_{cu}}$ [ $J/kgK$ ]	$\alpha_{cu}$
TF	200	0.287	$8.37 \cdot 10^{-6}$	$8.96 \cdot 10^6$	385	0.00393
VF	205	0.177	$8.37 \cdot 10^{-6}$	$8.96 \cdot 10^6$	385	0.00393
QF	512	0.095	$8.37 \cdot 10^{-6}$	$8.96 \cdot 10^6$	385	0.00393
Wire	100	0.002	$8.37 \cdot 10^{-6}$	$8.96 \cdot 10^6$	385	0.00393

**Table 6.2:** Table of coil and wire values to determine the temperature increase

Now with formula 6.1 and the values of table 6.2 a graph can be created where the increase in temperature of the various coils is shown over time.



**Figure 6.2:** Increase temperature coils and wire CIRCUS over time.

The figure above shows the temperature increase over an operating period of 2 hours. In this time frame the temperature of the TF coil, which is heating the fastest, is increased from  $20^{\circ}[C]$  to  $92.6^{\circ}[C]$ . As stated previously the maximal coil temperature is  $80^{\circ}[C]$ . This temperature is only reached after operating for 1 hour and 36 minutes, which is still a relatively long period of time. This figure also shows that the wire heats up relatively slow.

In this calculation heat loss has not been taken into account. Due to the high heat attained by the coils some heat may be lost due to conduction, convection and radiation. This may increase the time before the maximum coil temperature is reached. It can be assumed that conduction and convection is low, since the density in the vacuum chamber is low and the coils are only connected in a few places. The heat energy loss due to black-body radiation of an object can be calculated using equation 6.5.

$$q = \sigma \cdot T^4 \cdot A_s \quad (6.5)$$

Here  $q$  is the heat transferred in  $[W]$  and  $\sigma$  is the Stefan-Boltzmann constant in  $[W/m^2K^4]$ .  $T$  is the temperature of the coil in  $[K]$ , while  $A_s$  is the outer surface area  $[m^2]$  of the coils. The outer surface area of the coils has been calculated using figure 6.1, where sections of the coils are shown. Calculating this for the coils and wire show that the heat loss due to black-body radiation is low. For the TF coil this is around  $16[W]$  at a temperature of  $100^\circ[C]$ , which is not high enough to affect the results from figure 6.2, because of the large mass of the coils.

From figure 6.2 it can be concluded that the temperature increase of the coils does not significantly influence the magnetic field architecture, on the assumption that the supplied current can stay constant. It also can be concluded that a cooling system is not required.

## **6.2 Disturbance magnetic field due to magnetic forces on coils and wire**

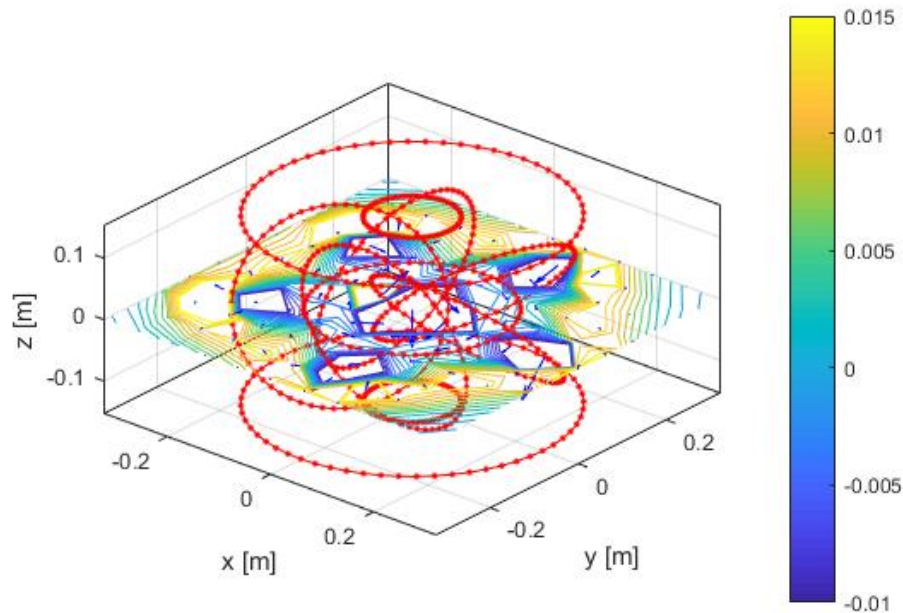
As stated in the beginning of this chapter, magnetic forces generated by CIRCUS is a factor that can influence the architecture of the magnetic field. If the magnetic forces are strong enough to displace the coils the magnetic field generated by the coils will be changed as well. Small deviations in the location or angle of a coil can have a drastic effect on the flux surfaces, as indicated earlier in this report. Therefore, in this section the change in the magnetic field generated by CIRCUS due to the magnetic forces will be investigated.

The force on a current carrying wire can be calculated using Lorentz's law, pointed out in equation 6.6.

$$F = I\ell \times B \quad (6.6)$$

In this equation  $F$  stands for the amount of force generated in  $[N]$ .  $I$  stands for the current in the wire in  $[A]$  and  $\ell$  is the length of the wire in  $[m]$ , while  $B$  stands for the strength of the magnetic field in  $[T]$ . As can be seen in this equation a vector cross-product is used. The reason for this is that the generated force, the current and the magnetic force are perpendicular to each other.





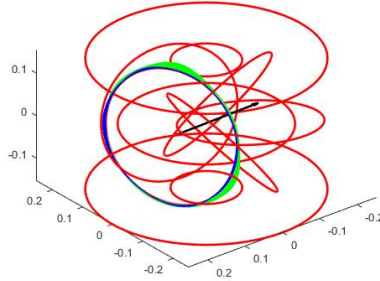
**Figure 6.3:** *MATLAB-model of magnetic field strength CIRCUS in the z direction on the x-y plane in [T].*

As demonstrated in figure 6.3 the magnetic field generated by CIRCUS is complex, signifying that the force distribution on the coils will be complex as well. In view of this complexity the MATLAB-model explained in section 2.2 will be used to simulate the magnetic forces on the coils during operation.

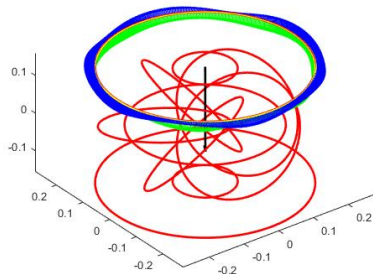
Subsequently, after the force distribution on the coils has been determined, the construction around the coils must be investigated. The calculated forces on the coils will be applied to the construction provided in figure 6.6. As stated in section 4.2 if the location of the coils deviates less than 1% of the intended location than the magnetic field will still follow the intended path. This percentage is taken as a limit for maximal deflection, because as mentioned this error in location can still be corrected by the generated poloidal field.

### 6.2.1 Forces in between coils

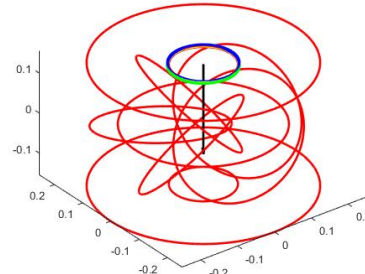
This subsection takes a closer look at the magnetic field externally applied to the toroidal, vertical and quadruple field coils. Figure 6.4 shows these external magnetic forces simulated by the MATLAB-code.



(a) Simulation of CIRCUS where the magnitude and direction of Lorentz force on TF coil are shown, simulated by the MATLAB-code.



(b) Simulation of CIRCUS where the magnitude and direction of Lorentz force on VF coil are shown, simulated by the MATLAB-code.



(c) Simulation of CIRCUS where the magnitude and direction of Lorentz force on QF coil are shown, simulated by the MATLAB-code.

**Figure 6.4:** Forces in between coils.

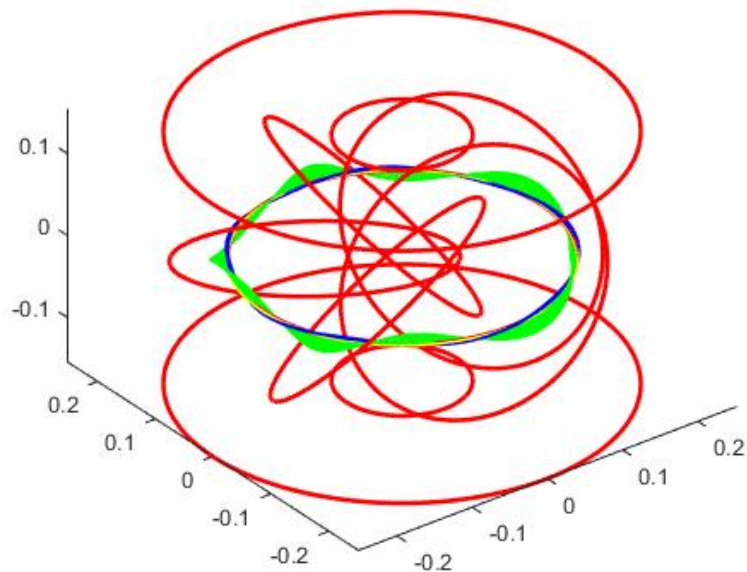
Figures 6.4a, 6.4b and 6.4c show the direction of the external forces generated by the external magnetic field of the TF, VF and QF coils, respectively.

Equation 6.6 is used to calculate the strength and the direction of the applied Lorentz force. The blue vectors represent the magnetic field strengths and directions of the external magnetic field in the 600 points around the coil, while the green vectors are the resulting force vectors for each element. The currents used in the coils are listed in table 3.1. The direction of the current has been taken into account for all 600 points. The total current in the TF coils is  $13.8[kA]$ , since it has 69 windings and each winding has a current of  $200[A]$ . The total current in the VF and QF coils are  $11.1[kA]$  and  $28.7[kA]$ , respectively.

The direction of the total force on the coils is indicated by the black vector. The values of these forces can be found in table 6.3 of subsection 6.2.3, here it has been assumed that the coils are rigid objects.

### 6.2.2 Force on poloidal magnetic field wire

The next study involves the Lorentz force on the poloidal wire. In order to determine the force's direction and magnitude on the poloidal wire the MATLAB-code is applied. The result of this simulation is illustrated in figure 6.5.



**Figure 6.5:** Lorentz force on the poloidal wire, simulated by the MATLAB-code.

The total Lorentz force has a value close to zero. The reason for this is the symmetrical nature of CIRCUS and that the majority of the magnetic field is in the same direction as the current. This implies that it is possible to hang the wire with thin rods in such a manner that there is no movement due to its load.

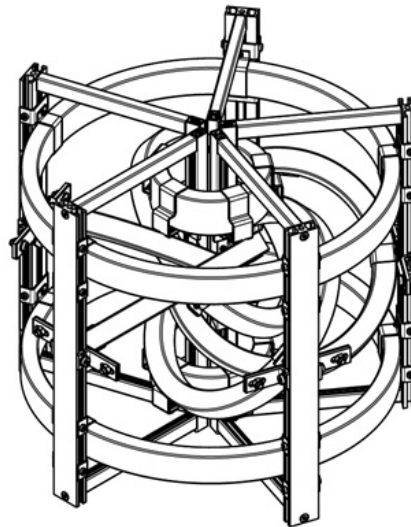
### 6.2.3 Table of forces on coils

Table 6.3 lists the determined Lorentz forces on the five toroidal field coils, two vertical field coils, two quadrupole field coils and the wire, obtained by the calculations in the previous subsections. The force vectors are displayed for all three dimensions.

Coils	X Force in [N]	Y Force in [N]	Z Force in [N]
TF1	-25	-87	3.2
TF2	76	-51	3.3
TF3	71	56	4.1
TF4	-31	-85	4.1
TF5	-91	-3.5	3.7
VF1	0	0	-272
VF2	0	0	272
QF1	0	0	-552
QF2	0	0	552
Wire	0	0	0

**Table 6.3:** Loads on individual coils in [N].

The total magnitude due to forces of the toroidal field coils is around  $91[N]$ . The Lorentz force on the vertical and quadrupole field coils are  $272[N]$  and  $552[N]$ , respectively. From these low values and the fact that the construction which holds the coils in place has been over-designed, as seen in figure 6.6, it can be concluded that there is no movement for the coils due the Lorentz forces. If the coils do not move the magnetic field is also not changed due to magnetic forces.

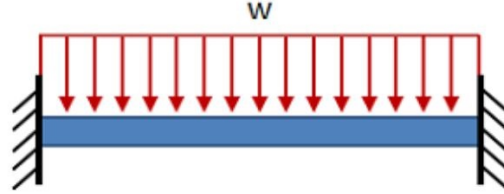


**Figure 6.6:** Graphical representation of construction CIRCUS.

#### 6.2.4 Deformations in coils

For the investigation in this chapter it is assumed that the coils and the wire are rigid objects. In this subsection this claim will be investigated. The coils and the wire are mounted at different points on the construction. In the previous sections

it was concluded that these mounting locations do not deflect under the load the coils are subject to. Therefore the force load on the coils and wire can be seen as a build-in beam with a uniformly distributed load, shown in figure 6.7.



**Figure 6.7:** Build-in beam with uniformly distributed load (S.Gurumoorthy, 2019).

To approximate the uniformly distributed load for the coils the average force per element of the coils from the MATLAB-code is used. To describe the deflection of the center of the build-in beam equation 6.7 can be used.

$$d = \frac{-wL^4}{384EI} \quad (6.7)$$

In this equation  $d$  is the deflection in the center of the beam in  $[m]$ . The uniformly distributed load is expressed by  $w$ , which is in  $[N/m]$ .  $E$  represents the Young's modulus in  $[Pa]$ . Then using the length of the beam,  $L$ , and the second moment of inertia,  $I$ , the deflection can be calculated. For the coils the Young's modulus of epoxy resin is used. It is assumed that the epoxy "EPOX-TEK TK7110" and not the copper is the load bearing material. The Young's modulus of epoxy resin is  $3[GPa]$  (Engineering ToolBox, 2003). The wire is made out of solid copper, which has a Young's modulus of  $117[GPa]$ . The length of each beam is the part between two mounting points. This concludes that the beam length is the circumference of a coil divided by the amount of mounting points. The second moment of inertia for the rectangular shaped coils can be calculated using equation 6.8.

$$I = \frac{bh^3}{12} \quad (6.8)$$

The width and height of the TF, VF and QF coils are respectively,  $52.2 \times 29,4[mm^2]$ ,  $91.4 \times 13[mm^2]$  and  $58.8 \times 19.6[mm^2]$ . To calculate the second moment of inertia for a rectangle the orientation of the rectangle is important. For this calculation the worst case scenario is used, in which the height is the lower value and the width the higher.

For the round copper wire equation 6.9 can be applied to calculate the second moment of inertia.

$$I = \frac{\pi D^4}{64} \quad (6.9)$$

In this equation  $D$  is the diameter of the wire, which was determined in section 6.1 to be  $3.264[mm]$ . The uniformly distributed load on the coils are calculated using the MATLAB-code. The mean force per element, of which there are 600 in each coil, is calculated. Then using the circumference of the coil the uniformly distributed load can be approximated. These values are listed in table 6.4. The deflection calculated by employing equation 6.7 for the coils and wire are also included in this table.

Coils	Mount points	Circumference [m]	Mean force per element [N]	w [N/m]	I [m <sup>4</sup> ]	d [mm]
TF	2	0.97	0.67	692	$1.11 * 10^{-7}$	-0.3
VF	5	1.59	0.47	177	$1.67 * 10^{-8}$	-0.09
QF	5	0.47	0.46	587	$1.05 * 10^{-9}$	-0.04
Wire	5	1	0.003	1.5	$5.57 * 10^{-12}$	-0.01

**Table 6.4:** Mean force on elements in MATLAB-simulation and values from build-in beam with uniformly distributed load calculation.

The table above illustrates that the deflection is small for all coils. The largest deformation is in the TF coils and is only  $0.19[mm]$ . This results from a worst case scenario, where all the uniformly distributed force is directly applied to the thinnest orientation of the rectangular coil. The assumption that there is no deformation in the coils and the wire, so that they are rigid, may be considered correct.

### 6.3 Conclusions

In this chapter two possible mechanical disturbances on the magnetic field have been investigated, namely the effect of temperature and mechanical forces.

Running a current through a material such as copper increases the temperature. Increasing the temperature of a material increases its resistivity, implying that for the same amount of power a lower current is reached in a coil. Changing the current in a coil changes the generated magnetic field. Therefore it was assumed that the supplied current can stay constant.

The epoxy the coils are made out of is a limiting factor on the amount of heat the coils can have. This material has a maximal operating temperature of  $80^{\circ}[C]$ . In this chapter it was shown that the TF coil heats the fastest, but only reaches this temperature after 1 hour and 36 minutes. It can therefore be concluded that temperature does not have an effect on the magnetic architecture as long as the operating time stays below the previously mentioned limit. The heat loss in the coils and wire due to black-body radiation can be neglected since it is so small. The show heating of the coils means that a cooling system is not required to operate CIRCUS.

This chapter also showed that the generated magnetic field does not influence the coils significantly, because the forces involved are low and the construction of CIRCUS is over designed. This means that the magnetic field will not change because of the displacement of the coils due to the forces on the coils.

The MATLAB-code showed that the Lorentz forces on the coils due to the magnetic field are low. The total magnetic force on the toroidal field coil is  $91[N]$ , for the vertical field coil  $272[N]$  and  $552[N]$  on the quadrupole coil. The Cartesian coordinates of these vectors are shown in table 6.3.

The wire heats up very slowly, therefore this wire can be relatively thin. A wire diameter of  $3.264[mm]$  is chosen. This is the same diameter as was used for the coils. The force on the poloidal wire has been calculated using the MATLAB-code, resulting in a force load of  $0[N]$ . The reason for the low force on the wire is that the magnetic field and the current are running in the same toroidal direction over the central axis of the plasma.

The deformation inside the coils and wire is also relatively small, the values for this are listed in table 6.4. From this it can be assumed that the coils are rigid.

The poloidal wire can be mounted in such a way that there is a very low displacement below the external magnetic load.

From this chapter it can be concluded that the mechanical influences of temperature and magnetic force does not influence the magnetic architecture.

## 7 Final conclusions and outlook

This chapter summarizes the results and conclusions based on the research conducted in this report and provides an outlook to further investigations.

### 7.1 Final conclusions

In this report a magnetic field architecture for CIRCUS has been found, which satisfies all predefined conditions discussed in the research problem section. To achieve this a MATLAB-model to simulate the magnetic field of CIRCUS has been developed. This model makes use of the Biot-Savart law to calculate the magnetic field strength in certain points in a three dimensional volume. These integration points can be controlled for different types of analyses. In this model the coils are approximated by infinitely thin elements. In chapter 2 it is demonstrated that the developed MATLAB-code is a good approach to calculate the magnetic field, since it complies with an earlier performed VMEC study (Volpe et al., 2014) if the same parameters are used. The gyration radius of electrons is small enough to be confined in the plasma.

The coil currents at which the initial conditions described in the research problem are met, are displayed in table 3.1. These results have been found performing calculations and applying the developed MATLAB-model. For the toroidal field coil a current of  $200[A]$  is required, while the vertical and quadrupole field coils require  $205[A]$  and  $512[A]$  respectively to satisfy all conditions. A lower TF current can be used as long as the same ratio's are applied. These findings differ from the results by Volpe (Volpe et al., 2014), which can be seen in appendix B. In subsection 3.3 it is proven that increasing the current used to generate the QF field higher than what is used for the VF field increases the volume of the plasma.

When using the values of table 3.1 there are field lines that lay on closed flux surfaces. A commercially available microwave could be used to heat the plasma at the center of the device, since the toroidal magnetic field has a magnetic strength of  $0.0875[T]$ , which corresponds to  $2.45[GHz]$ . The magnetic field however is not stable under these conditions. This research has proven in section 3.6 that without an additional poloidal magnetic field the sensitivity of the magnetic field with respect to misalignments is too high. With a slight, less than 1%, deviation in the location or current of the coils the field lines no longer follow the intended path.

MATLAB simulations have shown that a poloidal magnetic field generated by an enclosed current of  $100[A]$  is enough to decrease the sensitivity in such a way that it allows for location and current deviations of at least 1%.

In this research four different ways to generate a poloidal magnetic field have been explored in chapter 5, namely: neutral beam current drive, ohmic current drive, electron cyclotron current drive and a wire along the center of the plasma. Neutral beam current drive is not a possible current drive method for CIRCUS, since the minor radius and the particle density are too low for the neutral beam to



successfully ionize in the plasma, namely  $n_e = 10^{-3}[10^{20}m^{-3}]$  and  $a = 0.01[m]$ . For the calculation on ohmic current drive a central solenoid made out of Niobium Tin was used. A plasma current of  $100[A]$  could be generated, but only for a period of  $0.05[ms]$ . This is relatively short. Additionally, this would require a significant change in the central construction of the device. Electron cyclotron current drive for the current setup of CIRCUS, which makes use of a commercial microwave, is possible but the plasma density must be lower than what was assumed in section 1.5. The amount of plasma current ECCD can generate however is not clear. The remaining possibility is a wire through the center axis of the plasma. MATLAB calculations however show that the shape of the plasma axis on a major radius of  $R = 18.5[cm]$  is not perfectly circular. This method allows for flexible and direct manipulation of the poloidal magnetic field and is therefore suitable for experiments. This method is the chosen solution for generating a poloidal magnetic field.

The mechanical influences of temperature and magnetic force have also been investigated in this report. The maximum coil operating temperature of the coils is  $80^\circ[C]$ , because of the employed epoxy insulation. The TF coil increases its temperature during operation the fastest. After operating for 1 hour and 36 minutes this maximum temperature may be reached in this coil, as found out in section 6.1. This is a relatively long time, it can therefore be concluded that the temperature in the coils does not influence the operation of CIRCUS. A cooling system is therefore not required. Heating of the wire does also not significantly change the magnetic field, since the current is relatively low.

The forces inside the coils are relatively low. This was concluded in section 6.2. Simulating the TF coil, QF coil, VF coil and the wire, results in table 6.3, in which the forces on the coils are indicated. The construction of CIRCUS is overdesigned in such a way that no movement, due to the magnetic forces, of the coils can be assumed. This implies that the Lorentz force does not influence the magnetic field. In this chapter it was also calculated that there is no deformation inside the coils or wire between to points where they are connected to the construction. The MATLAB-model calculated that the poloidal wire has a force close to  $0[N]$ . Therefore only thin stutts are required to hang the poloidal wire. It is recommended that the stutts are as thin as possible so they do not interfere with the plasma. A steel wire with a thickness of  $1[mm]$  to hang the poloidal copper wire is already far above the minimum required.

## 7.2 Outlook

In the Poincaré figures of section 4.1 the wire that generates the poloidal magnetic field does not go straight through the center of the plasma. This causes an island shaped center of the flux surfaces which is not completely in the center of the plasma. If a wire is used with a slightly pentagonal shape this may be resolved. For future experiments the shape of the wire can be changed to improve the operation of CIRCUS. If the shape displayed in figure 5.3 is used the wire will go straight

through the center of the plasma for currents up to  $140[A]$ . This may give better results. Additional simulations will have to be performed to confirm this.

Higher plasma currents than  $140[A]$  maybe used to increase the plasma volume further. But then a smaller major radius is required. Subsequently experiments may be performed to confirm this.

As stated ECCD is possible in CIRCUS, but the plasma current that may be generated by this technique is unclear. An experiment will have to be designed where the effectiveness of this plasma current generation method is measured.

The main concept of CIRCUS is using the tilted interlocked toroidal field coils to partly contribute to the poloidal magnetic field. However, the extent of this contribution is not clear, which may need further investigation. This could be realised by changing the angle of attack of the coils. By looking at what effect the changing angle of the coils has on the shape and stability of the closed flux surfaces the contribution of the tilted interlocked architecture may be investigated.

Lately an experiment has been performed on CIRCUS with the poloidal wire inside with a current of  $0[A]$  and  $100[A]$ . This experiment shows that it possible to create a plasma inside CIRCUS. These experiments can be seen in figure 7.1a and figure 7.1b (der Kraats, 2021). If the two are compared it may be noticed that the plasma connects with the wire when a current of  $100[A]$  is used.



(a) Picture of CIRCUS experiment with poloidal wire with a wire current of  $0[A]$ . (b) Picture of CIRCUS experiment with poloidal wire with a wire current of  $100[A]$ .

## References

- Bruhns, H., Brendel, R., Raupp, G., & Steiger, J. (1987, dec). Study of the low aspect ratio limit tokamak in the heidelberg spheromak experiment. *Nuclear Fusion*, 27(12), 2178–2182. Retrieved from [https://doi.org/10.1088/0029-5515/27/12/017](https://doi.org/10.1088/0029-5515%2F27%2F12%2F017) doi: 10.1088/0029-5515/27/12/017
- Department of Energy United States of America. (2015). AN ASSESSMENT OF ENERGY TECHNOLOGIES AND RESEARCH OPPORTUNITIES.
- der Kraats, T. E. V. (2021). Visualization of magnetic field lines and quantitative field strength measurements in CIRCUS.
- De Rooij, T. (2019). *Modeling the misalignment sensitivity of CIRCUS* (Tech. Rep.).
- Earthjustice. (2017). Harm to human health from breathing and ingesting coal ash toxicants. Retrieved from <https://earthjustice.org/sites/default/files/files/coal-ash-man{ }COMPLETE{ }2017-07-20{ }light.pdf>
- Engineering ToolBox. (2003). *Young's Modulus - Tensile and Yield Strength for common Materials*. Retrieved from <https://www.engineeringtoolbox.com/young-modulus-d{ }417.html>
- Euratom-CEA. (2016). *Confinement and magnetic bottles*. Retrieved 2021-01-03, from <http://www-fusion-magnetique.cea.fr/gb/fusion/principes/principes02.htm>
- Everett, B., Boyle, G., Peake, S., & Ramage, J. (2011). *Energy Systems and Sustainability: Power for a Sustainable Future*.
- Freidberg, J. P. (2007). *Plasma physics and fusion energy* (Vol. 9780521851077). doi: 10.1017/CBO9780511755705
- Japan Atomic Energy. (2000). *Principle of steady-state (continuous) operation of tokamak fusion reactor*. Retrieved from <https://rdreview.jaea.go.jp/tayu/ACT00E/02/z02021.htm>
- Leiser, M. (1991). Plasma Physics and Controlled Nuclear Fusion Research. *IAEA*, 24(4). Retrieved from <https://www.iaea.org/sites/default/files/publications/magazines/bulletin/bull124-4/24406184950.pdf> doi: 10.1051/epr/19760712012
- Maragkoudakis, E. (2019). Design & Production Of Coils For CIRCUS Plasma Confinement Device.
- Quéval, L. (2015). *BSmag Toolbox*.

- Ritchie, H. (2020). *What are the safest sources of energy?* Retrieved from <https://ourworldindata.org/safest-sources-of-energy>
- Schleussner, C.-F., Lissner, T. K., Fischer, E. M., Wohland, J., Perrette, M., Golly, A., ... Schaeffer, M. (2016). Differential climate impacts for policy-relevant limits to global warming: the case of 1.5 °c and 2 °c. *Earth System Dynamics*, 7(2), 327–351. Retrieved from <https://esd.copernicus.org/articles/7/327/2016/> doi: 10.5194/esd-7-327-2016
- S.Gurumoorthy, L. B. R. (2019). Simulation and Experimental Substantiation of Beam Deflection under Guided End Conditions. *International Journal of Innovative Technology and Exploring Engineering*, 9(2).
- Smits, T. (2021). Numerical optimisation of the coil currents in the CIRCUS stellarator.
- Superconwire. (2021). *Nb3Sn Superconducting Wires*. Retrieved 2021-05-17, from <http://www.supercon-wire.com/content/nb3sn-superconducting-wires>
- Van Zalk, J., & Behrens, P. (2018). The spatial extent of renewable and non-renewable power generation: A review and meta-analysis of power densities and their application in the u.s. *Energy Policy*, 123, 83 - 91. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0301421518305512> doi: <https://doi.org/10.1016/j.enpol.2018.08.023>
- van Rooij, G., Lopes Cardozo, N., & de Blank, H. (2017). *Fusion reactors: Extreme materials, intense plasma-wall interaction*.
- Volpe, F., Clark, A., Doumet, M., Hammond, K., Kornbluth, Y., Spong, D., & Sweeney, R. (2014). Proto-circus tilted-coil tokamak-torsatron hybrid: Design and construction. *Fusion Engineering and Design*, 89(11), 2732 - 2737. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0920379614004645> doi: <https://doi.org/10.1016/j.fusengdes.2014.07.012>
- Wesson, J. (2004). *Tokamaks* (third edit ed.). Oxford: Clarendon Press.

## Appendices

### A Locations six toroidal field coils CIRCUS

Here are the coil locations and angles of the six toroidal field coil version of CIRCUS shown.

Coils	x [cm]	y [cm]	z [cm]	Radius [cm]	Angle [°]	Windings [-]
TF1	5.35	9.27	0	15.4	45	69
TF2	-5.35	9.27	0	15.4	45	69
TF3	-10.7	0	0	15.4	45	69
TF4	-5.35	-9.27	0	15.4	45	69
TF5	5.35	-9.27	0	15.4	45	69
TF6	10.7	0	0	15.4	45	69
VF1	0	0	15.4	25.25	0	54
VF2	0	0	-15.4	25.25	0	54
QF1	0	0	15	7.25	0	56
QF2	0	0	-15	7.25	0	56

**Table A.1:** Table of coil locations in meters, with center of device as origin.

### B Coil parameters CIRCUS proposed by Volpe

These are the proposed solutions to the magnetic field by Volpe.

Coil	R [cm]	$Z_{coil}$ [cm]	$R_{coil}$ [cm]	I [A]	$N_{turns}$ [-]	R [mΩ]	V [V]
TF	16	0	10.7	75	69	297	22.3
VF	25.25	15.4	0	75	54	177	13.3
QF	7.25	15	0	75	56	95	7.1

**Table B.1:** Proposed coil settings for 6 TF version CIRCUS (Volpe et al., 2014)

## C MATLAB-code Quéval

MATLAB-code by Quéval to solve Biot-Savart law.

```

1 function [BSmag] = BSmag_add_filament (BSmag, Gamma, I,
    ↪ dGamma)
2 %-----
3 % NAME:      BSmag_add_filament.m
4 % WHAT:      Adds a filament to the BSmag analysis.
5 % REQUIRED:   BSmag Toolbox 20150407
6 % AUTHOR:    20150407, L. Quéval (loic.queval@gmail.
    ↪ com)
7 % COPYRIGHT: 2015, Loic Quéval, BSD License (http://
    ↪ opensource.org/licenses/BSD-3-Clause).
8 %
9 % USE:
10 % [BSmag] = BSmag_add_filament (BSmag, Gamma, I, dGamma)
11 %
12 % INPUTS:
13 %   BSmag      = BSmag data structure
14 %   Gamma      = Filament points coordinates (x,y,z),
    ↪ one point per line [m,m,m]
15 %   I          = Filament current (flows from first
    ↪ point towards last point) [A]
16 %   dGamma     = Filament max discretization step [m]
17 %
18 % OUTPUTS:
19 %   BSmag      = Updated BSmag data structure
20 %   BSmag.Nfilament = Number of
    ↪ filaments
21 %   BSmag.filament(*).* = Filament
    ↪ structure
22 %   BSmag.filament(*).Gamma = Filament points
    ↪ coordinates (x,y,z), one point per line [m,m,m]
23 %   BSmag.filament(*).I = Filament current
    ↪ (flows from first point towards last point) [A]
24 %   BSmag.filament(*).dGamma = Filament max
    ↪ discretization step [m]
25 %-----
26
27 n = BSmag.Nfilament+1;
28 BSmag.filament(n).Gamma = Gamma;
29 BSmag.filament(n).I = I;
30 BSmag.filament(n).dGamma = dGamma;

```

```

31 BSmag.Nfilament = n;
32
33 %Plot P (where there is a current source)
34 figure(1)
35     plot3(Gamma(:,1), Gamma(:,2), Gamma(:,3), '-r')
36 axis tight

```

---

```

1 function [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,X,
    ↪ Y,Z)
2 -----
3 % NAME:      BSmag_get_B.m
4 % WHAT:      Calculates B at field points.
5 % REQUIRED:   BSmag Toolbox 20150407
6 % AUTHOR:    20150407, L. Quéval (loic.queval@gmail.
    ↪ com)
7 % COPYRIGHT: 2015, Loic Quéval, BSD License (http://
    ↪ opensource.org/licenses/BSD-3-Clause).
8 %
9 % USE:
10 % [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,X,Y,Z)
11 %
12 % INPUTS:
13 % BSmag      = BSmag data structure
14 % X          = Field points x-coordinate vector or
    ↪ matrix
15 % Y          = Field points y-coordinate vector or
    ↪ matrix
16 % Z          = Field points z-coordinate vector or
    ↪ matrix
17 %
18 % OUTPUTS:
19 % BSmag      = BSmag data structure (no update)
20 % X          = Field points x-coordinate vector or
    ↪ matrix
21 % Y          = Field points y-coordinate vector or
    ↪ matrix
22 % Z          = Field points z-coordinate vector or
    ↪ matrix
23 % BX        = Field points B x-component vector or
    ↪ matrix
24 % BY        = Field points B y-component vector or
    ↪ matrix
25 % BZ        = Field points B z-component vector or
    ↪ matrix

```

```

26 %-----
27
28 mu0 = 4*pi*1e-7; % vacuum permeability [N/A^2]
29
30 BX = zeros(size(X,1), size(X,2), size(X,3));
31 BY = zeros(size(X,1), size(X,2), size(X,3));
32 BZ = zeros(size(X,1), size(X,2), size(X,3));
33
34 for nF = 1:BSmag.Nfilament % Loop on each filament
35
36     Gamma = BSmag.filament(nF).Gamma;
37     dGamma = BSmag.filament(nF).dGamma;
38     I = BSmag.filament(nF).I;
39
40     % Discretization of Gamma
41     x_P = []; y_P = []; z_P = [];
42     N = size(Gamma,1)-1; % Number of points defining
43     ↪ Gamma
44     for i = 1:N % Loop on the segments defining gamma
45         L_Gamma_i = norm(Gamma(i,:) - Gamma(i+1,:));
46         NP = ceil(L_Gamma_i/dGamma); % Number of
47         ↪ points required to have a discretization
48         ↪ step smaller than dGamma
49         x_P = [x_P, linspace(Gamma(i,1), Gamma(i+1,1),
50         ↪ NP)]; % discretization of Gamma for x
51         ↪ component
52         y_P = [y_P, linspace(Gamma(i,2), Gamma(i+1,2),
53         ↪ NP)]; % discretization of Gamma for y
54         ↪ component
55         z_P = [z_P, linspace(Gamma(i,3), Gamma(i+1,3),
56         ↪ NP)]; % discretization of Gamma for z
57         ↪ component
58     end
59
60     % Add contribution of each source point P on each
61     ↪ field point M (where we want to calculate
62     ↪ the field)
63     for m = 1:size(X,1);
64         for n = 1:size(X,2);
65             for p = 1:size(X,3);
66
67                 % M is the field point
68                 x_M = X(m,n,p);
69                 y_M = Y(m,n,p);

```



```

59         z_M = Z(m,n,p);
60
61         % Loop on each discretized segment of
62         ↪ Gamma PkPk+1
63         for k = 1:length(x_P)-1
64             PkM3 = (sqrt((x_M-x_P(k))^2 + (y_M-y_P
65                 ↪ (k))^2 + (z_M-z_P(k))^2))^3;
66             DBx(k) = ((y_P(k+1)-y_P(k))*(z_M-z_P(k)
67                 ↪ ) - (z_P(k+1)-z_P(k))*(y_M-y_P(k)
68                 ↪ ))/PkM3;
69             DBy(k) = ((z_P(k+1)-z_P(k))*(x_M-x_P(k)
70                 ↪ ) - (x_P(k+1)-x_P(k))*(z_M-z_P(k)
71                 ↪ ))/PkM3;
72             DBz(k) = ((x_P(k+1)-x_P(k))*(y_M-y_P(k)
73                 ↪ ) - (y_P(k+1)-y_P(k))*(x_M-x_P(k)
74                 ↪ ))/PkM3;
75         end
76     % Sum
77     BX(m,n,p) = BX(m,n,p) + mu0*I/4/pi*sum(DBx
78         ↪ );
79     BY(m,n,p) = BY(m,n,p) + mu0*I/4/pi*sum(DBy
80         ↪ );
81     BZ(m,n,p) = BZ(m,n,p) + mu0*I/4/pi*sum(DBz
82         ↪ );
83 end
84 end
85 end
86 end
87 end

```

```

1 function [BSmag] = BSmag_init()
2 %-----
3 % NAME:      BSmag_init.m
4 % WHAT:      Initializes a Biot-Savart magnetostatic
5 ↪ analysis.
6 % REQUIRED:   BSmag Toolbox 20150407
7 % AUTHOR:    20150407, L. Queval (loic.queval@gmail.
8 ↪ com)
9 % COPYRIGHT: 2015, Loic Quéval, BSD License (http://
10 ↪ opensource.org/licenses/BSD-3-Clause).
11 %
12 % USE:
13 % [BSmag] = BSmag_init()

```

```

11 %
12 % INPUTS:
13 %
14 % OUTPUTS:
15 %     BSmag     = Initialized BSmag data structure
16 %     BSmag.Nfilament     = Number of filaments
17 %-----
18
19 BSmag.Nfilament = 0; %Number of source filament
20
21 % Open default figure to plot source points and field
    ↪ points
22 figure(1), hold on, grid on, box on, axis equal
23 xlabel('x [m]'), ylabel('y [m]'), zlabel('z [m]')
24 view(3), axis tight

1 function [] = BSmag_plot_field_points(BSmag,X,Y,Z)
2 %-----
3 % NAME:     BSmag_plot_field_points.m
4 % WHAT:     Plots all the field points (where we
    ↪ want to calculate the field) in the default
    ↪ figure.
5 % REQUIRED:  BSmag Toolbox 20150407
6 % AUTHOR:   20150407, L. Queval (loic.queval@gmail.
    ↪ com)
7 % COPYRIGHT: 2015, Loic Quéval, BSD License (http://
    ↪ opensource.org/licenses/BSD-3-Clause).
8 %
9 % USE:
10 %     [] = BSmag_plot_field_points(BSmag,X,Y,Z)
11 %
12 % INPUTS:
13 %     BSmag     = BSmag data structure
14 %     X         = Field points x-coordinate vector or
    ↪ matrix
15 %     Y         = Field points y-coordinate vector or
    ↪ matrix
16 %     Z         = Field points z-coordinate vector or
    ↪ matrix
17 %
18 % OUTPUTS:
19 %-----
20
21 %Plot M (where we want to calculate the field)

```

```
22 figure(1)
23     plot3(X(:),Y(:),Z(:),'kx')
24 axis tight
```

## D MATLAB-code

This code is designed to simulate the magnetic field of CIRCUS and calculate magnetic forces.

```

1  %-----
2  %
3  %  AUTHOR:      P.M.Q. van der Voort
4  %-----
5
6  % Initialize
7  clear all, close all, clc
8  BSmag = BSmag_init(); % Initialize BSmag analysis
9
10 %% Coil location constants
11 Major_R = 0.16;           %Major radius
12 Minor_a = 0.05;         %Minor radius
13 r = 0.154;              %TF coil radius
14
15 R = 0.0924;             %TF coil center location
16
17 qf_R = 0.0748;          %QF coil radius
18 qf_z = 0.15;           %QF coil z location
19 vf_R = 0.2525;         %VF coil radius
20 vf_z = 0.154;          %VF coil z location
21
22 phi1 = 2*pi/5;
23 phi2 = 4*pi/5;
24 phi3 = 6*pi/5;
25 phi4 = 8*pi/5;
26 phi5 = 10*pi/5;
27
28 % phi1 = 2*pi/6;         %6 TF coil CIRCUS
29 % phi2 = 4*pi/6;         %6 TF coil CIRCUS
30 % phi3 = 6*pi/6;         %6 TF coil CIRCUS
31 % phi4 = 8*pi/6;         %6 TF coil CIRCUS
32 % phi5 = 10*pi/6;        %6 TF coil CIRCUS
33 % phi6 = 12*pi/6;        %6 TF coil CIRCUS
34
35 aoa = -pi/4;           % Angle of attack
36
37 %% Solutions for CIRCUS
38
39 % I_TF = 200*69;

```

```
40 % I_VF = 205*54;
41 % I_QF = 513*56;
42 % I_wire = 0;
43 % Center_plasma = Major_R;
44 % wire_z_extra = 0;
45 % meshgrid_size = 0.007;
46 % meshgrid_points = 6;
    ↪
    ↪ %streamline meshgrid
47 % meshgrid_center = Major_R+0.008;
48 % Streamline_resolution = [0.01,999999];
49
50 % I_TF = 75*69;
51 % I_VF = 75*54;
52 % I_QF = 75*56;
53 % I_wire = 800;
54 % Center_plasma = Major_R+0.014;
55 % wire_z_extra = 0;
56 % meshgrid_size = 0.007;
57 % meshgrid_points = 6;
    ↪
    ↪ %streamline meshgrid
58 % meshgrid_center = Major_R+0.008;
59 % Streamline_resolution = [0.001,999999];
60 %
61 % [X0,Y0,Z0] = ndgrid(linspace(0.155, 0.175,6),0,0); %
    ↪ define tubes starting point
62
63
64 % I_TF = 75*69;
65 % I_VF = 75*54;
66 % I_QF = 75*56;
67 % I_wire = 1600;
68 % Center_plasma = Major_R+0.014;
69 % wire_z_extra = 0;
70 % meshgrid_size = 0.007;
71 % meshgrid_points = 6;
    ↪
    ↪ %streamline meshgrid
72 % meshgrid_center = Major_R+0.008;
73 % Streamline_resolution = [0.001,999999];
74
75 % I_TF = 75*69;
76 % I_VF = 75*54;
```

```
77 % I_QF = 75*56;
78 % I_wire = 2500;
79 % Center_plasma = Major_R;
80 % wire_z_extra = 0;
81 % meshgrid_size = 0.007;
82 % meshgrid_points = 6;
    ↪
    ↪ %streamline meshgrid
83 % meshgrid_center = Major_R+0.008;
84 % Streamline_resolution = [0.001,999999];
85 % [X0,Y0,Z0] = ndgrid(linspace(0.10, 0.155,6),0,0); %
    ↪ define tubes starting point
86
87
88 I_TF = 200*69;
89 I_VF = 205*54;
90 I_QF = 513*56;
91 I_wire = 60;
92 Center_plasma = Major_R+0.025;
93
94 Streamline_resolution = [0.001,999999];
95 [X0,Y0,Z0] = ndgrid(linspace(0.1610, 0.175,6),0,0);
    ↪ % define
    ↪ streamlines starting point
96
97
98
99
100
101
102
103
104 %%
105 teta=-pi:(pi/300):pi;
106
107
108 vf_x = vf_R*cos(teta);
109 vf_y = vf_R*sin(teta);
110 vf_z1 = vf_z + 0*teta;
111 vf_z2 = -vf_z + 0*teta;
112
113 qf_x = qf_R*cos(teta);
114 qf_y = qf_R*sin(teta);
115 qf_z1 = qf_z + 0*teta;
```

```

116 qf_z2 = -qf_z + 0*teta;
117
118 x1 = R*cos(phi1) + r*cos(teta)*cos(phi1) + r*sin(teta)
    ↪ *cos(aoa)*cos(phi1-(pi/2));
119 y1 = R*sin(phi1) + r*cos(teta)*sin(phi1) + r*sin(teta)
    ↪ *cos(aoa)*sin(phi1-(pi/2));
120 z1 = 0 + r*cos(teta)*0 + r*sin(teta)
    ↪ *sin(aoa);
121
122 x2 = R*cos(phi2) + r*cos(teta)*cos(phi2) + r*sin(teta)
    ↪ *cos(aoa)*cos(phi2-(pi/2));
123 y2 = R*sin(phi2) + r*cos(teta)*sin(phi2) + r*sin(teta)
    ↪ *cos(aoa)*sin(phi2-(pi/2));
124 z2 = 0 + r*cos(teta)*0 + r*sin(teta)
    ↪ *sin(aoa);
125
126 x3 = R*cos(phi3) + r*cos(teta)*cos(phi3) + r*sin(teta)
    ↪ *cos(aoa)*cos(phi3-(pi/2));
127 y3 = R*sin(phi3) + r*cos(teta)*sin(phi3) + r*sin(teta)
    ↪ *cos(aoa)*sin(phi3-(pi/2));
128 z3 = 0 + r*cos(teta)*0 + r*sin(teta)
    ↪ *sin(aoa);
129
130 x4 = R*cos(phi4) + r*cos(teta)*cos(phi4) + r*sin(teta)
    ↪ *cos(aoa)*cos(phi4-(pi/2));
131 y4 = R*sin(phi4) + r*cos(teta)*sin(phi4) + r*sin(teta)
    ↪ *cos(aoa)*sin(phi4-(pi/2));
132 z4 = 0 + r*cos(teta)*0 + r*sin(teta)
    ↪ *sin(aoa);
133
134 x5 = R*cos(phi5) + r*cos(teta)*cos(phi5) + r*sin(teta)
    ↪ *cos(aoa)*cos(phi5-(pi/2));
135 y5 = R*sin(phi5) + r*cos(teta)*sin(phi5) + r*sin(teta)
    ↪ *cos(aoa)*sin(phi5-(pi/2));
136 z5 = 0 + r*cos(teta)*0 + r*sin(teta)
    ↪ *sin(aoa);
137
138 % x6 = R*cos(phi6) + r*cos(teta)*cos(phi6) + r*sin(
    ↪ teta)*cos(aoa)*cos(phi6-(pi/2)); %6 TF
    ↪ coil CIRCUS
139 % y6 = R*sin(phi6) + r*cos(teta)*sin(phi6) + r*sin(
    ↪ teta)*cos(aoa)*sin(phi6-(pi/2)); %6 TF
    ↪ coil CIRCUS
140 % z6 = 0 + r*cos(teta)*0 + r*sin(

```

```

    ↪ teta)*sin(aoa); %6 TF
    ↪ coil CIRCUS
141
142
143 wire_x1 = Center_plasma*cos(teta);
144 wire_y1 = Center_plasma*sin(teta);
145 wire_z1 = 0*teta;
146
147
148
149 %%
150
151
152 Gamma1 = [qf_x;qf_y;qf_z1]';
153
154 I1 = -I_QF; % filament current [A]
    ↪ %Turn off if calculate
    ↪ force QF coil
155 dGamma1 = 1; % filament max discretization step [m]
    ↪ %Turn off if calculate force QF coil
156 [BSmag] = BSmag_add_filament(BSmag, Gamma1, I1, dGamma1);
    ↪ %Turn off if calculate force QF coil
157
158
159 Gamma2 = [qf_x;qf_y;qf_z2]';
160
161 I2 = -I_QF; % filament current [A]
162 dGamma2 = 1; % filament max discretization step [m]
163 [BSmag] = BSmag_add_filament(BSmag, Gamma2, I2, dGamma2);
164
165
166 Gamma3 = [vf_x;vf_y;vf_z1]';
167
168 I3 = -I_VF; % filament current [A]
169 dGamma3 = 1; % filament max discretization step [m]
170 [BSmag] = BSmag_add_filament(BSmag, Gamma3, I3, dGamma3);
171
172
173 Gamma4 = [vf_x;vf_y;vf_z2]';
174
175 I4 = -I_VF; % filament current [A]
    ↪ %Turn off if
    ↪ calculate force VF coil
176 dGamma4 = 1; % filament max discretization step [m]

```



```

    ↪           %Turn off if calculate force VF coil
177 [BSmag] = BSmag_add_filament (BSmag, Gamma4, I4, dGamma4);
    ↪           %Turn off if calculate force VF coil
178
179 Gamma5 = [x1;y1;z1]';
180
181 I5 = -I_TF; % filament current [A]
182 dGamma5 = 1; % filament max discretization step [m]
183 [BSmag] = BSmag_add_filament (BSmag, Gamma5, I5, dGamma5);
184
185 Gamma6 = [x2;y2;z2]';
186
187 I6 = -I_TF; % filament current [A]
188 dGamma6 = 1; % filament max discretization step [m]
189 [BSmag] = BSmag_add_filament (BSmag, Gamma6, I6, dGamma6);
190
191
192 Gamma7 = [x3;y3;z3]';
193
194 I7 = -I_TF; % filament current [A]
195 dGamma7 = 1; % filament max discretization step [m]
196 [BSmag] = BSmag_add_filament (BSmag, Gamma7, I7, dGamma7);
197
198 Gamma8 = [x4;y4;z4]';
199
200 I8 = -I_TF; % filament current [A]
201 dGamma8 = 1; % filament max discretization step [m]
202 [BSmag] = BSmag_add_filament (BSmag, Gamma8, I8, dGamma8);
203
204
205 Gamma9 = [x5;y5;z5]';
206
207 I9 = -I_TF; % filament current [A]
    ↪           %Turn off if calculate force
    ↪ TF coil
208 dGamma9 = 1; % filament max discretization step [m]
    ↪           %Turn off if calculate force TF coil
209 [BSmag] = BSmag_add_filament (BSmag, Gamma9, I9, dGamma9);
    ↪           %Turn off if calculate force TF coil
210
211
212 Gamma10 = [wire_x1;wire_y1;wire_z1]';
213
214 I10 = -I_wire; % filament current [A]

```

```

    ↪                               %Turn off if calculate no wire
215 dGamma10 = 1; % filament max discretization step [m]
    ↪                               %Turn off if calculate force no wire
216 [BSmag] = BSmag_add_filament(BSmag,Gamma10,I10,
    ↪ dGamma10); %Turn off if calculate force
    ↪ no wire
217
218 % Gamma11 = [x6;y6;z6]';
219 %
220 % I11 = -I_TF; % filament current [A]
    ↪                               %6 coil CIRCUS
221 % dGamma11 = 1; % filament max discretization step [m]
    ↪                               %6 coil CIRCUS
222 % [BSmag] = BSmag_add_filament(BSmag,Gamma11,I11,
    ↪ dGamma11); %6 coil CIRCUS
223
224
225 %%
226
227
228
229
230
231 % Field points (where we want to calculate the field)
232 % x_M = linspace(-0.23, 0.23, 61); % x [m]
233 % y_M = linspace(-0.23, 0.23, 61); % y [m]
234 % z_M = linspace(-0.06+0.0001, 0.06+0.0001, 21); % y [
    ↪ m]
235
236 % x_M = linspace(-0.23, 0.23, 101); % x [m]
237 % y_M = linspace(-0.23, 0.23, 101); % y [m]
238 % z_M = linspace(-0.06+0.0001, 0.06+0.0001, 31); % y [
    ↪ m]
239
240 x_M = linspace(-0.225, 0.225, 121); % x [m]
241 y_M = linspace(-0.225, 0.225, 121); % y [m]
242 z_M = linspace(-0.045+0.0001, 0.045+0.0001, 31); % y [
    ↪ m]
243 % z_M = linspace(-0.05+0.0001, 0.05+0.0001, 31); % y [
    ↪ m]
244
245 % cyl_linpoints = 31;
246 %
247 % for phi_cyl = 1:1:cyl_linpoints

```

```

248 %
249 % r_M_cyl = linspace(0.155, 0.22, cyl_linpoints); % x
      ↪ [m]
250 % z_M_cyl = linspace(-0.03+0.0001, 0.03+0.0001,
      ↪ cyl_linpoints); % y [m]
251 %
252 % [R_M_cyl,Z_M_cyl] = meshgrid(r_M_cyl,z_M_cyl);
253 %
254 % X_M(:,:,phi_cyl) = R_M_cyl*cos(phi_cyl*(2*pi)/
      ↪ cyl_linpoints);
255 % Y_M(:,:,phi_cyl) = R_M_cyl*sin(phi_cyl*(2*pi)/
      ↪ cyl_linpoints);
256 % Z_M(:,:,phi_cyl) = Z_M_cyl;
257 %
258 %
259 % end
260
261
262
263
264 [X_M,Y_M,Z_M]=meshgrid(x_M,y_M,z_M);
265 BSmag_plot_field_points(BSmag,X_M,Y_M,Z_M); % shows
      ↪ the field points volume
266
267 % Biot-Savart Integration
268 [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,X_M,Y_M,Z_M
      ↪ );
269
270 % Plot B/|B|
271 figure(1)
272     normB=sqrt(BX.^2+BY.^2+BZ.^2);
273     quiver3(X,Y,Z,BX./normB,BY./normB,BZ./normB,'b')
274 %axis tight
275
276 % Plot Bz on the volume
277 % figure(2), hold on, box on, grid on
278 % plot3(Gamma1(:,1),Gamma1(:,2),Gamma1(:,3),'-r') %
      ↪ plot filament
279 % plot3(Gamma2(:,1),Gamma2(:,2),Gamma2(:,3),'-r')
      ↪ % plot filament
280 % plot3(Gamma3(:,1),Gamma3(:,2),Gamma3(:,3),'-r')
      ↪ % plot filament
281 % plot3(Gamma4(:,1),Gamma4(:,2),Gamma4(:,3),'-r')
      ↪ % plot filament

```

```

282 % plot3(Gamma5(:,1),Gamma5(:,2),Gamma5(:,3),'-r') %
    ↪ plot filament
283 % plot3(Gamma6(:,1),Gamma6(:,2),Gamma6(:,3),'-r')
    ↪ % plot filament
284 % plot3(Gamma7(:,1),Gamma7(:,2),Gamma7(:,3),'-r')
    ↪ % plot filament
285 % plot3(Gamma8(:,1),Gamma8(:,2),Gamma8(:,3),'-r')
    ↪ % plot filament
286 % plot3(Gamma9(:,1),Gamma9(:,2),Gamma9(:,3),'-r')
    ↪ % plot filament
287 % plot3(Gamma10(:,1),Gamma10(:,2),Gamma10(:,3),'-r')
    ↪ r') % plot filament
288 % slice(X,Y,Z,BZ,[0],[],[-1,0,1]), colorbar % plot
    ↪ Bz
289 % xlabel('x [m]'), ylabel('y [m]'), zlabel('z [m]')
    ↪ , title('Bz [T]')
290 % view(3), axis equal, axis tight
291 % caxis([-0.5,0.5]*1e-5)
292
293 %% Plot some flux tubes
294 % figure(3), hold on, box on, grid on
295 % plot3(Gamma1(:,1),Gamma1(:,2),Gamma1(:,3),'-r') %
    ↪ plot filament
296 % plot3(Gamma2(:,1),Gamma2(:,2),Gamma2(:,3),'-r')
    ↪ % plot filament
297 % plot3(Gamma3(:,1),Gamma3(:,2),Gamma3(:,3),'-r')
    ↪ % plot filament
298 % plot3(Gamma4(:,1),Gamma4(:,2),Gamma4(:,3),'-r')
    ↪ % plot filament
299 % plot3(Gamma5(:,1),Gamma5(:,2),Gamma5(:,3),'-r') %
    ↪ plot filament
300 % plot3(Gamma6(:,1),Gamma6(:,2),Gamma6(:,3),'-r')
    ↪ % plot filament
301 % plot3(Gamma7(:,1),Gamma7(:,2),Gamma7(:,3),'-r')
    ↪ % plot filament
302 % plot3(Gamma8(:,1),Gamma8(:,2),Gamma8(:,3),'-r')
    ↪ % plot filament
303 % plot3(Gamma9(:,1),Gamma9(:,2),Gamma9(:,3),'-r')
    ↪ % plot filament
304 % plot3(Gamma10(:,1),Gamma10(:,2),Gamma10(:,3),'-r')
    ↪ r') % plot filament
305 % [X0,Y0,Z0] = ndgrid(-1.5:0.5:1.5,-1.5:0.5:1.5,-2)
    ↪ ; % define tubes starting point
306 % % [X0,Y0,Z0] = ndgrid(0.16:0.025:0.165,0,0); %

```

```

    ↪ define tubes starting point
307 % htubes = streamtube(stream3(X,Y,Z,BX,BY,BZ,X0,Y0,
    ↪ Z0), [0.2 10]);
308 % hstream = streamline(stream3(X,Y,Z,BX,BY,BZ,X0
    ↪ ,Y0,Z0), [0.1 10000]);
309 % xlabel('x [m]'), ylabel('y [m]'), zlabel('z [m]')
    ↪ , title('Some flux tubes')
310 % view(3), axis equal, axis tight
311 % set(htubes,'EdgeColor','none','FaceColor','c') %
    ↪ change tube color
312 % camlight left % change tube light
313
314 %% Plot By on the plane
315 figure(4), hold on, box on, grid on
316
317
318 fig4_Xpermute = permute(X(:,Linpoint_mid,:), [1 3 2]);
319 fig4_Zpermute = permute(Z(:,Linpoint_mid,:), [1 3 2]);
320 fig4_BZpermute = permute(BZ(:,Linpoint_mid,:), [1 3 2])
    ↪ ;
321
322 contour(fig4_Xpermute, fig4_Zpermute,
    ↪ fig4_BZpermute, [linspace(-0.045,0.045,20)]),
    ↪ colorbar
323
324 xp=Minor_a*cos(ang);
325 yp=Minor_a*sin(ang);
326 plot(Major_R+xp,yp);
327 plot(-Major_R+xp,yp);
328
329 xlabel('x [m]'), ylabel('z [m]') %, title('Contour
    ↪ plot Vertical Field [T]')
330 hold off
331
332 %% Plot magnetic field in section right half X axis BX
333 figure(5), hold on
334
335
336 X_section = X(Linpoint_mid:Linpoints,Linpoint_mid,
    ↪ Linpoint_mid);
337 Y_section = Y(Linpoint_mid:Linpoints,Linpoint_mid,
    ↪ Linpoint_mid);
338 Z_section = Z(Linpoint_mid:Linpoints,Linpoint_mid,
    ↪ Linpoint_mid);

```

```

339 BX_section = BX(Linpoint_mid:Linpoints, Linpoint_mid,
    ↪ Linpoint_mid);
340 BY_section = BY(Linpoint_mid:Linpoints, Linpoint_mid,
    ↪ Linpoint_mid);
341 BZ_section = BZ(Linpoint_mid:Linpoints, Linpoint_mid,
    ↪ Linpoint_mid);
342
343 plot(X_section , BX_section)
344
345 plot([Major_R-Minor_a Major_R-Minor_a],[-0.15 0.15])
346 plot([Major_R+Minor_a Major_R+Minor_a],[-0.15 0.15])
347
348 [minValue, closestIndex] = min(abs(X_section - Major_R
    ↪ )); %Find closest point to major_r along
    ↪ x_section
349
350
351
352 plot(X_section(closestIndex), BX_section(closestIndex),
    ↪ 'r*')
353 text(X_section(closestIndex)+0.001 , BX_section(
    ↪ closestIndex)+0.001 , [ '(' , num2str(X_section(
    ↪ closestIndex)), ', ', num2str(BX_section(
    ↪ closestIndex)), ') '])
354 xlabel ('x [m]'), ylabel ('B [T]')%, title ('BX
    ↪ section right half X axis')
355 ylim([-0.15 0.15])
356 hold off
357
358 %% Plot magnetic field in section right half X axis BY
359 figure(6), hold on
360 set(gcf, 'Position', [100, 100, 500/1.2, 400/1.2])
361
362 plot(X_section , BY_section)
363
364 plot([Major_R-Minor_a Major_R-Minor_a],[-0.4 0.5])
365 plot([Major_R+Minor_a Major_R+Minor_a],[-0.4 0.5])
366
367 plot(X_section(closestIndex), BY_section(closestIndex),
    ↪ 'r*')
368 %text(X_section(closestIndex)+0.001 , BY_section(
    ↪ closestIndex)+0.001 , [ '(' , num2str(X_section(
    ↪ closestIndex)), ', ', num2str(BY_section(
    ↪ closestIndex)), ') '])

```

```

369 xlabel ('x [m]'), ylabel ('B [T]')%, title ('BY
    ↪ section right half X axis')
370 ylim([-0.42 0.5])
371
372 hold off
373
374 %% Plot magnetic field in section left half X axis BY
375 figure(7), hold on
376 set(gcf, 'Position', [100, 100, 500/1.2, 400/1.2])
377
378
379 X2_section = X(1:Linpoint_mid, Linpoint_mid,
    ↪ Linpoint_mid);
380 BY2_section = BY(1:Linpoint_mid, Linpoint_mid,
    ↪ Linpoint_mid);
381 BZ2_section = BZ(1:Linpoint_mid, Linpoint_mid,
    ↪ Linpoint_mid);
382
383 plot(X2_section , BY2_section)
384
385 plot([-Major_R-Minor_a -Major_R-Minor_a],[-0.6 0.3])
386 plot([-Major_R+Minor_a -Major_R+Minor_a],[-0.6 0.3])
387
388
389 [minValue, closestIndex2] = min(abs(X2_section +
    ↪ Major_R)); %Find closest point to major_r along
    ↪ x_section
390
391 plot(X2_section(closestIndex2), BY2_section(
    ↪ closestIndex2), 'r*')
392 %text(X2_section(closestIndex2)+0.001 , BY2_section(
    ↪ closestIndex2)+0.001 , [ '(' , num2str(X2_section(
    ↪ closestIndex2)), ', ', num2str(BY2_section(
    ↪ closestIndex2)), ') '])
393 xlabel ('x [m]'), ylabel ('B [T]')%, title ('BY
    ↪ section left half X axis')
394 ylim([-0.6 0.3])
395 hold off
396
397
398 %% Plot magnetic field in section right half X axis
399 figure(8), hold on
400 set(gcf, 'Position', [100, 100, 500/1.2, 400/1.2])
401

```

```
402
403 plot(X_section , BZ_section)
404 plot([Major_R-Minor_a Major_R-Minor_a],[-0.55 0.35])
405 plot([Major_R+Minor_a Major_R+Minor_a],[-0.55 0.35])
406
407 plot(X_section(closestIndex),BZ_section(closestIndex),
      ↪ 'r*')
408 %text(X_section(closestIndex)+0.001 , BZ_section(
      ↪ closestIndex)+0.001 , [ '(' ,num2str(X_section(
      ↪ closestIndex)),',',',num2str(BZ_section(
      ↪ closestIndex)),',')'])
409
410
411
412 xlabel ('x [m]'), ylabel ('B [T]')%, title ('BZ
      ↪ section right half X axis')
413 ylim([-0.55 0.35])
414 hold off
415
416
417 %% Plot magnetic field in section left half X axis
418 figure(9), hold on
419 set(gcf, 'Position', [100, 100, 500/1.2, 400/1.2])
420
421
422 plot(X2_section , BZ2_section)
423 plot([-Major_R-Minor_a -Major_R-Minor_a],[-0.1 0.8])
424 plot([-Major_R+Minor_a -Major_R+Minor_a],[-0.1 0.8])
425
426 plot(X2_section(closestIndex2),BZ2_section(
      ↪ closestIndex2),'r*')
427 %text(X2_section(closestIndex2)+0.001 , BZ2_section(
      ↪ closestIndex2)+0.001 , [ '(' ,num2str(X2_section(
      ↪ closestIndex2)),',',',num2str(BZ2_section(
      ↪ closestIndex2)),',')'])
428
429
430
431 xlabel ('x [m]'), ylabel ('B [T]')%, title ('BZ
      ↪ section left half X axis')
432 ylim([-0.1 0.8])
433 hold off
434
435
```



```
436 %% Plot magnetic field streamline 3D
437 figure(12), hold on
438
439 axis equal
440
441 % [X0,Y0,Z0] = meshgrid(meshgrid_x,meshgrid_y,
    ↪ meshgrid_z);
442 % [X0,Y0,Z0] = meshgrid
    ↪ (0.001,0.143:0.01:0.1930,-0.025:0.01:0.025);
443
444 % permute_vector = [2 1 3];
445 permute_vector = [1 2 3];
446 data_x = permute(X,permute_vector);
447 data_y = permute(Y,permute_vector);
448 data_z = permute(Z,permute_vector);
449
450 data_bx = permute(BX,permute_vector);
451 data_by = permute(BY,permute_vector);
452 data_bz = permute(BZ,permute_vector);
453
454
455 %%
456 for t = 1:1:20
457     t
458
459     format longE
460
461     figure(12), hold on
462
463     axis equal
464     % quiver3(data_x,data_y,data_z,data_bx,data_by,data_bz
    ↪ )
465     %streamline(stream3(data_x,data_y,data_z,data_bx,
    ↪ data_by,data_bz,X0,Y0,Z0,Streamline_resolution))
466     Streamline3D = streamline(stream3(data_x,data_y,data_z
    ↪ ,data_bx,data_by,data_bz,X0,Y0,Z0,
    ↪ Streamline_resolution));
467     % Streamline3D = streamline(stream3(X_M,Y_M,Z_M,
    ↪ data_bx,data_by,data_bz,X0,Y0,Z0,
    ↪ Streamline_resolution));
468     % Streamline3D = streamline(stream3(X_M,Y_M,Z_M,BZ,BY,
    ↪ BZ,X0,Y0,Z0,Streamline_resolution));
469
470 % Draw coils
```

```

471 line(Gamma1(:,1)', Gamma1(:,2)', Gamma1(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
472 line(Gamma2(:,1)', Gamma2(:,2)', Gamma2(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
473 line(Gamma3(:,1)', Gamma3(:,2)', Gamma3(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
474 line(Gamma4(:,1)', Gamma4(:,2)', Gamma4(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
475 line(Gamma5(:,1)', Gamma5(:,2)', Gamma5(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
476 line(Gamma6(:,1)', Gamma6(:,2)', Gamma6(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
477 line(Gamma7(:,1)', Gamma7(:,2)', Gamma7(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
478 line(Gamma8(:,1)', Gamma8(:,2)', Gamma8(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
479 line(Gamma9(:,1)', Gamma9(:,2)', Gamma9(:,3)', 'Color', '
    ↪ red', 'LineWidth', 2)
480 line(Gamma10(:,1)', Gamma10(:,2)', Gamma10(:,3)', 'Color'
    ↪ , 'red', 'LineWidth', 2)
481
482 xlabel('x [m]'), ylabel('y [m]'), zlabel('z [m]')%,
    ↪ title('3D Streamline plot')
483
484
485
486
487 % Poincare x-z plane in 3D plot
488
489 n = length(Streamline3D);
490
491
492
493 for n = 1:n
494 figure(12), hold on
495
496 Streamline_x_data= Streamline3D(n).XData; % * my
    ↪ signal input* it's like sinusoidal signal
497 Streamline_y_data= Streamline3D(n).YData; % * my
    ↪ signal input* it's like sinusoidal signal
498 Streamline_z_data= Streamline3D(n).ZData; % * my
    ↪ signal input* it's like sinusoidal signal
499
500 Streamline_datapoints=1:length(Streamline_y_data);

```

```

501 Streamline_n=length(Streamline_y_data);
    ↪
    ↪ % Length of my signal
502 Streamline_t = (0:(Streamline_n-1));
    ↪
    ↪ % Time appointed for every sample, first sample
    ↪ at time t=0 , next tn=t0+n*(1/fs)
503 Streamline_datapoints=Streamline_t;
504 zci = @(v) find(v(:).*circshift(v(:), [-1 0]) <= 0);
505 % Returns Approximate Zero-Crossing Indices Of
    ↪ Argument Vector
506
507 Poincare_points = zci(Streamline_y_data);
508 if length(Poincare_points) == 0
509
510 else
511 Poincare_points(end) = [];
    ↪ %Remomves
    ↪ last point, since end of vector
512 end
513
514 scatter3(Streamline_x_data(Poincare_points),
    ↪ Streamline_y_data(Poincare_points),
    ↪ Streamline_z_data(Poincare_points), 'm*')
515 view(30,30)
516 hold off
517
518
519 % Poincare x-z plane in 3D plot
520 figure(13)
521
522 hold on
523 set(gca,'FontSize',20)
524
525 xlim([0.14 0.21])
526 % xlim([-0.22 -0.14])
527 % ylim([-0.025 0.025])
528 ylim([-0.035 0.035])
529 yticks(-0.05:(0.1/10):0.05);
530
531 % xp=Minor_a*cos(ang);
532 % yp=Minor_a*sin(ang);
533 %plot(Major_R+xp,yp);
534 %plot(-Major_R+xp,yp);

```

```
535
536 if n == 1 | n == 1+6 | n == 1+12 | n == 1+18 | n ==
    ↪ 1+24 | n == 1+30
537 plot(Streamline_x_data(Poincare_points),
    ↪ Streamline_z_data(Poincare_points), 'b.')
538 if t == 1
539 h(1) = scatter(X0(n), Z0(n), 'b*');
540 end
541 elseif n == 2 | n == 2+6 | n == 2+12 | n == 2+18 | n
    ↪ == 2+24 | n == 2+30
542 plot(Streamline_x_data(Poincare_points),
    ↪ Streamline_z_data(Poincare_points), 'y.')
543 if t == 1
544 h(2) = scatter(X0(n), Z0(n), 'y*');
545 end
546 elseif n == 3 | n == 3+6 | n == 3+12 | n == 3+18 | n
    ↪ == 3+24 | n == 3+30
547 plot(Streamline_x_data(Poincare_points),
    ↪ Streamline_z_data(Poincare_points), 'r.')
548 if t == 1
549 h(3) = scatter(X0(n), Z0(n), 'r*');
550 end
551 elseif n == 4 | n == 4+6 | n == 4+12 | n == 4+18 | n
    ↪ == 4+24 | n == 4+30
552 plot(Streamline_x_data(Poincare_points),
    ↪ Streamline_z_data(Poincare_points), 'g.')
553 if t == 1
554 h(4) = scatter(X0(n), Z0(n), 'g*');
555 end
556 elseif n == 5 | n == 5+6 | n == 5+12 | n == 5+18 | n
    ↪ == 5+24 | n == 5+30
557 plot(Streamline_x_data(Poincare_points),
    ↪ Streamline_z_data(Poincare_points), 'm.')
558 if t == 1
559 h(5) = scatter(X0(n), Z0(n), 'm*');
560 end
561 else
562 plot(Streamline_x_data(Poincare_points),
    ↪ Streamline_z_data(Poincare_points), 'c.')
563 if t == 1
564 h(6) = scatter(X0(n), Z0(n), 'c*');
565 end
566 end
567
```

```

568
569 xlabel ('x [m]'), ylabel ('z [m]')%, title ('Poincar
    ↪ fluxlines')
570
571 end
572 legend(h, {'0.1610', '0.1638 ', '0.1666', '0.1694 ', '
    ↪ 0.1722', '0.1750'}, 'Location', 'northeast', '
    ↪ FontSize', 12)
573 hold off
574
575
576 % Poincare all points x-z plane in 3D plot
577 figure(14)
578
579 hold on
580 set(gca, 'fontweight', 'bold', 'FontSize', 14)
581 xlim([0.08 0.22])
582 xticks(0.08:0.02:0.22);
583 ylim([-0.05 0.05])
584 yticks(-0.05:0.01:0.05);
585 set(gca, 'DataAspectRatio', [10 10 1])
586
587 % xp=Minor_a*cos(ang);
588 % yp=Minor_a*sin(ang);
589 % plot(Major_R+xp,yp);
590 % plot(-Major_R+xp,yp);
591
592 n = 1;
593 plot(sqrt((Streamline3D(n).XData).^2 + (Streamline3D(n)
    ↪ ).YData).^2), Streamline3D(n).ZData, 'b.')
594 if t == 1
595 scatter(X0(n), Z0(n), 'b*')
596 end
597 n = 2;
598 plot(sqrt((Streamline3D(n).XData).^2 + (Streamline3D(n)
    ↪ ).YData).^2), Streamline3D(n).ZData, 'y.')
599 if t == 1
600 scatter(X0(n), Z0(n), 'y*')
601 end
602 n = 3;
603 plot(sqrt((Streamline3D(n).XData).^2 + (Streamline3D(n)
    ↪ ).YData).^2), Streamline3D(n).ZData, 'r.')
604 if t == 1
605 scatter(X0(n), Z0(n), 'r*')

```

```

606 end
607 n = 4;
608 plot(sqrt((Streamline3D(n).XData).^2 + (Streamline3D(n)
    ↪ ).YData).^2), Streamline3D(n).ZData, 'g.')
609 if t == 1
610 scatter(X0(n), Z0(n), 'g*')
611 end
612 n = 5;
613 plot(sqrt((Streamline3D(n).XData).^2 + (Streamline3D(n)
    ↪ ).YData).^2), Streamline3D(n).ZData, 'm.')
614 if t == 1
615 scatter(X0(n), Z0(n), 'm*')
616 end
617 n = 6;
618 plot(sqrt((Streamline3D(n).XData).^2 + (Streamline3D(n)
    ↪ ).YData).^2), Streamline3D(n).ZData, 'c.')
619 if t == 1
620 scatter(X0(n), Z0(n), 'c*')
621 end
622
623 xlabel('x [m]'), ylabel('z [m]')%, title('Poincar
    ↪ fluxlines')
624
625 hold off
626
627 % X0 = [Streamline3D(1).XData(Streamline_resolution
    ↪ (2));Streamline3D(2).XData(Streamline_resolution
    ↪ (2));Streamline3D(3).XData(Streamline_resolution
    ↪ (2))];
628 % Y0 = [Streamline3D(1).YData(Streamline_resolution
    ↪ (2));Streamline3D(2).YData(Streamline_resolution
    ↪ (2));Streamline3D(3).YData(Streamline_resolution
    ↪ (2))];
629 % Z0 = [Streamline3D(1).ZData(Streamline_resolution
    ↪ (2));Streamline3D(2).ZData(Streamline_resolution
    ↪ (2));Streamline3D(3).ZData(Streamline_resolution
    ↪ (2))];
630
631 % X0 = [X0(1);X0(2);X0(3);X0(4);X0(1);Streamline3D(6)
    ↪ .XData(Streamline_resolution(2))];
632 % Y0 = [Y0(1);Y0(2);Y0(3);Y0(4);Streamline3D(5).YData
    ↪ (Streamline_resolution(2));Streamline3D(6).YData
    ↪ (Streamline_resolution(2))];
633 % Z0 = [Z0(1);Z0(2);Z0(3);Z0(4);Streamline3D(5).ZData

```

```
        ↪ (Streamline_resolution(2));Streamline3D(6).ZData
        ↪ (Streamline_resolution(2))];
634 % X0 = [X0(1);X0(2);X0(3);X0(4);X0(5);Streamline3D(6)
        ↪ .XData(Streamline_resolution(2))];
635 % Y0 = [Y0(1);Y0(2);Y0(3);Y0(4);Y0(5);Streamline3D(6)
        ↪ .YData(Streamline_resolution(2))];
636 % Z0 = [Z0(1);Z0(2);Z0(3);Z0(4);Z0(5);Streamline3D(6)
        ↪ .ZData(Streamline_resolution(2))];
637
638 % if length(Streamline3D(1).YData) ==
        ↪ Streamline_resolution(2)
639 % X0_1 = Streamline3D(1).XData(Streamline_resolution
        ↪ (2));
640 % Y0_1 = Streamline3D(1).YData(Streamline_resolution
        ↪ (2));
641 % Z0_1 = Streamline3D(1).ZData(Streamline_resolution
        ↪ (2));
642 % else
643 %     X0_1 = X0(1);
644 %     Y0_1 = Y0(1);
645 %     Z0_1 = Z0(1);
646 % end
647 %
648 % if length(Streamline3D(2).YData) ==
        ↪ Streamline_resolution(2)
649 % X0_2 = Streamline3D(2).XData(Streamline_resolution
        ↪ (2));
650 % Y0_2 = Streamline3D(2).YData(Streamline_resolution
        ↪ (2));
651 % Z0_2 = Streamline3D(2).ZData(Streamline_resolution
        ↪ (2));
652 % else
653 %     X0_2 = X0(2);
654 %     Y0_2 = Y0(2);
655 %     Z0_2 = Z0(2);
656 % end
657 %
658 % if length(Streamline3D(3).YData) ==
        ↪ Streamline_resolution(2)
659 % X0_3 = Streamline3D(3).XData(Streamline_resolution
        ↪ (2));
660 % Y0_3 = Streamline3D(3).YData(Streamline_resolution
        ↪ (2));
661 % Z0_3 = Streamline3D(3).ZData(Streamline_resolution
```

```
        ↪ (2));
662 % else
663 %     X0_3 = X0(3);
664 %     Y0_3 = Y0(3);
665 %     Z0_3 = Z0(3);
666 % end
667 %
668 % if length(Streamline3D(4).YData) ==
        ↪ Streamline_resolution(2)
669 % X0_4 = Streamline3D(4).XData(Streamline_resolution
        ↪ (2));
670 % Y0_4 = Streamline3D(4).YData(Streamline_resolution
        ↪ (2));
671 % Z0_4 = Streamline3D(4).ZData(Streamline_resolution
        ↪ (2));
672 % else
673 %     X0_4 = X0(4);
674 %     Y0_4 = Y0(4);
675 %     Z0_4 = Z0(4);
676 % end
677 %
678 % if length(Streamline3D(5).YData) ==
        ↪ Streamline_resolution(2)
679 % X0_5 = Streamline3D(5).XData(Streamline_resolution
        ↪ (2));
680 % Y0_5 = Streamline3D(5).YData(Streamline_resolution
        ↪ (2));
681 % Z0_5 = Streamline3D(5).ZData(Streamline_resolution
        ↪ (2));
682 % else
683 %     X0_5 = X0(5);
684 %     Y0_5 = Y0(5);
685 %     Z0_5 = Z0(5);
686 % end
687 %
688 %
689 %
690 % if length(Streamline3D(6).YData) ==
        ↪ Streamline_resolution(2)
691 % X0_6 = Streamline3D(6).XData(Streamline_resolution
        ↪ (2));
692 % Y0_6 = Streamline3D(6).YData(Streamline_resolution
        ↪ (2));
693 % Z0_6 = Streamline3D(6).ZData(Streamline_resolution
```



```
        ↪ (2));
694 % else
695 %     X0_6 = X0(6);
696 %     Y0_6 = Y0(6);
697 %     Z0_6 = Z0(6);
698 % end
699 %
700 %
701 % X0 = [X0_1;X0_2;X0_3;X0_4;X0_5;X0_6];
702 % Y0 = [Y0_1;Y0_2;Y0_3;Y0_4;Y0_5;Y0_6];
703 % Z0 = [Z0_1;Z0_2;Z0_3;Z0_4;Z0_5;Z0_6];
704
705
706
707 X0 = [Streamline3D(1).XData(Streamline_resolution(2))
        ↪ ;Streamline3D(2).XData(Streamline_resolution(2))
        ↪ ;Streamline3D(3).XData(Streamline_resolution(2))
        ↪ ;Streamline3D(4).XData(Streamline_resolution(2))
        ↪ ;Streamline3D(5).XData(Streamline_resolution(2))
        ↪ ;Streamline3D(6).XData(Streamline_resolution(2))
        ↪ ];
708 Y0 = [Streamline3D(1).YData(Streamline_resolution(2))
        ↪ ;Streamline3D(2).YData(Streamline_resolution(2))
        ↪ ;Streamline3D(3).YData(Streamline_resolution(2))
        ↪ ;Streamline3D(4).YData(Streamline_resolution(2))
        ↪ ;Streamline3D(5).YData(Streamline_resolution(2))
        ↪ ;Streamline3D(6).YData(Streamline_resolution(2))
        ↪ ];
709 Z0 = [Streamline3D(1).ZData(Streamline_resolution(2))
        ↪ ;Streamline3D(2).ZData(Streamline_resolution(2))
        ↪ ;Streamline3D(3).ZData(Streamline_resolution(2))
        ↪ ;Streamline3D(4).ZData(Streamline_resolution(2))
        ↪ ;Streamline3D(5).ZData(Streamline_resolution(2))
        ↪ ;Streamline3D(6).ZData(Streamline_resolution(2))
        ↪ ];
710
711 % X0 = [Streamline3D(1).XData(Streamline_resolution
        ↪ (2));Streamline3D(2).XData(Streamline_resolution
        ↪ (2));Streamline3D(3).XData(Streamline_resolution
        ↪ (2));X0(4);X0(5);X0(6)];
712 % Y0 = [Streamline3D(1).YData(Streamline_resolution
        ↪ (2));Streamline3D(2).YData(Streamline_resolution
        ↪ (2));Streamline3D(3).YData(Streamline_resolution
        ↪ (2));Y0(4);Y0(5);Y0(6)];
```

```

713 % Z0 = [Streamline3D(1).ZData(Streamline_resolution
      ↪ (2));Streamline3D(2).ZData(Streamline_resolution
      ↪ (2));Streamline3D(3).ZData(Streamline_resolution
      ↪ (2));Z0(4);Z0(5);Z0(6)];
714 %
715 end
716
717
718
719
720 %% Force TF coil
721 % % Turn off Gamma5 = TF_5 coil in section Calculate
      ↪ magnetic field
722 %
723 % % Center point VF_5 coil
724 % % CP_x = R*cos(phi1);
725 % % CP_y = R*sin(phi1);
726 % % CP_z = 0;
727 % %
728 % % x_diff = diff(x1);
729 % % y_diff = diff(y1);
730 % % z_diff = diff(z1);
731 %
732 % CP_x = R*cos(phi5);
733 % CP_y = R*sin(phi5);
734 % CP_z = 0;
735 %
736 % x_diff = diff(x5);
737 % y_diff = diff(y5);
738 % z_diff = diff(z5);
739 %
740 % x_diff(601)= x_diff(1);
741 % y_diff(601)= y_diff(1);
742 % z_diff(601)= z_diff(1);
743 %
744 % [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,x1,y1,z1)
      ↪ ;
745 % % [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,x5,y5,
      ↪ z5);
746 % normB=sqrt(BX.^2+BY.^2+BZ.^2);
747 %
748 % %Perimeter_vf_coil_element = 2*pi*vf_R/60;
749 % %E = sqrt(x_diff.^2 + y_diff.^2 + z_diff.^2);
750 % %B=[x_diff.*(Perimeter_vf_coil_element./E);y_diff.*(

```

```

    ↪ Perimeter_vf_coil_element./E);z_diff.*(
    ↪ Perimeter_vf_coil_element./E)]'*I_VF;
751 % A=[BX;BY;BZ]';
752 % B=[x_diff.*I_TF;y_diff.*I_TF;z_diff.*I_TF]';
753 % C = cross(B,A);
754 %
755 % C_center = sum(C(1:600,:))
756 % Mean_force=mean(sqrt(C(:,1).^2 + C(:,2).^2 + C(:,3)
    ↪ .^2)) %mean force per element
757 % normC=sqrt(C_center(1)^2+C_center(2)^2+C_center(3)
    ↪ ^2)
758 %
759 % figure(15)
760 % hold on
761 % axis equal
762 %
763 % quiver3(X,Y,Z,BX./normB,BY./normB,BZ./normB,'b')
764 % quiver3(X,Y,Z,x_diff,y_diff,z_diff,'y')
765 % quiver3(X,Y,Z,C(:,1)',C(:,2)',C(:,3)', 'g')
766 % quiver3(CP_x,CP_y,CP_z,0.25*C_center(1)/normC,0.25*
    ↪ C_center(2)/normC,0.25*C_center(3)/normC,'k',
    ↪ LineWidth',2)
767 %
768 % Draw coils
769 % line(Gamma1(:,1)',Gamma1(:,2)',Gamma1(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
770 % line(Gamma2(:,1)',Gamma2(:,2)',Gamma2(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
771 % line(Gamma3(:,1)',Gamma3(:,2)',Gamma3(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
772 % line(Gamma4(:,1)',Gamma4(:,2)',Gamma4(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
773 % line(Gamma5(:,1)',Gamma5(:,2)',Gamma5(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
774 % line(Gamma6(:,1)',Gamma6(:,2)',Gamma6(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
775 % line(Gamma7(:,1)',Gamma7(:,2)',Gamma7(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
776 % line(Gamma8(:,1)',Gamma8(:,2)',Gamma8(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
777 % line(Gamma9(:,1)',Gamma9(:,2)',Gamma9(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)
778 % line(Gamma10(:,1)',Gamma10(:,2)',Gamma10(:,3)',
    ↪ Color', 'red', 'LineWidth',2)

```

```

779 % view(3), axis equal, axis tight
780 %
781 % hold off
782
783
784
785
786 %% Force VF coil
787 % % Turn off Gamma8 = VF_1 coil in section Calculate
    ↪ magnetic field
788 %
789 % %Center point VF_1 coil
790 % CP_x = 0;
791 % CP_y = 0;
792 % CP_z = -vf_z;
793 %
794 % x_diff = diff(vf_x);
795 % y_diff = diff(vf_y);
796 % z_diff = diff(vf_z2);
797 %
798 % x_diff(601)= x_diff(1);
799 % y_diff(601)= y_diff(1);
800 % z_diff(601)= z_diff(1);
801 %
802 % [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,vf_x,vf_y
    ↪ ,vf_z2);
803 % normB=sqrt(BX.^2+BY.^2+BZ.^2);
804 %
805 % %Perimeter_vf_coil_element = 2*pi*vf_R/60;
806 % %E = sqrt(x_diff.^2 + y_diff.^2 + z_diff.^2);
807 % %B=[x_diff.*(Perimeter_vf_coil_element./E);y_diff.*(
    ↪ Perimeter_vf_coil_element./E);z_diff.*(
    ↪ Perimeter_vf_coil_element./E)]'*I_VF;
808 % A=[BX;BY;BZ]';
809 % B=[x_diff.*I_VF;y_diff.*I_VF;z_diff.*I_VF]';
810 % C = cross(B,A);
811 %
812 % C_center = sum(C(1:600,:))
813 % Mean_force=mean(sqrt(C(:,1).^2 + C(:,2).^2 + C(:,3)
    ↪ .^2)) %mean force per element
814 % normC=sqrt(C_center(1)^2+C_center(2)^2+C_center(3)
    ↪ ^2);
815 %
816 % figure(16)

```

```

817 % hold on
818 % axis equal
819 %
820 % quiver3(X,Y,Z,BX./normB,BY./normB,BZ./normB,'b')
821 % quiver3(X,Y,Z,x_diff,y_diff,z_diff,'y')
822 % quiver3(X,Y,Z,C(:,1)',C(:,2)',C(:,3)', 'g')
823 % quiver3(CP_x,CP_y,CP_z,0.25*C_center(1)/normC,0.25*
      ↪ C_center(2)/normC,0.25*C_center(3)/normC,'k',
      ↪ LineWidth',2)
824 %
825 % % Draw coils
826 % line(Gamma1(:,1)',Gamma1(:,2)',Gamma1(:,3)', 'Color
      ↪ ','red','LineWidth',2)
827 % line(Gamma2(:,1)',Gamma2(:,2)',Gamma2(:,3)', 'Color
      ↪ ','red','LineWidth',2)
828 % line(Gamma3(:,1)',Gamma3(:,2)',Gamma3(:,3)', 'Color
      ↪ ','red','LineWidth',2)
829 % line(Gamma4(:,1)',Gamma4(:,2)',Gamma4(:,3)', 'Color
      ↪ ','red','LineWidth',2)
830 % line(Gamma5(:,1)',Gamma5(:,2)',Gamma5(:,3)', 'Color
      ↪ ','red','LineWidth',2)
831 % line(Gamma6(:,1)',Gamma6(:,2)',Gamma6(:,3)', 'Color
      ↪ ','red','LineWidth',2)
832 % line(Gamma7(:,1)',Gamma7(:,2)',Gamma7(:,3)', 'Color
      ↪ ','red','LineWidth',2)
833 % line(Gamma8(:,1)',Gamma8(:,2)',Gamma8(:,3)', 'Color
      ↪ ','red','LineWidth',2)
834 % line(Gamma9(:,1)',Gamma9(:,2)',Gamma9(:,3)', 'Color
      ↪ ','red','LineWidth',2)
835 % line(Gamma10(:,1)',Gamma10(:,2)',Gamma10(:,3)',
      ↪ Color','red','LineWidth',2)
836 % view(3), axis equal, axis tight
837 % hold off
838
839
840
841
842
843 %% Force QF coil
844 % %Turn off Gamma6 = QF_1 coil in section Calculate
      ↪ magnetic field
845 %
846 % % Center point QF_1 coil
847 % CP_x = 0;

```

```

848 % CP_y = 0;
849 % CP_z = qf_z;
850 %
851 % x_diff = diff(qf_x);
852 % y_diff = diff(qf_y);
853 % z_diff = diff(qf_z1);
854 %
855 % x_diff(601)= x_diff(1);
856 % y_diff(601)= y_diff(1);
857 % z_diff(601)= z_diff(1);
858 %
859 % [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,qf_x,qf_y
    ↪ ,qf_z1);
860 % normB=sqrt(BX.^2+BY.^2+BZ.^2);
861 %
862 % %Perimeter_qf_coil_element = 2*pi*qf_R/60;
863 % %E = sqrt(x_diff.^2 + y_diff.^2 + z_diff.^2);
864 % %B=[x_diff.*(Perimeter_qf_coil_element./E);y_diff.*(
    ↪ Perimeter_qf_coil_element./E);z_diff.*(
    ↪ Perimeter_qf_coil_element./E)]'*I_QF;
865 % A=[BX;BY;BZ]';
866 % B=[x_diff.*I_QF;y_diff.*I_QF;z_diff.*I_QF]';
867 % C = cross(B,A);
868 %
869 % C_center = sum(C(1:600,:))
870 % Mean_force=mean(sqrt(C(:,1).^2 + C(:,2).^2 + C(:,3)
    ↪ .^2)) %mean force per element
871 % normC=sqrt(C_center(1)^2+C_center(2)^2+C_center(3)
    ↪ ^2);
872 %
873 % figure(17)
874 % hold on
875 % axis equal
876 %
877 % quiver3(X,Y,Z,BX./normB,BY./normB,BZ./normB,'b')
878 % quiver3(X,Y,Z,x_diff,y_diff,z_diff,'y')
879 % quiver3(X,Y,Z,C(:,1)',C(:,2)',C(:,3)', 'g')
880 % quiver3(CP_x,CP_y,CP_z,0.25*C_center(1)/normC,0.25*
    ↪ C_center(2)/normC,0.25*C_center(3)/normC,'k', '
    ↪ LineWidth',2)
881 %
882 % % Draw coils
883 % line(Gamma1(:,1)',Gamma1(:,2)',Gamma1(:,3)', 'Color
    ↪ ', 'red', 'LineWidth',2)

```

```

884 % line(Gamma2(:,1)',Gamma2(:,2)',Gamma2(:,3)','','Color
      ↪ ','red','LineWidth',2)
885 % line(Gamma3(:,1)',Gamma3(:,2)',Gamma3(:,3)','','Color
      ↪ ','red','LineWidth',2)
886 % line(Gamma4(:,1)',Gamma4(:,2)',Gamma4(:,3)','','Color
      ↪ ','red','LineWidth',2)
887 % line(Gamma5(:,1)',Gamma5(:,2)',Gamma5(:,3)','','Color
      ↪ ','red','LineWidth',2)
888 % line(Gamma6(:,1)',Gamma6(:,2)',Gamma6(:,3)','','Color
      ↪ ','red','LineWidth',2)
889 % line(Gamma7(:,1)',Gamma7(:,2)',Gamma7(:,3)','','Color
      ↪ ','red','LineWidth',2)
890 % line(Gamma8(:,1)',Gamma8(:,2)',Gamma8(:,3)','','Color
      ↪ ','red','LineWidth',2)
891 % line(Gamma9(:,1)',Gamma9(:,2)',Gamma9(:,3)','','Color
      ↪ ','red','LineWidth',2)
892 % line(Gamma10(:,1)',Gamma10(:,2)',Gamma10(:,3)','','
      ↪ Color','red','LineWidth',2)
893 % view(3), axis equal, axis tight
894 % hold off
895
896 %% Force poloidal wire
897 %Turn off coil in section Calculate magnetic field
898
899
900
901 % %Center point poloidal wire
902 % CP_x = 0;
903 % CP_y = 0;
904 % CP_z = 0;
905 %
906 % x_diff = diff(wire_x1);
907 % y_diff = diff(wire_y1);
908 % z_diff = diff(wire_z1);
909 %
910 % x_diff(601)= x_diff(1);
911 % y_diff(601)= y_diff(1);
912 % z_diff(601)= z_diff(1);
913 %
914 % [BSmag,X,Y,Z,BX,BY,BZ] = BSmag_get_B(BSmag,wire_x1,
      ↪ wire_y1,wire_z1);
915 % normB=sqrt(BX.^2+BY.^2+BZ.^2);
916 %
917 % A=[BX;BY;BZ]';

```

```

918 % B=[x_diff.*I_wire;y_diff.*I_wire;z_diff.*I_wire]';
919 % C = cross(B,A);
920 %
921 % C_center = sum(C(1:600,:))
922 % Mean_force=mean(sqrt(C(:,1).^2 + C(:,2).^2 + C(:,3)
    ↪ .^2)) %mean force per element
923 % normC=sqrt(C_center(1)^2+C_center(2)^2+C_center(3)
    ↪ ^2);
924 %
925 %
926 % figure(18)
927 % hold on
928 % axis equal
929 %
930 % quiver3(X,Y,Z,BX./normB,BY./normB,BZ./normB,'b')
931 % quiver3(X,Y,Z,x_diff,y_diff,z_diff,'y')
932 % quiver3(X,Y,Z,C(:,1)',C(:,2)',C(:,3)','','g')
933 % quiver3(CP_x,CP_y,CP_z,0.25*C_center(1)/normC,0.25*
    ↪ C_center(2)/normC,0.25*C_center(3)/normC,'k',
    ↪ LineWidth',2)
934 %
935 % % Draw coils
936 % line(Gamma1(:,1)',Gamma1(:,2)',Gamma1(:,3)','','Color
    ↪ ','red','LineWidth',2)
937 % line(Gamma2(:,1)',Gamma2(:,2)',Gamma2(:,3)','','Color
    ↪ ','red','LineWidth',2)
938 % line(Gamma3(:,1)',Gamma3(:,2)',Gamma3(:,3)','','Color
    ↪ ','red','LineWidth',2)
939 % line(Gamma4(:,1)',Gamma4(:,2)',Gamma4(:,3)','','Color
    ↪ ','red','LineWidth',2)
940 % line(Gamma5(:,1)',Gamma5(:,2)',Gamma5(:,3)','','Color
    ↪ ','red','LineWidth',2)
941 % line(Gamma6(:,1)',Gamma6(:,2)',Gamma6(:,3)','','Color
    ↪ ','red','LineWidth',2)
942 % line(Gamma7(:,1)',Gamma7(:,2)',Gamma7(:,3)','','Color
    ↪ ','red','LineWidth',2)
943 % line(Gamma8(:,1)',Gamma8(:,2)',Gamma8(:,3)','','Color
    ↪ ','red','LineWidth',2)
944 % line(Gamma9(:,1)',Gamma9(:,2)',Gamma9(:,3)','','Color
    ↪ ','red','LineWidth',2)
945 % line(Gamma10(:,1)',Gamma10(:,2)',Gamma10(:,3)','','
    ↪ Color','red','LineWidth',2)
946 % view(3), axis equal, axis tight

```