COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Master in Industrial Engineering

Master's final project

ANALYSIS OF NOVEL TECHNOLOGIES OF SUPERCONDUCTING MAGNETS FOR COMPACT CYCLOTRONS

Author

Carlos Hernando López de Toledo

Supervised by

Javier Munilla López

Madrid

July 2020

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

"Analysis of Novel Technologies of Superconducting Magnets for Compact Cyclotrons"

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2019/20 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Carlos Hernando López de Toledo Fecha: 14/ 07/ 2020

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Javier Munilla López Fecha: 14/ 07/ 2020

Master in Industrial Engineering

Master's final project

ANALYSIS OF NOVEL TECHNOLOGIES OF SUPERCONDUCTING MAGNETS FOR COMPACT CYCLOTRONS

Author

Carlos Hernando López de Toledo

Supervised by

Javier Munilla López

Madrid

July 2020

Analysis of Novel Technologies of Superconducting Magnets for Compact Cyclotrons

Autor: Carlos Hernando López de Toledo

Director: Javier Munilla López

Entidades colaboradoras: CYCLOMED TECHNOLOGIES SL.

Resumen

En Cyclomed, spin-off del CIEMAT, se ha desarrollado un ciclotrón superconductor para la producción de radioisótopos. Pensando en una segunda generación se hará un análisis de las alternativas existentes en el diseño de un nuevo ciclotrón clásico superconductor. El proyecto incluye, creación de una base de datos de distintos materiales superconductores, desarrollo de un programa para el análisis y comparación, y finalmente un análisis específico de la solución óptima encontrada en el anĺisis.

Palabras Clave: Ciclotrón, Superconductor, Radioisótopos

Introducción

Los avances tecnológicos de este último siglo han tenido grandes implicaciones en la sociedad. Las mejoras en el sector médico han conseguido aumentar la esperanza de vida en todos los países del mundo, trasladando las principales causas de muerte de enfermedades infecciosas a enfermedades crónicas y degenerativas, como el cáncer, el alzheimer o enfermedades cardiovasculares [1].

El uso de radioisótopos se encuentra actualmente muy extendido dentro del sector médico, en especial en la medicina nuclear. Este sector se muestra como uno de los sectores clave para afrontar los problemas relacionados con el envejecimiento de la población. Los radiofármacos tienen dos principales aplicaciones: Radioterapia y Radiodiagnóstico. En la primera se aplican altas dosis de radiación para destruir células cancerosas. En la segunda aplicación, el radiofármaco puede ser utilizado para el estudio de procesos metabólicos en conjunto con equipos de captación de imágenes [2].

La mayor parte de la producción de radioisótopos se concentra actualmente en reactores nucleares. Debido a los problemas asociados, la cantidad de reactores operativos mundialmente está decreciendo, por lo que se necesitan formas de producción alternativas [3].

Con este objetivo se ha desarrollado en CIEMAT, y ahora en CYCLOMED, un acelerador de partículas tipo ciclotrón para la producción de radioisótopos [4]. La novedad de este desarrollo es el uso de tecnología superconductora, que permite ciclotrones compactos de bajo peso y tamaño, preparados para la instalación en hospitales y laboratorios.

Definición del proyecto

El desarrollo del actual ciclotrón se encuentra en sus últimas fases pero la dificultad de alcanzar la temperatura de operación (4.2 K) ha provocado ciertos retrasos.

El objetivo de este proyecto es explorar las opciones en tecnología superconductora que existen actualmente en el mercado y podrían tener aplicación en futuros desarrollos, con especial atención a aquellos materiales que permitan trabajar a temperaturas más altas para relajar las condiciones del sistema.

Descripción del modelo

Para facilitar el análisis se ha llevado a cabo el desarrollo de una aplicación software. Esta aplicación permite obtener la geometría y características básicas del circuito magnético de un ciclotrón compacto superconductor dadas unas condiciones iniciales. Estas condiciones iniciales incluyen la elección del material superconductor, el yugo o demás características.

En el estudio de los materiales superconductores se ha creado una base de datos con aquellos disponibles comercialmente y sus características. También se ha estudiado su proceso de fabricación y las técnicas actuales en la fabricación de imanes.

La aplicación software utiliza dos modos de resolución, el primero de forma analítica, y el segundo, para una solución más precisa, hace uso de un software de simulación por elementos finitos embebido dentro de la aplicación. El programa desarrollado ha sido validado con un segundo software FEM.

Para el análisis e identificación del ciclotrón con características óptimas se ha desarrollado un método de evaluación (en forma de puntuación de 1-10) que permite discernir entre distintas soluciones, teniendo en cuenta las características finales del ciclotrón (precio y peso) y las no tangibles, como la facilidad técnica en la fabricación de imanes. Dentro de la aplicación se permiten varios tipos de anÍisis para dar flexibilidad y rapidez.

Resultados

La versatibilidad de esta solución es que permite obtener comparaciones rápidas y con buena precisión, permitiendo cambiar las condiciones iniciales del problema de manera sencilla. A continuación, se muestran las alternativas actuales que tiene CYCLOMED si quiere aumentar su temperatura de operación de 4.2 K a 10 K, para poder relajar las pérdidas del sistema.

Material	Puntuación	Peso (kg.)	Diametro (m.)	Precio (€)
Nb_3Sn	7.87	1,193	0.77	32,863
MgB_2	4.49	1,342	0.83	79,590
HTS	5.56	694	0.58	65,703

Tabla 1. Características de tres ciclotrones con distintos materiales superconductores trabajando a 10 K

Actualmente la principal alternativa en el desarrollo de un nuevo ciclotrón es el Nb_3Sn. El MgB_2 ofrece una solución con peores características a un precio mayor, mientras que los HTS pueden ser una alternativa interesante, pero la tecnología de fabricación tanto del propio material como de un posible imán siguen estando muy inmaduras.

Se han explorado también las ventajas que tendría el uso de distintos aceros en el yugo de la máquina, llegando a una reducción del 15 % en peso con el acero con mejores características magnéticas.

Conclusión

Los objetivos propuestos para el desarrollo del proyecto han sido conseguidos en su totalidad, se ha elaborado una base de datos de los materiales superconductores disponibles comercialmente, y elaborado un analisis para desarrollos futuros. Además se deja a disposición de CYCLOMED una útil herramienta que puede ser utilizada para nuevos analisis futuros según avance la técnica y se encuentren disponibles nuevos materiales superconductores.

Referencias

[1] World Health Organization. URL: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[2] Vasken Dilsizian. "SPECT and PET Techniques". En: Atlas of Nuclear Cardiology. London: Current Medicine Group, 2003, págs. 19-46. ISBN: 978-1-4615-6496-6. DOI: 10.1007/978-1-4615-6496-6_2. URL: https://doi.org/10.1007/978-1-4615-6496-6_2.

[3] IAEA. Cyclotron produced radionuclides: principles and practice. Inf. téc. International atomic energy agency, 2008.

[4] Javier Munilla Lopez. "Development of a Novel Concept of Efficient Superconducting Magnet for Radioisotopes Production Cyclotron". Tesis doct. Universidad Pontificia de Comillas, 2019.

Analysis of Novel Technologies of Superconducting Magnets for Compact Cyclotrons

Author: Carlos Hernando López de Toledo

Supervisor: Javier Munilla López

Associates: CYCLOMED TECHNOLOGIES SL.

Abstract

A novel superconducting cyclotron for radioisotope production has been developed in Cyclomed, a spin-off from CIEMAT. Regarding the possibility of developing a new generation of compact cyclotrons an analysis of the current alternatives will be conducted. This project includes the creation of a database with different superconducting material and its properties, the development of a program for the analysis and comparison of the alternatives, and the specific analysis of the most feasible solution in the short-term.

Keywords: Ciclotron, Superconductor, Radioisotopes

Introduction

The technological advances of the last century have had great implications for society. Improvements in the medical sector have been successful in increasing life expectancy in all the world, transferring the main causes of death from infectious diseases to chronic and degenerative diseases, such as cancer, Alzheimer's or cardiovascular diseases [1].

The use of radioisotopes is extended to all the medical sector, especially in nuclear medicine. It is one of the key sectors to face the problems related to the aging of the population. Radiopharmaceuticals, which includes radioisotopes in their formulation, have two main applications: Radiotherapy and Radiodiagnosis. In the first, high doses of radiation are applied in order to destroy cancer cells. In the second application, the radiopharmaceutical can be used to study metabolic processes in conjunction with imaging equipment [2].

Most of the production in radioisotopes is currently concentrated in nuclear reactors. Due to associated problems, the number of operating reactors worldwide is decreasing, so alternative forms of production are needed [3].

To this end, a cyclotron-type particle accelerator for the production of radioisotopes has been developed at CIEMAT [4], and now at CYCLOMED. The novelty of this development is the use of superconducting technology, which allows compact cyclotrons of low weight and size, prepared for installation in hospitals and laboratories.

Definition of the project

The development of the current cyclotron is in its last stages, but the difficulty of reaching the operating temperature (4.2 K) has caused some delays.

The objective of this project is to explore the options in superconducting technology that currently exist in the market and could have application in future developments, with special attention to those materials that allow working at higher temperatures to relax system conditions.

Description of the model

To facilitate the analysis, the development of a software application has been carried out. This application allows obtaining the geometry and basic characteristics of the magnetic circuit of a superconducting compact cyclotron given initial conditions. These initial conditions include the choice of the superconducting material, the yoke, or other characteristics.

In the study of superconducting materials, a database has been created with those commercially available, containing their characteristics. The wire manufacture and current techniques in the manufacture of magnets have also been studied.

The software application uses two resolution methods, the first uses an analytical approach, and the second for a more precise solution makes use of finite element simulation software embedded within the application. The developed program has been validated with a second FEM software.

For the analysis and identification of the cyclotron with optimal characteristics, an evaluation method has been developed that allows distinguishing between different solutions, taking into account the final characteristics of the cyclotron (price and weight) and the non-tangible ones, such as the technical ease in the manufacture of magnets. Various types of analysis are allowed within the application to give flexibility and speed.

Results

The versatility of this solution is that it allows obtaining comparisons quickly and with good precision, allowing the initial conditions of the problem to be changed easily. Below are the current alternatives that CYCLOMED has if it wants to increase its operating temperature from 4.2 K to 10 K, in order to relax system losses.

Material	Score	Weight (kg.)	Diameter (m.)	Price (€)
Nb_3Sn	7.87	1,193	0.77	32,863
MgB_2	4.49	1,342	0.83	79,590
HTS	5.56	694	0.58	65,703

Table 1. Characteristics of three cyclotrons with different superconducting materials working at 10 K

Currently, the main alternative in the development of a new cyclotron is the Nb_3Sn. MgB_2 offers a solution with worse characteristics at a higher price, while HTS can be an interesting alternative, but the manufacturing techniques are still in a very immature phase.

The advantages that the use of different steels in the yoke of the machine have been explored, reaching a reduction of 15 % in weight with the steel with better magnetic characteristics.

Conclusion

The proposed objectives for the development of the project have been fully achieved, a database of commercially available superconducting materials has been prepared, and an analysis for future developments has been elaborated. In addition, a useful tool is made available

to CYCLOMED that can be used for future analysis as the technique advances and new superconducting materials are available.

References

[1] World Health Organization. URL: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[2] Vasken Dilsizian. “SPECT and PET Techniques”. In: Atlas of Nuclear Cardiology. London: Current Medicine Group, 2003, pp. 19–46. ISBN: 978-1-4615-6496-6. DOI: 10.1007/978-1-4615-6496-6_2. URL: https://doi.org/10.1007/978-1-4615-6496-6_2.

[3] IAEA. Cyclotron produced radionuclides: principles and practice. Tech. rep. International atomic energy agency, 2008.

[4] Javier Munilla Lopez. “Development of a Novel Concept of Efficient Superconducting Magnet for Radioisotopes Production Cyclotron”. PhD thesis. Universidad Pontificia de Comillas, 2019.

To those who inspired it

and those who will read it

Acknowledgments

I would like to express my most sincere gratitude to Javier Munilla for providing valuable guidance and regular suggestion as a supervisor throughout this study. His experience and dedication have proved essential during the last months.

I would also like to express my deepest appreciation to Luis García Tabares for his commitment and patience during all this process, specially at the beginning, and transmitting me, even if is a small piece, part of his knowledge.

Of course, I would also like to thank the entire CYCLOMED team, especially Carlos Langeber, for its pleasant reception within CYCLOMED and his contagious spirit of entrepreneurship.

I would like to acknowledge my gratitude to my family, for their unconditional support and love, without which I would not be here. To my friends, for those of us who have been around for a few years, and those that I have known in recent years, thank you for the support and good moments.

Last but not least, thank you to the reader, I hope he finds this thesis informative and find what he was looking for. Please, feel free to contact me.

Table of Contents

List of Figures

List of Tables

Acronyms

AC	Alternating current
AISI	American Iron and Steel Institute
AMIT	Advanced Molecular Imaging Technologies
BCS	Bardeen, Cooper and Schrieffer
BSSCO	Bismuth strontium calcium copper oxide
BR	Bronce route
CAD	Coronary artery disease
CDTI	Centro de Desarrollo Tecnológico Industrial
CERN	Conseil Européen pour la Recherche Nucléaire
CIEMAT	Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas
COP	Coefficient of performance
CSS	Cryogenic supply system
DC	Direct current
FCC	Future circular collider
FE	Finite element
FEM	Finite element modeling
HTS	High temperature superconductors
ICAI	Insitituto Católico de Artes e Industrias
IT	Internal tin
LBCO	Lanthanum barium copper oxide
LHC	Large hadron collider
LTS	Low temperature superconductors
MQE	Minimal quench energy

MRI	Magnetic resonance imaging
NRM	Magnetic resonance resonance
PET	Positron emission tomography
PIG	Penning Ionization Gauge
PiT	Powder-in-tube
QF	Quickfield
ReBCO	Rare-earth barium copper oxide
RF	Radio-frequency
RI	Radioisotope
SDGs	Sustainable development goals
SPECT	Single-photon emission computed tomography
UI	User interface
YBCO	Yttrium barium copper oxide

Symbols

ϵ Strain (m/m)

μ Magnetic permeability (H/m)

μ_o Vacuum permeability (H/m)

μ_r Relative permeability

μ_{eng} Engineering permeability (H/m)

B_{c1} Lower critical magnetic field (T)

B_{c2} Upper critical magnetic field (T)

B_o Magnetic field at the center (T)

B_{oL} Magnetic field in linear iron (T)

B_{oS} Magnetic field in saturated iron (T)

B_{sc} Magnetic field at the superconductor (T)

B_c Saturation field (T)

c Iron-coil distance (mm)

FF Fill factor

g Airgap (mm)

h Cyclotron height from the poles (mm)

I_c Critical density (A)

J_c Critical current density (A/mm^2)

J_q Operating current density (A/mm^2)

l Length of the coil (mm)

p Pole height (mm)

r_o External radius (mm)

r_i Internal radius (mm)

r_p Pole radius (mm)

r_e Extraction radius (mm)

T_c Critical temperature (K)

T_k Operating temperature (K)

t Width of the coil (mm)

WP Working point

PART I

MEMORY

Chapter 1

Introduction

Population is aging: older persons are increasing in number and each year they represent a greater share of the total population [1]. This trend has been present in most developed countries during the 20th century. Japan life expectancy now tops the 84 years old, and in most European countries the life expectancy is above 80 years old.

This movement is emerging around the globe, in developing countries, the number of people aged > 65 will triple from 2000 to 2030. In other words, this tendency is not longer a first world problem, population aging is now a global phenomenon. This phenomenon has implications in all sectors of society, economic and social, but specially in the health sector.

In spite of the recent events and the unfortunate appearance of COVID-19, the world has experienced a transition in the leading causes of death, from infectious diseases to chronic and degenerative illnesses. Cardiovascular diseases, Alzheimer and other dementia's are now the leading causes of death, according to WHO [2]. Therefore, techniques and methods that allows to obtain a better diagnose and provide a better treatment for these diseases should be promoted. With this objective the AMIT project was launch in Spain in 2010.

1.1. AMIT project

In the autumn of 2010, the AMIT (Advanced Molecular Imaging Technologies) project was born within the CENIT program as one of the three projects in the area of biotechnology, healthcare and nutrition. Three objectives were defined for this project, overcome the limitations of molecular imaging technology due to the high cost, facilitate the access to radiopharmaceuticals to medical centers and improve the tools capable of diagnosis [3].

This framework was an excellent starting point to the development of new technologies in this area. In CIEMAT they started developing a compact and efficient superconducting cyclotron for the production of radioisotopes [4], with the main objective of making radiopharmaceuticals (^{18}F and ^{11}C) more accessible (Nowadays most of the production is carried out in nuclear reactors, with the logistic problems associated).

These type of radiopharmaceuticals are able, in combination with imaging devices, to give a detailed image of the area of study. Moreover, they can be customized to obtain images of specific areas, organs or cells.

1.2. CYCLOMED

The authors and stakeholders of this project considered the commercial potential of the project and it was presented to the Mind the Gap program [5] in 2019. The project was successful and a spin-off, CYCLOMED, was born. CYCLOMED and his stakeholders are now focused in two main goals, exploring the commercial possibilities of CYCLOMED's cyclotron and develop a sustainable business plan, and achieve autonomous operation of the cyclotron. The project successfully completed the design and manufacturing phase and is now in the latest stages of the testing phase, but it has encountered technical difficulties to achieve the correct working temperature.

This project has been developed for the CYCLOMED company and will examine the technical possibilities it has in the development of future cyclotrons. First, research of the superconductors materials currently available in the market will be carried with the objective of evaluating the use of alternative materials in future developments. With special attention to those who allows to rise the operating temperature, so that in this way the operating conditions of the refrigeration system are relaxed. In addition, a software tool will be developed to facilitate future analysis.

Chapter 2

State of the art

2.1. Radioisotope production

The use of radioisotopes has become a fundamental part of nuclear medicine since they allow marking molecules of medical interest for diagnostic and / or therapeutic purposes [6]. In its therapeutic use, the radiopharmaceutical emits controlled radiation to the targeted cells, minimizing the damage to healthy cells. For diagnosis, the radiopharmaceutical is administered to the patient, which, in combination with the use of imaging devices, allows to obtain an accurate image of the organ, or damaged cell that is wanted to study. Moreover, radiopharmaceuticals can be customized to locate certain types of cells.

There are two main techniques for diagnosis with radiopharmaceuticals: single-photon emission computed tomography (SPECT) or positron emission tomography (PET) [7]. It is important to note that the production of a radioisotope will depend on the implementation of its specific characteristics in a given application, that is, the radioisotopes produced will be different if the PET or SPECT technique is to be used, and in the same way they will be different if for example within the first technique it is intended to make a study of a certain metabolic process than if the study consists of monitoring blood flow. In this case the project will focus on PET diagnosis for medical imaging.

The SPECT technique uses gamma ray emitting radioisotopes. These are detected with a set of cameras from which a 3D image of the desired area is reconstructed. The most used radioisotopes for SPECT techniques are ^{99m}Tc and ^{123}I, with a half-life of more than 6 and 13 hours respectively [8].

PET diagnosis is based on the emission of a positron by an unstable nucleus. The isotope undergoes what is called beta decay, emitting a positron that will later be annihilated in the collision with an electron emitting two photons in opposite directions. The photons are detected by the corresponding equipment, where the time and place of arrival are recorded. The advantage of this technique is that it uses this greater amount of information to obtain more accurate images.

^{11}C, ^{13}N, ^{15}O or ^{18}F isotopes are mainly used in PET diagnosis. The following radiopharmaceuticals can be obtained from these radioisotopes for the following applications.

Radiopharmaceuticals	*Field*	*Application*
	Oncology	Diagnosis, evaluation and treatment of tumors
^{18}F-DG	Neurology	Diagnosis of diseases of the central nervous system
	Cardiology	Study of cardiovascular pathologies
^{18}F-DOPA	Neurology	Measure of dopamine activity
		Evaluation of low-grade gliomas
^{18}F-DG	Oncology	Early detection of recurrence
		Differentiation between tumor recurrence and radionecrosis
^{15}O-water	Cardiology	Myocardial perfusion studies
^{13}N-ammonia	Cardiology	Diagnosis of coronary artery disease (CAD)

Table 2. Radiopharmaceuticals for PET study and applications [9]

For the production of PET radioisotopes, the only possibility is the artificial production, where three options are distinguished: nuclear reactors, generators and particle accelerators [10]. The differences of each production method can be found in Table 3.

Historically the production of radioisotopes has been concentrated in nuclear reactors but due to the high costs, supply issues, and the uncertain future, the number of active reactors has diminished over the years. Cyclotrons are well positioned to substitute reactors as the main production method of certain radioisotopes for the below mentioned advantages.

	Production	Advantages	Limitations
Nuclear reactor	Neutron bombardment of targets installed in experimental nuclear reactors, causing nuclear fission reactions or neutron capture.	✓ High Production rate. ✓ Neutron-rich radioisotopes, useful for therapeutic applications. ✓ Centralized production system, limited number of reactors supply worldwide.	× Low specific activity of the radioisotope, produces long-lasting radioactive waste. × Supply issues for the closure of nuclear reactors. × Centralized production: Limited to the production of long half-life radioisotopes. × Cost overruns due to the obligatory change of targets.
Generator	Disintegrating a parent isotope with a relatively long half-life, produced by experimental reactors or accelerators.	✓ Simple logistics. ✓ Waste control from the supplier. ✓ Limited supply cost. ✓ Non-specialized staff.	× Limited to long-life parent generators produced in nuclear reactors. × The elution frequency sets the production rhythm, being subject to cycles. × High contamination, with traces of the parent radioisotope, affecting the quality of the medical image obtained.
Cyclotron	Irradiation of the target material by charged particles	✓ Radioisotopes with high specific activity, minimum waste production. ✓ Relatively low acquisition and operational cost. Simple operation and compact deign. ✓ Decentralized production.	× Low production rate. High cost targets to improve production. × Complex maintenance, need for specialized staff. × Fixed energy and limited current. Need for shielding.

Table 3. Radioisotopes productions methods comparison

2.2. History of cyclotrons

The cyclotron was patented in 1932 by Ernest O. Lawrence at the University of California at Berkeley. It is a particle accelerator in which charge particles are propelled using a high frequency alternating voltage and a constant magnetic field that guide the particles. The basic components of a cyclotron are shown in Figure 1. Three main different systems are distinguished: the ion source, which is in charge of the production of particles, the radiofrequency system, that provides an alternating electric field for the two dees, and the magnet, producing the magnetic field.

The governing equation for cyclotrons is shown in Equation 1, from which it can be derived the electric field frequency, Equation 2.

$$\frac{(mv^2)}{r}\widehat{r} = q\overrightarrow{v}x\overrightarrow{B} \tag{1}$$

$$w = \frac{q * B}{m} \tag{2}$$

The mass presented in Equation 1 and Equation 2, is subjected to the relativistic effects, so the Lorentz term needs to be accounted for the computation of the real frequency. This effect limits the energy, Equation 3, a cyclotron can reach, as the apparent mass will increase as the velocity changes. It can be derived from this equation that, once accounted for the relativistic term, the energy is proportional to the magnetic field and extraction radius.

$$K = \frac{e^2}{2m}(B * \rho)^2 \tag{3}$$

To overcome the relativistic limit, scientists developed new variations of the cyclotron. Nowadays three main families are identified: classical, synchrocyclotron and isochronous [11].

The classical cyclotron, or Lawrence cyclotrons, operates at a fixed frequency and ignores the relativistic effects and limitations. This limits the energy they are capable to provide to a maximum of 15-20 MeV. The magnetic field in this type of cyclotrons has a slowly decreasing radial gradient to provide what is called as weak focusing. This effect is used to prevent particle deviations from its ideal trajectory. It can be shown [12], that to provide simultaneous radial and axial focusing the n-index, defined in Equation 4, has to be between 0 and 1.

$$n = -\frac{R}{B_z}\frac{\partial B}{\partial r} \tag{4}$$

One possibility, as it was mentioned, to overcome the relativistic effects is the use of synchrocyclotron. The basics of this type of cyclotrons is that they match the frequency of the radiofrequency system to the increase of the relativistic term.

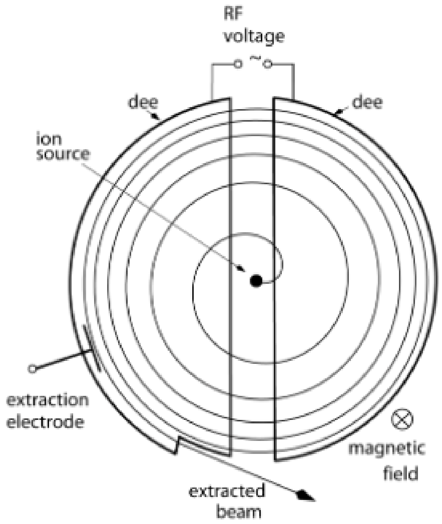


Figure 1. Basics components of a cyclotron [13]

Another possibility is the use of isochronous cyclotrons, which has an increasing magnetic field along its radius to cancel the relativistic effect, while maintaining the frequency constant as in a classical disposition. An immediate consequence of this configuration is that the magnet layout and structure must be different. This type of configuration often uses what is called as strong focusing. More about this type of focusing and configuration can be found in the literature [13] and [14].

2.3. Cyclotrons for Radioisotope Production

Cyclotrons are commonly used in medical applications. The applications include hadron/proton therapy to radioisotopes production. Cyclotrons for radioisotope production require a particle beam with sufficient energy to produce the required nuclear reactions [15]. This is why they are usually classified in terms of energy, Table 4.

Low-energy cyclotrons are focused on the production of short-lived standard radionuclides for PET studies, needing a minimum energy value of 3.7 MeV for the production of ^{15}O. They are able, with an energy of 10 MeV, to produce the most demanded nuclides, ^{11}C, ^{13}N, ^{15}O or ^{18}F. Cyclotrons with a higher energy, between 20 and 35 MeV, are considered intermediate energy cyclotrons or medium cyclotrons. They are also capable of delivering PET nuclides, but they are frequently used for producing radioisotopes for SPECT and other novel PET radionuclides, or parent nuclides for generators. Higher energy cyclotrons are scarce and usually concentrated in research institutes, they are tailored to specific needs, and they produced unique radionuclides.

Cyclotron Type	*Energy Range*	*Approximate number*	*Typical Location*
Small Medical Cyclotron	< 20 MeV	1050	Hospitals Universities Local commercial plants
Intermediate Energy Cyclotron	20 - 35 MeV	100	Research institutes Regional commercial plants
High Energy Cyclotron	> 35 MeV	50	Research institutes Cancer proton therapy center

Table 4. Distinction of cyclotron types [16]

Low energy cyclotrons are sufficient for PET production. In addition, they could benefit from on-site production due to the total size, and weight, since, the energy of a cyclotron is proportional to its size and the magnetic field, as can be derived from Equation 3 (The term r indicates the extraction radius of the machine and B the intensity of the magnetic field, and the weight is proportional to the third power of the radius).

Among the most important manufacturers of low energy cyclotrons, the following are found in order of market share (according to [17]): General Electric Healthcare (GE Healthcare), Siemens, Ion Beam Applications (IBA), Sumitomo, Advanced Cyclotron Systems Incorporated (ACSI). In Table 5, the characteristics of the commercial cyclotrons are shown as well as others

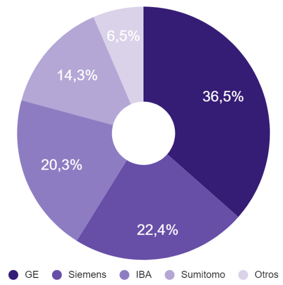


Figure 2. Market share of low energy cyclotrons

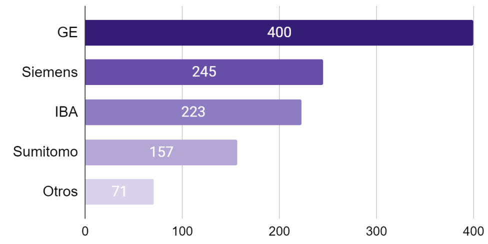


Figure 3. Number of cyclotrons sold with <18 MeV

for comparison, where can be observed how this companies commercialized more than 90% of the total low energy cyclotrons.

From Table 5, it can be appreciated that not every low energy cyclotron can be defined as compact. Certain accelerators that are in this energy range may be large enough that they cannot be installed in a hospital for on-site production of radioisotopes. In the next section the characteristics needed to be considered compact will be defined and one possibility to achieve it will be address.

Cyclotron Model	*Company*	*Energy Range (MeV)*	*Beam Current*	*Peak Field (T)*	*Weight (ton.)*	*Ref.*
MINItrace	GE	9.6	>50	1.8	5	[18]
Cyclone 3	IBA	3	60	1.9	13	[19]
Cyclone 11	IBA	11	120	1.9	13	[20]
Eclipse RD	Siemens	11	>120	1.9	11	[21]
TR19	ACSI	14-19	>300	2.1	22	[22]
BSCI 15p	Best Cyclotrons	15	100	-	14	[23]
HM-12	Sumitomo	12	-	-	-	[24]
BG-75	ABT	7.5	5	1.8	3.2	[25]

Table 5. Main manufacturers of low energy cyclotrons

2.4. Compact cyclotrons

2.4.1. Definition

In the previous section the main manufacturers for low energy cyclotrons were reviewed but the traditional machines still have some drawbacks (Size, available space or installation cost) to be installed on a larger scale, so great benefits can be obtained by improving the compactness of cyclotrons, among them we have [4]:

1. Use and production of radiopharmaceuticals could be done in the same site, reducing delivery time and waste via decayment. This will also facilitate the access to radiopharmaceuticals to small cities and poor connected areas.
2. Improve production flexibility and waste reduction, production would be linked to hospital needs reducing the total waste.
3. Better patient treatment, with on-demand non-standard radiopharmaceuticals to improve diagnosis of unique diseases.
4. Market competitiveness enhancement, ability from hospitals to produce their own radioisotopes not depending on traditional manufacturers.

From these a series of characteristic can be extracted that a cyclotron must fulfill in order to be considered as compact [9]:

1. Reduce installation footprint, for that the weight and size must be reduced by an order of magnitude.
2. Reliability and flexibility.
3. Reduced number of employees in the operation of the facility. In addition, staff must be specialized and trained.
4. Simple and user-friendly operation.
5. Low dose rates to guarantee the established dose limits, which implies the use of armored systems.

6. Low cost both in operation and maintenance. In addition, the purchase price must be affordable in the market for which it is targeted.

7. Low power consumption of the complete installation: accelerator and auxiliary systems such as refrigeration.

One possibility to fulfill these characteristics is the use of superconducting magnets. The benefits of this technology will be explained in the following sections.

2.4.2. Superconductivity

In 1911 Kammerlingh Onnes found that the resistance of mercury (Hg) reaches a value of zero ohms at temperatures below 4.19 K, to cool it down he used liquid helium which he had liquefied for the first time three years before in his laboratory in Leiden. Importantly, he noticed that the resistance drop was discontinuous, so it became clear that a phase transition to a qualitatively new state with zero resistance took place. A new state of a metal was discovered and it was named "superconductivity" [26].

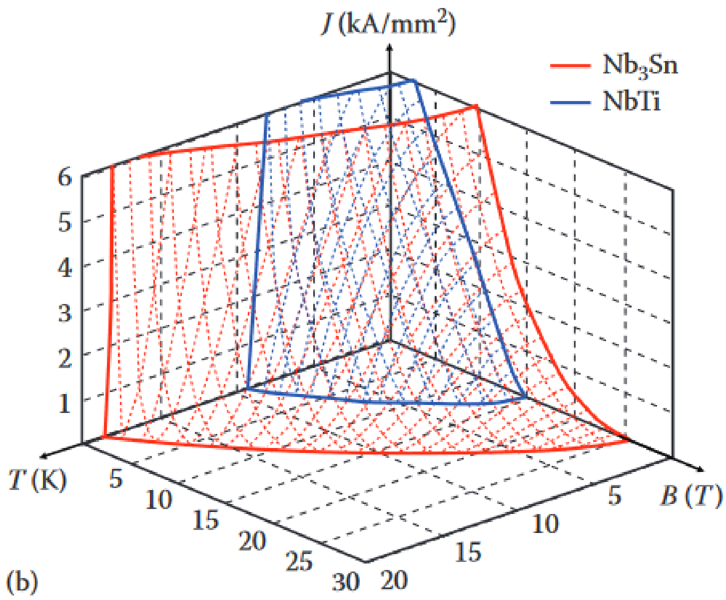


Figure 4. Critical surface of Nb_3Sn (red), NbTi (blue) versus temperature (K) and magnetic field (T) [14]

Meissner and Ochsenfeld (1933) discovered that when a superconductor is cooled below his critical temperature in the presence of a weak magnetic field H < Hc, the field is expelled from the sample (Meissner effect). Perfect diamagnetism is a fundamental property of a superconductor. Physically, this phenomenon is explained as the screening supercurrents flow in a thin surface layer of a sample, exactly canceling the external field. As a result from this effect, the magnetic field inside a superconductor is null. At a certain field, superior to the critical field, the superconducting state is lost and the material transition to a normal state with finite resistance occurs. Thus, a superconductor can be characterized by *perfect conductivity* and *perfect diamagnetism*.

The critical current, I_c, is defined as the maximum possible transport current that can flow without dissipation. The value of the critical current depends on various factors as the sample geometry or the sample quality. According to Silsbee criterion [27], a superconductor loses its zero resistance when at any point on the surface the total magnetic field strength, due to the transport current and applied magnetic field, exceeds the critical field strength Hc. This quantity Ic is called the thermodynamic critical current or the depairing current and depends on the external magnetic field.

It can be derived that the superconducting state depends on three parameters the magnetic field, the temperature, and the current density forming what is called as the *critical surface*, Figure 4. If the operating point changes, and surpass one of the three critical limits, the magnet will *quench*, and lose its superconductivity state.

The use of superconductivity has demonstrated very advantageous where high magnetic fields are required, since the null resistivity allows the magnet to reach high current densities. This phenomenon could be useful in the production of compact cyclotrons since the energy is $E \propto (r \times B)^2$, for a constant energy in order to reduce the size and weight the magnetic field must increase in the same proportion. The current technology uses copper magnets, which has a limit under 2 T due to the iron saturation, limiting the minimum weight a cyclotron can reach. The benefits of using superconducting magnets have been widely shown in the literature [4, 28], including:

- Increased in the density current provided by the superconducting coils by a factor of 500-1000, reaching higher magnetic fields. This increased magnetic field reduces the radius by a factor of 2-4, reducing the weight by a total factor of 8-12 (weight is roughly approximated to the third power of the extraction radius)

- Size reductions decreases the total footprint of the facility allowing its installation in hospitals with space restriction. Weight reduction facilitates the transportation and installation.
- The reduction in the magnet weight can lead to significant cost reductions regarding the mechanical support structure.
- Initial investment cost could be reduced, especially those related with the shielding.
- Some operational cost could be reduced, such as the electrical supply.

Although this technology also has some disadvantages [28]:

- The superconductor state is limited by the magnetic field, the temperature and the current density, the three form the *critical surface* of the material, Figure 4. If the operating point changes, and surpass one of the three critical limits, usually due to a higher temperature, the magnet will quench, and lose its superconductivity state. This could carry reliability problems and special effort need to be taken in the study of the *quench*.
- Cryogenics are needed due to low critical temperatures. Traditional LTS superconductors, NbTi and Nb_3Sn, need liquid helium to operate and HTS or MgB_2 could operate with helium or other options (Argon, Nitrogen...)
- The strong magnetic fields in the magnet affect the design of the mechanical support requiring it to counteract the effects of the Lorentz forces. Critical current can be sensitive to mechanical strain, as in the case of Nb_3Sn or MgB_2, so safety margins needs to be considered carefully.
- The high magnetic fields led to high fringe fields that could affect the health of the patient or interact with other equipment, prompting the use of passive shielding.
- The maintenance of superconductors is complicated and needs special know-how.

2.4.3. Available compact cyclotrons

Certain cyclotrons mentioned in Table 5 fulfill the definition given in the previous section. It is the case of GE cyclotron MINItrace or BG-75 from ABT. Although, neither of these cyclotrons takes advantage of the use of superconductivity.

High energy superconducting cyclotrons are relatively common, and a wide selection can be found [29]. On the other hand, the number of low energy superconducting cyclotrons are still limited even though the mentioned benefits from the on-site production. The cyclotrons currently available in the market or in their latest stages of production are shown in Table 6.

The approach each company has taken is radically different from each other. Each one opting from a different type of cyclotron, denoting the novelty of the technology. The ION-12SC, has successfully produced ^{13}N-ammonia for cardiology, the iMiTRACE from iMiGiNE is also commercially available, while the other two are in the latest stages of its development, the CYCLOMED cyclotron will be addressed more in detail in 2.5.

Cyclotron Model	*Company*	*Cyclotron Type*	*Energy Range (MeV)*	*Superconductor*	*Weight (kg)*	*Ref.*
AMIT	CYCLOMED	Classical	8.5	NbTi	1500	[4]
ION-12SC	Ionetix Corporation	Isochronous	12.5	Nb_3Sn	2300	[30]
-	VECC	Isochronous	25	NbTi	2000	[31]
iMiTRACE	iMiGiNE	Isochronous	12	NbTi	4500	[32]

Table 6. Low energy superconducting cyclotrons

Comparing the solutions from Table 5 and Table 6 one can make an idea of the benefits of using superconducting technology in cyclotrons. Moreover, in Table 7, a comparison elaborated by Ionetix [33] is shown. The weight is reduced by a factor 5-10, but if we take into account the shielding the differences become more significant. The price is also reduced, mainly because of the lack of shielding, the cheaper installation and the unneeded site adequation due to its smaller weight and size.

	Ionetix ION-12SC	*Siemens Eclipse*	*GE PETtrace*
Room Requirements	3.65×3.5 m^2	6.7×8 m^2	4.6×5.5 m^2
Weight	2.3	11	22
Shielding	Minimal	39 tons	47 tons
Price ($)	~1.5 MM	2.5 MM + 7.5 MM of infrastructure cost	2.5 MM + 7.5 MM of infrastructure cost
Staff Required	1	2-5	2-5
Power Requirement	34 kW	35 kW	70 kW

Table 7. Comparison of existing low energy cyclotrons [33]

2.5. AMIT Cyclotron

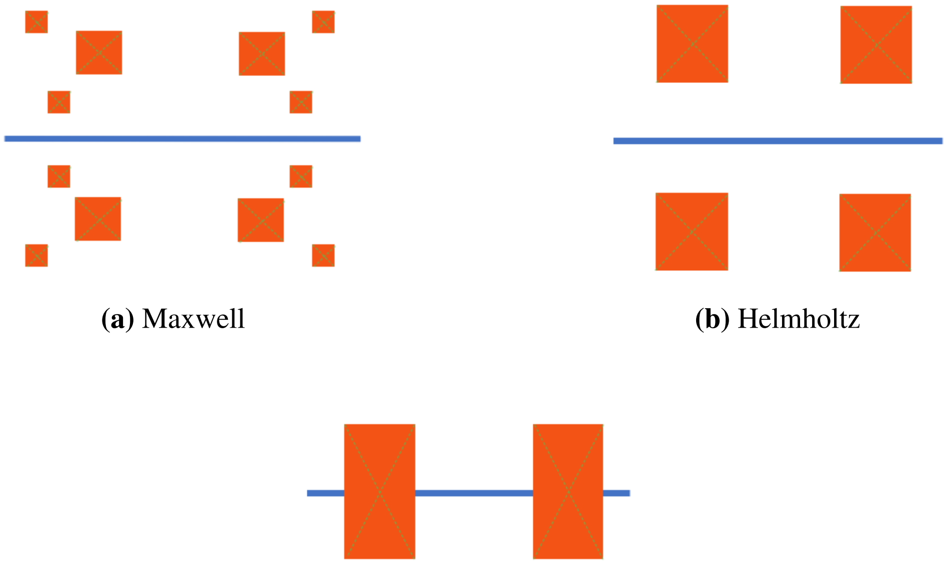
The AMIT cyclotron is a novel development of a compact cyclotron for radioisotope production. In the following sections the design of the subsystems will be reviewed, with special attention to the electromagnetic design. The following sections are based on the findings of the current design author [4].

2.5.1. Electromagnetic design

The designs of the different subsystems are not independent of each other. For the electromagnetic design an iterative process has to be carried for the beam dynamics and magnet specifications, Figure 6.

The initial requirements come from the strategic part of the project. The aim was to produce a compact cyclotron, that fulfills the definition of section 2.4. To reach that objective it was decided to use a classical lay-out, which for PET purposes with low energy requirements was find to be sufficient. The energy was set to 8.5 MeV.

A Helmholtz coil disposition was chosen for its simplicity and mean plane access. Alternatively the other options were a solenoid design, with a better field quality but no access to the midplane, or Maxwell coils, which can enhance the field uniformity but at greater cost and increased complexity.



(a) Maxwell

(b) Helmholtz

(c) Solenoid

Figure 5. Possible magnet configuration for a uniform field required for a cyclotron

The magnetic circuit will use a warm iron configuration and a flow refrigeration cryogen supply. It will be explained in more detailed in the next section. The electromagnetic implications are that the refrigeration system and the magnetic circuit can be design independently since there is no need to free space for a cryocooler.

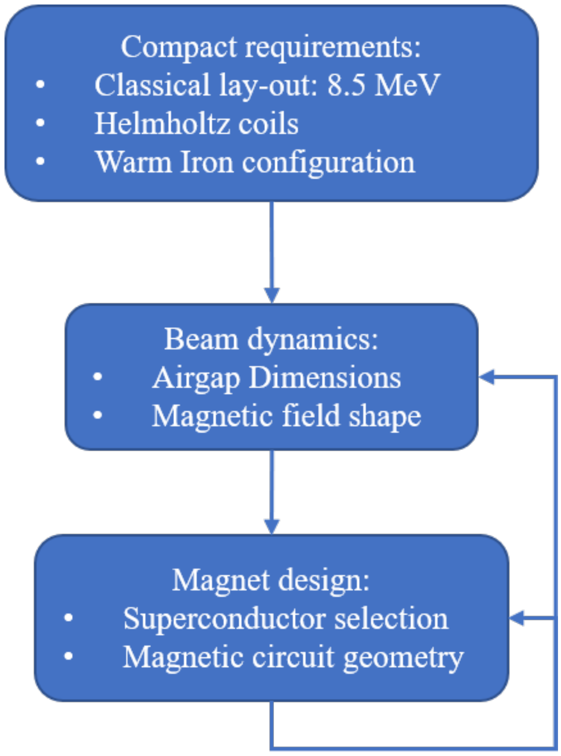


Figure 6. Iterative process for the electromagnetic design of a cyclotron

The first step of the iteration process is the beam dynamics analysis. The electromagnetic input data needed for the process is the energy required and the magnetic field at the center. The energy was fixed at 8.5 MeV and the magnetic field is an iterative variable, which reach a value of 4 T at the optimum point Figure 7. The output data from the beam dynamics is shown in Table 8, that serve for the magnet design.

Parameter	*Value*
Magnetic field	4 T
n index	1.5%
Airgap height	>74 mm
Extraction radius	108 mm

Table 8. Main magnet specifications [4]

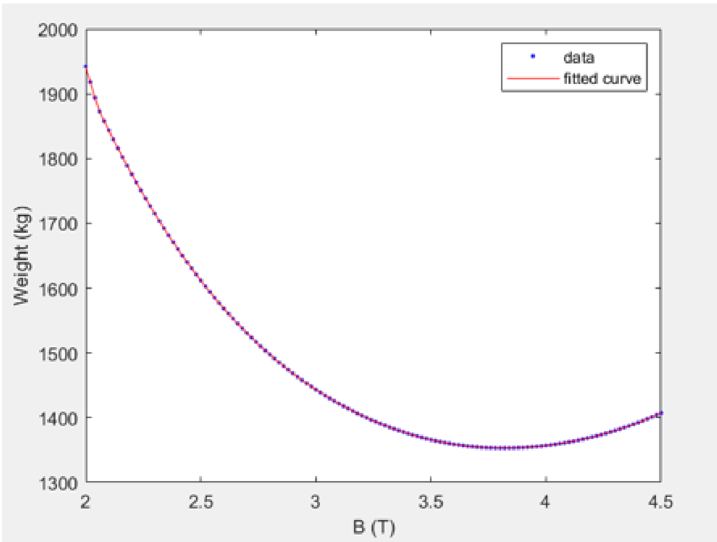


Figure 7. Weight vs Field for a NbTi based cyclotron, the minimum weight is reached at 4T

For the electromagnetic design simple analytical models cannot be used due to the non-linearity of the problem. Instead, finite element software is needed. For a first approach a 2D FE electromagnetic software (Quickfield [34]) was used, where non-linear properties of ferromagnetic materials can be defined by the user. The model was refined using a 3D FE software, Ansys Maxwell [35]. The magnet design is an iterative process itself, the following steps were followed:

1. Select a superconductor wire, with appropriate characteristics, performance (critical surface) and Copper/NonCopper ratio, and a safe load margin.

2. Define the geometry of the magnetic circuit, coil and iron yoke. Evaluate the density current needed and number of turns.

3. Results analysis: Evaluate the peak magnetic field at the center and in the coils, the real load margin, the field homogeneity and the characteristics of the magnet.

4. Perform new iteration until an optimum value is obtained.

The superconductive materials that were evaluated at the beginning of the project were NbTi, Nb_3Sn, MgB_2 and Rare-earth barium copper oxide (ReBCO) superconductors, a type of high temperature superconductors (HTS). The sections in which these materials were evaluated are shown in Figure 8, with the corresponding punctuation in each section. NbTi was selected since it has reached a certain degree of technology maturity that the others have not yet achieved. The commercial availability, price, length and coil manufacturing were the main decision factors.

Once the superconducting material is selected, certain parameters of the wire have to be defined. For example, the copper to the non-copper ratio, that affects the current density and the *quench* protection, or the cross-section area that would change the nominal current and length of the wire. The wires commercially available are limited, and the design must be done adapting and using the commercial wire specifications, and not the other way round.

Figure 8. Superconductor materials comparison for AMIT cyclotron

A starting point would be selecting a nominal current. A low nominal current could simplify the election of the commercial wire. Furthermore, the thermal losses in the current leads are proportional to the current. As a starting guess, the current interval would be between 100-300 A.

The copper to non-copper ratio is defined as the amount of copper divided by the amount of non-copper material, in this case NbTi. The quantity of copper affects the magnet protection and the current density. The advantages of a high quantity are:

- Thermal stability: Copper can help to evacuate heat during current transients (ramp-up for example) decreasing the peak temperature of the coil.
- Electrical stability: If one coil quenches the current flowing through the coils will decrease due to the increased resistivity and a voltage proportional to the product of the inductance will rise in the non-quenched coil. A high copper percentage can keep the electrical resistivity in a reasonable value and in case of quench the current decay will be slower.
- Quench protection: During quench the superconductive filaments become resistive and due to Joule losses the coil temperature and voltage could increase to dangerous levels. Increasing the amount of copper gives current a less resistive path in case of quench, since the resistivity of copper is lower than the resistivity of NbTi.

Although a higher ratio has certain disadvantages:

- Lower current density: for certain number of ampere-turns the coil size will increase proportionately to the quantity of copper.
- Higher weight and price of the coil as more material needs to be extruded. The coil size will impact the rest of the magnetic circuit, a bigger iron yoke will be needed.

To select the appropriate copper quantity an intense quench study must be done to evaluate all risks. As a starting point, from internal know-how and experience, the copper to non-copper ratio was set to an interval between 3 and 5.

In an electromagnetic design a certain safe margin must be defined to, in case of error during design or manufacturing, have an operating margin. For superconductivity applications, it is common to use the load margin, defined as the proportion of the actual current in the wire and the maximum current that the wire can reach. This method is shown in Figure 9, where the operating line is defined by evaluating the wire for several current values. The maximum operating point is defined as the intersection of the operating line and the critical surface of the material. If the operating point encounters to the right of the intersection the magnet will quench and to the left the magnet will operate in safe conditions. The margin, which corresponds to the proportion of the length from the operating point to the intersection, to the total length of the line, is the denominated the load line margin.

A high load margin would reduce the risks of quench, although the efficiency of the system would be reduced. A typical value of the working point in the load margin is 60-70 %.

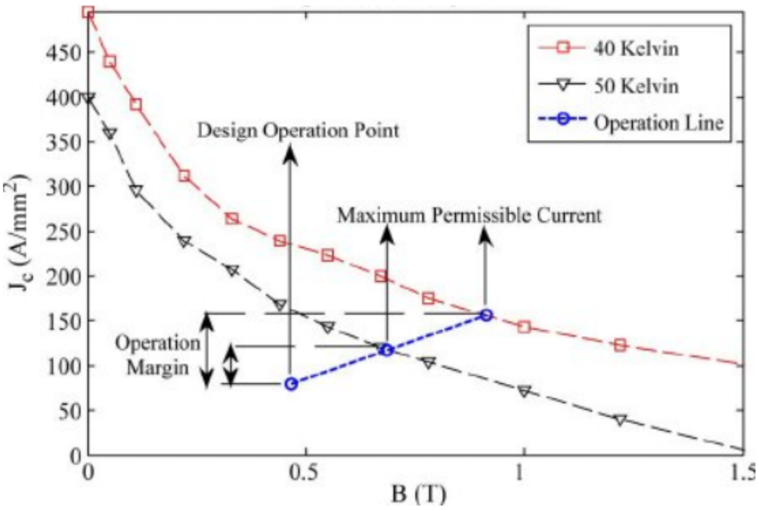


Figure 9. Load line of a magnet for a given design, safety margin for two possible operative temperatures [36]

The magnetic circuit consist on two coils, in Helmholtz disposition and an iron yoke. The objective is to produce a magnetic field with the characteristic provide by the beam dynamics analysis in the poles, the two iron mounds at the center of the iron yoke.

The distance between the coils and the iron is a boundary condition for the magnetic design that is try to be minimized as much as is technologically feasible. In this space there will be accommodated, the casing as supporting structure, the thermal shield for insulation of the cold mass and the cryostat wall as vacuum vessel. For the design this dimension is critical since a higher distance means higher distance to the magnetic axis and the efficiency is reduced. Furthermore, the coil cross section will increase, increasing the distance even more.

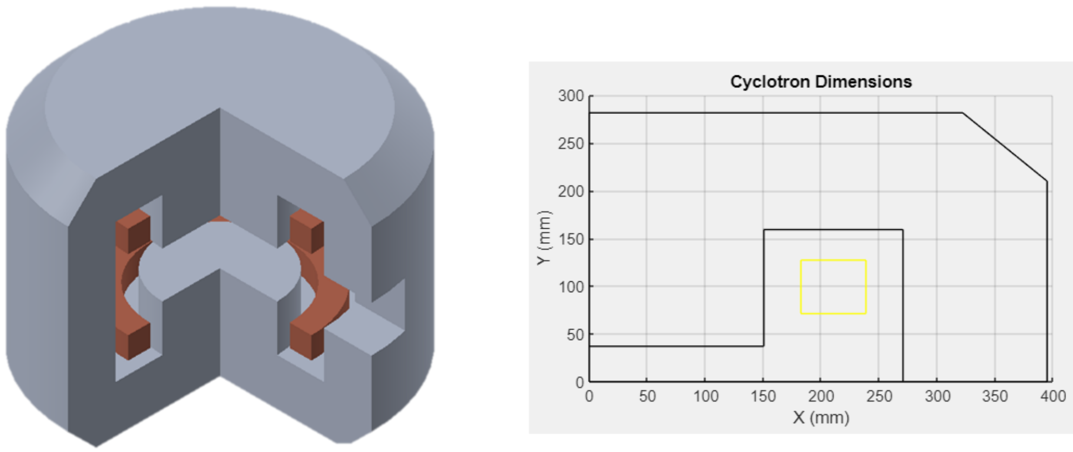


Figure 10. Magnetic circuit model of cyclotron (3D and pseudo-2D)

The problem is almost axially symmetric, but the cavity for the radio-frequency system spoils it. To simplify and use the 2D iron model, the iron yoke is split in two. The first one conserved the properties of the iron, while the second one has a permeability computed as the weighted average of iron and air permeability.

The last step is the magnetic field tuning and refining. This step is out of the scope of this project, but the idea is to reach a pole geometry that maximizes the similarity between the desired and actual magnetic field. For that a cost function is defined and through a series of iterations a final geometry is reached.

The magnets parameters are shown in the following table:

Parameter	*Value*
Wire Diameter(bare/insulated)	0.85/0.90 mm
Coil section (width/height)	54.65/54.44 mm
Operating point	70 %
Nominal current	108.6 A
Number of turns	4235
Self-Inductance (nominal current, both coils)	38.35 H
Cu/nCu ratio	4.5:1
Distance from coil to iron	32 mm

Table 9. Main magnet parameters from 2D model [4]

2.5.2. Subsystems

The rest of the subsystems that conform the cyclotron are:

- Accelerating system
- Ion source
- Target and particle extraction
- Control system
- Cooling system

The accelerating system is based on a single 180 ° dee, with an electric field up to 60 kV. The RF resonator is designed to be out of the yoke for compactness. The system uses an internal PIG ion source with a cold cathode. The ions are extracted by a carbon foil, that strips off the electrons of the H- particles. More information about the mentioned subsystems can be found in [4, 37, 38, 39].

To reach the desired temperature, it is impossible to avoid the use of cryogenics. The cooling configuration used is denominated warm iron, where only the magnet coils are cooled down and the iron-yoke is at ambient temperature. This configuration allows for a rapid set-up, the cold mass is small compared to the total mass, and it can achieve quickly the nominal temperature. On top of that the cooling power to cool down the mass is significantly lower, reducing the operating costs of the cooling system.

For the cryogen supply it was opted for a flow refrigeration system, where certain amount of cryogen is pumped into the system to cool it down. It allows for an autonomous and cheap operation while the main components can be kept away from the cyclotron, reducing magnetic interference and allowing scalable refrigeration power. The system itself is quite a novelty and an autonomous liquefactor for liquid helium production was developed in collaboration with CERN, it was named Cryogenic Supply System (CSS). The system has an available cooling power of 1.5 W at 4.2 K, this number is extremely low, so the expected thermal losses must be computed with extreme precaution.

Figure 11. CYCLOMED cyclotron prototype

Now, after some technical challenges to reduce the thermal losses, the project is in the latest stages of tuning up the CSS to achieve the correct operating temperature and autonomous operation.

2.6. Recent Developments

During the last 5 years, new approaches for compact cyclotrons have been proposed. The improvements have centered in the following aspects:

- Superconducting materials: The performance limits from NbTi, especially the low working temperature, are an incentive to change the coil material for a more comfortable operation. ION-12SC cyclotron, from Ionetix [33], have Nb_3Sn coils, but still operating at 4.2 K. Certain authors [40] have proposed to used an HTS superconductor for a compact variable-energy cyclotron for RI production, to take advantage of the high current density and high thermal stability. Alternatively MgB_2 magnets were proposed for MRI systems [41]. MRI systems have certain design similarities with cyclotrons. The main advantage is that MgB_2 have a higher critical temperature than NbTi.
- Yoke materials: In [42] the authors proposed a superconducting cyclotron with holmium poles. Holmium has a higher saturation field than iron allowing to increase the field and therefore reduce the weight and size of the cyclotron.

- Yoke configuration: Some authors, including [40], have proposed an iron-less (or air-core) cyclotron. This configuration reduces drastically the weight and size while maintaining a highly precise magnetic field. On the other hand, bigger and more expensive coils will be needed to reach the field specifications.

Chapter 3

Project Definition

3.1. Motivation

Compact superconducting cyclotrons are a promising technology and could have a positive impact in the medical sector the next years. The advantages superconductivity provides, in terms of weight and price, can overcome the historical difficulties these devices had in order to be implemented in a larger scale. Although, as listed in 2.4.2, the technology still has some drawbacks, mostly related with the necessity of cryogenics. In this thesis the author will explore and analyze the existing options regarding superconducting technology.

NbTi has been the most widely used superconductor, because its high ductility, which makes it an ideal candidate for winding. Although the superconductive properties of this material are limited, the critical field and temperature are extremely low compared to others (Figure 13 and Figure 12).

Among the conventional LTS superconductors, those who adequate to BCS theory (Green circles in Figure 13), Nb_3Sn and MgB_2 are the most promising. Nb_3Sn have a great capacity to transport high current densities even at high fields compared to NbTi, and the manufacturing and technical challenges it has historically presented have been improved over the years. MgB_2, is a recent discovery, with high critical temperature, similar current density properties to NbTi, and ease to manufacture that makes him an interesting candidate to substitute NbTi in low field applications.

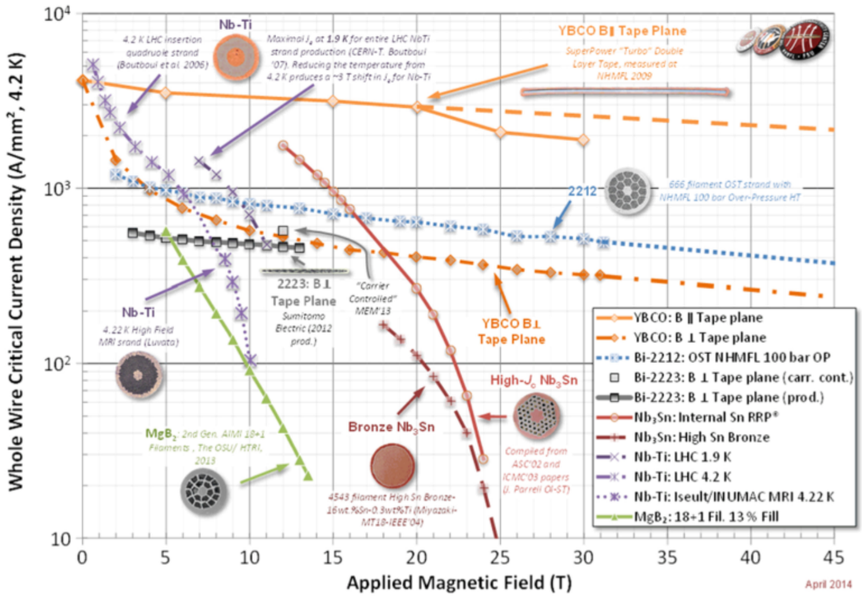


Figure 12. Critical current versus applied magnetic field for different superconducting materials. Data from the National High Magnetic Field Laboratory [43]

HTS superconductors, among which we have cuprates (Blue diamonds in Figure 13), have historically been very promising, but the prohibitive prices when compared with traditional superconductors have limited their applications, although in the past years the price has decreased and a wide selection of commercial HTS tapes are available. Other HTS, like iron-based superconductors (Yellow squares in Figure 13), are very recent and their application has been mainly limited to academic research.

3.2. Objectives

The motivation of this thesis is to examine the possibilities regarding superconducting technology that CYCLOMED has in future developments of cyclotrons. A series of primary objectives were defined, which its accomplishment will define the success of the project. Moreover, some enhanced objectives were included in addition to the primary objectives. These objectives were considered during the elaboration of the project and its achievement could benefit the project and the company. The primary objectives are the following:

- Creation of a database of the current superconducting materials available at the market. The database will include information regarding their properties and price.

- Development of a computer program that calculates the basic dimensions of a cyclotron, to provide preliminary rough values on weight and price. The program will use a 2D software to run.
- Analysis and comparison on the impact of using different superconducting materials in the cyclotron geometry.
- Detailed analysis of the most feasible solution in the short term.

Among the enhanced objectives there are:

- Detailed analysis of current available superconductors for magnet fabrication. Study of the current wire elaboration techniques and impact during the winding process.
- Elaboration of a computer program that facilitates the analysis to the user.
- Impact on the magnetic circuit of using superconductors with ferromagnetic materials in its matrix composition.
- Validation of the program with a 3D simulation software.
- Analysis and comparison of the varying the initial inputs in the cyclotron geometry, e.g., the iron-yoke material.

3.3. Resources

The resources used in this thesis will be provided by CYCLOMED and the University. The following resources will be used:

- Quickfield v.6.4 as the principal software to solve 2D magnetic problems [34]. It is a finite element analysis software package developed by the Danish company Tera Analysis. The main applications are simulations of electromagnetic fields, although it has packages for thermal and stress analysis.
- MATLAB R2019b as a calculation software and as an automation software for Quickfield problems.
- License of Solidworks 2018, as a 3D software modeler.

- Maxwell, now part of the electronics package from ANSYS [35]. It is a finite element software for electromagnetic problems. It will be used as a 3D software to validate the 2D solutions reach with Quickfield.
- Microsoft Office license.

3.4. Methodology

The project will be divided in the following phases:

1. Technological Prospective: Market analysis and state of the art of the current solutions that have been proposed in the last years related to cyclotrons for radioisotope production. During this phase we include the familiarization with the tools that will be used during the thesis and the technology covered.
2. Identification of superconductor materials: Market researches of commercially available superconductors. Contact with the manufacturers and collect information regarding the material properties, availability, and prices.
3. Creation of the analysis program: During this phase will be developed a software tool that creates an initial drawing of the cyclotron dimensions. The initial calculations will be done analytically and a 2D FEM software will be used to reach a refined solution. Finally, the solution will be compared with a 3D FEM software to validate the results.
4. Analysis and comparison: In the latest stage of the project an analysis of the most promising technologies will be carried out. The analysis will consist on a comparison of the projected weight, price, size, and working temperature of the cyclotrons using the superconducting technologies currently available at the market.
5. Memory elaboration. Finally, the final memory of the thesis will be prepared. The degree of achievement of the objectives will be evaluated. All results will be analyzed, and the final conclusions will be exposed.

Chapter 4

Potential superconducting materials for compact cyclotrons

The first observations of superconductivity were in relatively pure metallic elements. Initially, there was an over-enthusiasm for the immediate applications in energy or in electrical engineering. Unfortunately, it dissipated soon due to the sensitivity of the first tested superconductors to magnetic fields and therefore their low transport capacity of current. Most of these pure metals, that were initially tested, exhibit a *Type I* behavior, where the superconductor try to shield its interior preventing the magnetic field to penetrate (Meissner state), until superconductivity breaks down suddenly. Only three metals, among them Nb, exhibit *Type II* superconductivity. These type of superconductors tolerate higher magnetic fields since they let the flux penetrate inside in the form of flux lines (Shubnikov phase). Consequently, Nb is considered the only pure metal suitable for superconducting applications.

One common technique, alloying, has been broadly used in order to find Type II metals and/or enhances the critical magnetic field and temperature further. Moreover, some compounds present special characteristics that improve the properties of pure metals. It is the case of intermetallic compounds with a specific composition and crystal structure, for example, A15 structure, that show improved properties than common alloys. In spite of these modifications, the current these type of superconductors is capable to carry is significantly low. The Lorentz forces exerted on the flux lines cause their movement, that results in losses and heat generation. Therefore, it is common to introduce defects to the micro-structure to pin the flux lines. This type of superconductors are called hard superconductors.

For all relevant applications in industry the conductors take the form of multifilamentary wires for electromagnetic stability. It is common to find the filaments embedded in a copper matrix. From the point of view of the manufacturer their goal is to obtain high current capacity wires in long lengths to reduce the number of joints.

Historically, pure Nb, Nb-based alloy NbTi, and the Nb-based A15-type compound Nb_3Sn have been the only superconductor materials industrially produced in large quantities. Although, the number of superconductors has increased over the years (Figure 13), and the manufacturing methods have improved in order to make them commercially available. In the next sections the following components will be explored in more detail: NbTi, Nb_3Sn, MgB_2 and HTS cuprates.

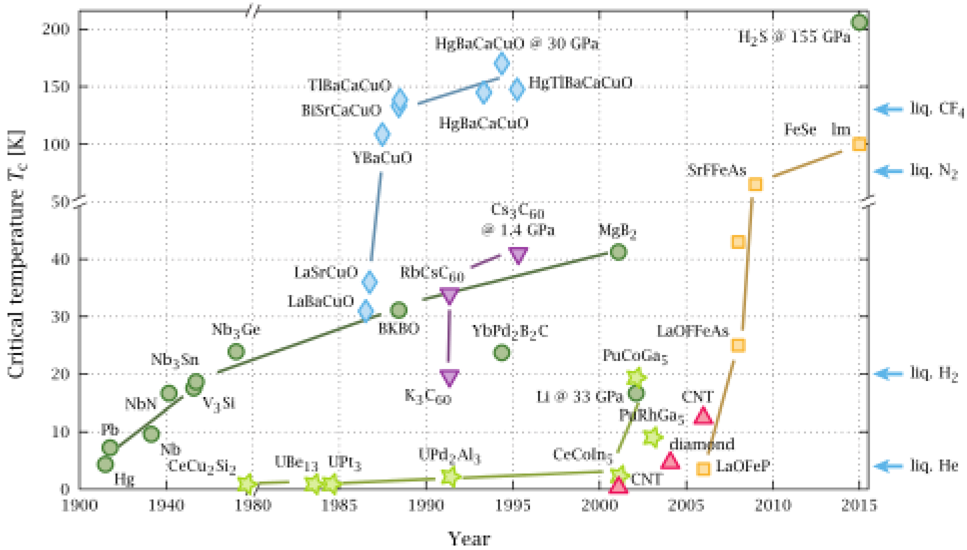


Figure 13. Date of discovery of certain superconducting materials versus its critical temperature [44]

The wire alone cannot provide the desired magnetic field for each application. Depending on the application two general magnet shapes for particle accelerators can be distinguished, solenoids and dipoles. Solenoids, Figure 14, are intended to produce an axis-parallel magnetic field, used for NMR or MRI magnets, or for cyclotrons. On the other hand, dipoles, Figure 15, produce an axis-perpendicular magnetic field, used in certain particle accelerators that need a perpendicular field to the motion of the particle.

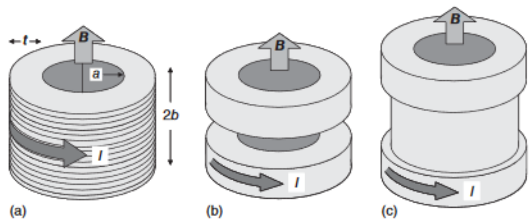


Figure 14. Solenoid windings, (a) simple solenoid, (b) Helmholtz coil, and (c) with end cheeks to improve field uniformity [45].

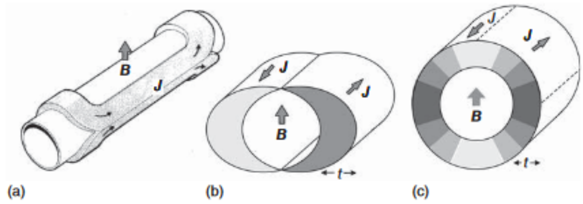


Figure 15. (a) Saddle winding to produce transverse field, (b) overlapping ellipse cross-section, or (c) current density varying as cos(θ) produce a perfect dipole field [45].

Two main winding techniques to produce solenoid magnets are identified [46]:

- Layer wound: In this technique the wire is coiled around a mandrel following a helical path. It is possible to add more layers on top of the previous one.
- Double pancake: Used when the wire is too rigid to follow a helical path. Two separated flat coils are wind parallel to each other join at the middle of the wire. An advantage is that the cable ends are located in the external part of the magnet, where the field is lower and the connections are less vulnerable to quench. For bigger coils, the usual procedure is to stack this "double pancakes".

During or after the winding, the coil is impregnated to fill the holes between wires. It reduces the movement of the wires during operation. There are two techniques:

- Wet impregnation: The resin is applied to the coil during the winding phase, once the coil is finished, if necessary, it is introduced in an oven and the resin polymerizes.
- Vacuum impregnation: Wet impregnation has certain disadvantages when the coil is large, the resin polymerizes before the process is done. This method overcomes this problem. It uses a watertight mold where the coil is placed, inject the resin from one side and from the other applied vacuum.

After the impregnation, one of the numerous possibilities to add mechanical protection, is to wrap the coil with a fiber glass layer. Also, it is possible to use, shim cylinders to add certain precompression to the coil.

In the next sections of the project the aforementioned materials will be review, first their manufacturing process [45] and winding issues will be addressed. Afterwards, a modeling section for each material will be included. In this case their superconducting characteristics, and the procedure followed to obtain the critical surface will be explained. In the last point of this section a procedure will be elaborated in order to categorize the superconducting material from best to worst for a certain operating point.

4.1. NbTi

NbTi/Cu multifilamentary wires are the most commonly used in superconducting applications. Every year hundreds of kilometers of NbTi are manufactured. NbTi has the perfect conditions to be produced in a multifilamentary superconductor wire. It has the capacity to transport large critical current densities even up to 10 T at 4.2 K, in spite of its relatively low T_c (9.6 K). It usually works together with Cu, that gives additional electromagnetic and thermal stability in case of quench.

The traditional manufacturing process of NbTi consist on four steps; Melting of a NbTi alloy under vacuum with a typical composition of 50% Ti, with special care to achieve the desire composition and homogeneity; β-quench, heating to 800 °C and fast cooling to room temperature; Warm extrusion in a copper matrix, where the NbTi is usually wrap around a Nb sheet to avoid the formation of Cu-Ti; Finally the monofilament rod is cold drawn, and then convert into hexagonal shape. The hexagonal rods are stacked into a Cu can to form a multifilamentary wire.

An alternative process is to drill holes into a Cu bar and fill those holes with NbTi rods. A third method is putting round NbTi rods into hexagonal Cu tubes with round hole and assembly in a Cu can [45].

The winding process is relatively easy for NbTi due to its high ductility, and no further actions are needed than the described in the previous section.

4.1.1. NbTi modeling

Due to the limited resources from manufacturers the available data from commercial wires is scarce, usually limited to a few sets of tests of B_{c2} vs J_c at a constant temperature of 4.2 K. Scaling laws are common to solve the short sampling issue. These scaling laws are intended to describe the critical surface formed by T_c, B_{c2}, J_c, and ϵ. Starting with measurements at, for example, 4.2 K they allow us to estimate J_c at conditions not easily accessible for experiments. Of course, this tool has its limitations as the critical surface depends on the manufacturing process and is therefore not universal. For these purpose the practical fit proposed in [47] will be used, where the fit for the critical surface is according to the author:

$$F_c \equiv J_c \times B = C_o b^{\alpha}(1-b)^{\beta}(1-t^n)^{\gamma} \tag{5}$$

Further information can be found in [47]. One can notice from this expression the non-dependability of the critical density to strain, showing the high ductility of the wire that was mentioned in the previous section.

To adapt this expression to the actual data a range for the values of the coefficients is given in [47]. Since the values of the coefficients differs across manufacturers the author adopted the following process:

1. Open the Curve Fitting Tool from MATLAB and upload the set of points provided by the manufacturer.
2. Use a custom equation as the fitting function. In this case the equation will be Equation 5.
3. Define the range of values for each coefficient. Also define the starting point for the function to iterate, initially a good starting point could be the middle value of the defined range.
4. Define fitting method and conditions. It is recommended to use the default settings defined by MATLAB
5. Check the adjusted R-squared to evaluate the goodness of the fit.

In Figure 16 the obtained fit was applied to a multifilamentary wire from Supercon. Although the correlation between the sample and the fit is high, of 0.99, a security coefficient is applied in order to compute the critical surface (Figure 17). In certain occasions the lack of tested points could create an illusion of precision on the fitted curve, obtaining a high R-squared. This phenomenon is known as overfitting. To address this problem the author insist on the use of security coefficients when the data is limited.

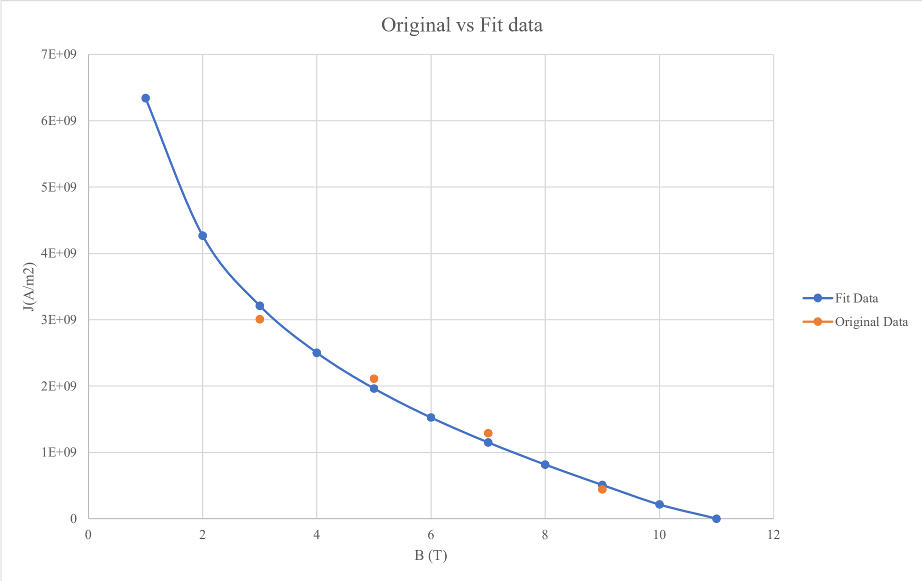


Figure 16. Field dependence of the normalized Jc(B, T) for a NbTi Supercon multifilamentary wire, measured at 4.2 K. The fit to the data is shown (solid lines) together with points provided by the manufacturer

The current density of NbTi quickly declines with temperature Figure 17, reaching its critical temperature at 9.6 K, although at temperatures higher than 6 K the current they are capable to carry is extremely reduced. For that reason NbTi is commonly operated at 4.2 K, there are some examples in which it is operated at lower temperatures. The LHC in Geneva operates at 1.9 K to increase the critical field from 10 T to 14 T [48].

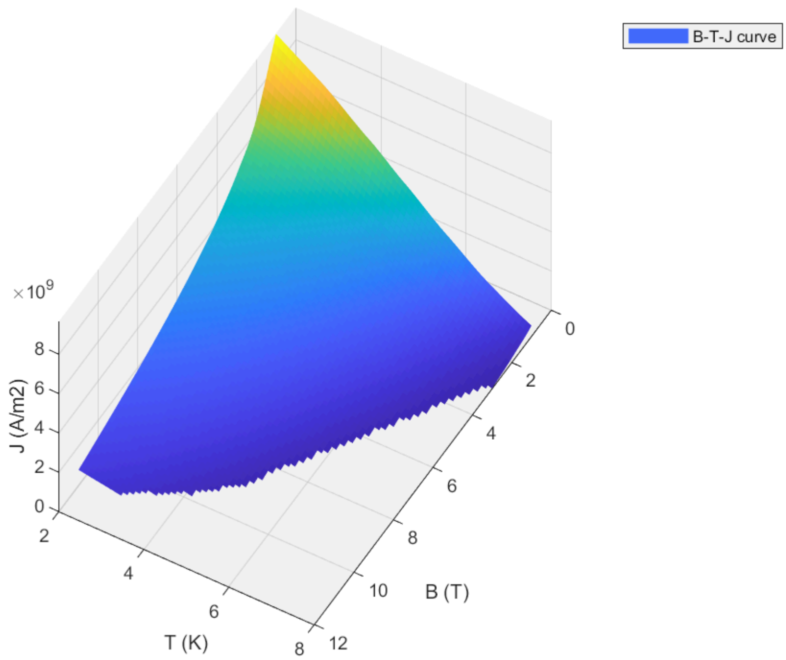


Figure 17. Critical surface from a Supercon NbTi wire

4.2. Nb_3Sn

Nb_3Sn, and most of A15 materials are very brittle and the manufacturing methods developed for NbTi cannot be used to fabricate a wire. Nb_3Sn was first commercialized in the form of tapes (Now commonly used for high temperature superconductors (HTSs)), by coating of wide thin substrates. Although this technique have some inconveniences since magnets built with these tapes exhibited severe flux jumps and quenches, especially when there is a perpendicular component of the magnetic field in the broad face of the tape. Currently, there are various methods, but solely the industrially implemented ones will be addressed: Bronce route (BR), the internal tin process (IT), and a powder-in-tube method (PiT).

Among these techniques the most popular is the bronze process [48], in which filaments of pure Nb are drawn down in a matrix of CuSn, bronze. Once the wire reach the desired size, a heat treatment is applied. During the course of the heat treatment the tin diffuses through the bronze and reacts with the niobium to produce Nb_3Sn.

The internal tin process is based on the bronze-routed process, with the objective of increasing J_c by increasing the concentration of Sn. To this day, numerous types of internal tin processes are found, and they all commonly use the three materials, Sn, Nb and Cu. One of the several wire making methods includes assembling modules, each integrating Cu, Nb core and Sn alloy, in advance. Another method involves assembling together a module combining a Cu and Nb core and another module combining Cu and Sn alloy. Similarly, to the bronze-routed process, the last step is applying a heat treatment where Nb_3Sn is formed by diffusion reaction. The resulting wire, Figure 18, has a cross-section with multiple Nb cores and Sn alloy, all embedded in a Cu matrix.

The powder-in-tube (PiT) process involve the following steps: Fill a Nb tube with Sn powder, the powder can be pure Sn, Sn alloys or Sn compounds. Then the tube is placed in a sheath of copper, assemble as a hexagonal column, forming the composite. Finally, the Sn diffuses in the Nb alloy tube forming Nb_3Sn.

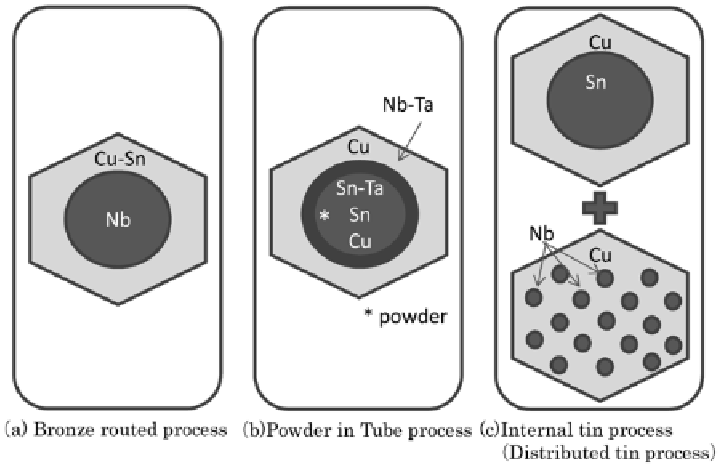


Figure 18. Schematic cross section of Nb_3Sn single cores in different processes [49]

In this method a Cu-Sn alloy is formed during the production process of the Nb_3Sn, that contributes to reduce the adverse effect that occurs during the drawing. During the diffusion process the Cu matrix is not involved, resulting in a high rate of Nb-Sn diffusion. This give the possibility to increase the volume fraction it can be obtained of Nb_3Sn resulting in a higher critical density current. However, this effect has its disadvantages when compared to the bronze-routed process, or to the internal tin process. The higher Nb_3Sn core increases hysteresis losses when used for AC applications.

The winding process is not straightforward for the Nb_3Sn due to its brittleness. At a certain curvature radius the wire could suffer irreparable damage. The most common technique used for solenoid manifacturing is the denominated "wind-and-react". In this process the wire is wound before the heat treatment, since the unreacted Nb filaments conserved their ductility. After the reaction the crystaline and brittle Nb_3Sn is formed. For certain magnets, there is also the possibility to use the "react-and-wind" technique [50].

4.2.1. Nb_3Sn modeling

The data available from manufacturers is, as from NbTi, very limited and is common to use scaling laws to extrapolate the data. There numerous of models for the critical surface of Nb_3Sn and in general A15 superconductors, that have been developed during the last 20 years [51]. For this case, we will use the fit presented in [52]:

$$F_c \equiv J_c \times B = C[b_{c2}(\epsilon)]^s(1-t^{1.5})^{\eta-1}(1-t^2)b^p(1-b)^q \tag{6}$$

The procedure followed to obtain the value of the coefficients of Equation 6 is the same as the described in subsection 4.1.1. Unsurprisingly, Nb_3Sn has a certain dependency on the strain applied to the wire. The strain dependency for a certain wire, is shown in Figure 19, it can be appreciated how, as the deformation increases, the value of the coefficient b_{c2} decreases, reducing the current density the material is capable to transport.

Great deformations in the magnet are not expected, since the magnetic field for the cyclotron should be less than 6 T, it will be assumed that there is no deformation in the magnet [53]. With this decision the coefficient b_{c2} is now constant. In Figure 20 the fit data versus the points given by the manufacturer can be observed. For this case, the correlation is high, around 0.99.

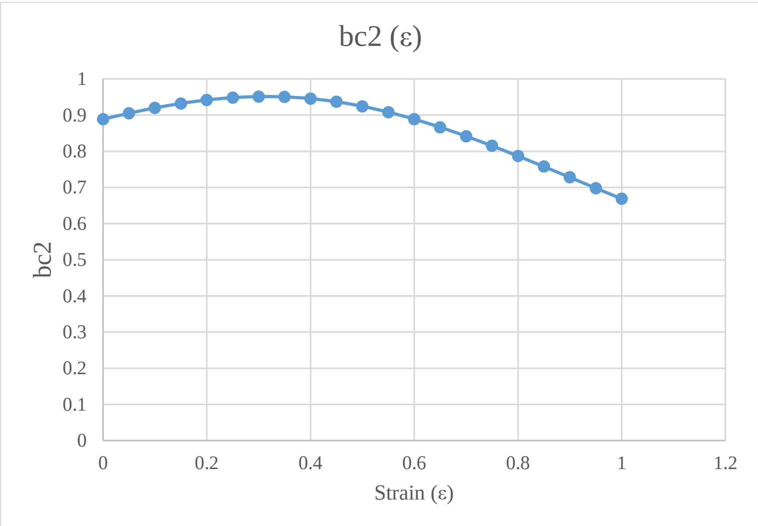


Figure 19. Strain dependence of coefficient b_{c2}

From the critical surface presented in Figure 21, the critical temperature and field of Nb_3Sn can be derived, and they are greater when comparing with the NbTi equivalent. The price, brittleness and therefore the difficulty in the manufacturing process has limited its use. Nowadays, the manufacturing methods has been industrialized and the price has reduced, allowing its use in more applications such as the new FCC magnets that are being developed at CERN [54], which are expected to reach a field up to 16 T.

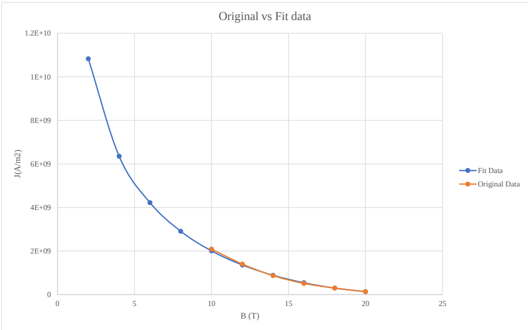


Figure 20. Field dependence of the normalized Jc(B,T) for a Nb_3Sn Supercon multifilamentary wire, measured at 4.2 K.

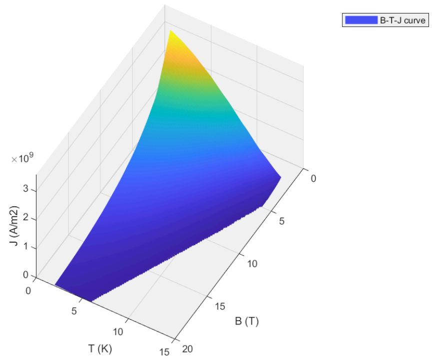


Figure 21. Critical surface from a Supercon NbTi wire

4.3. MgB_2

Magnesium diboride or MgB_2, is a simple binary compound, that has been well-known and widely used since the 1950s. Although its high critical temperature and superconductivity properties had been invisible until its discovery in 2001. This superconductor belongs to the BCS (Bardeen, Cooper and Schrieffer) family (Green circles in Figure 13), among which we have NbTi or Nb_3Sn. The high critical temperature, similar current density properties to NbTi, and relative ease to manufacture makes him an interesting candidate to substitute NbTi in low field applications.

Immediately after the discovery of MgB_2, researchers have been able to fabricate long-length MgB_2 wires using the 50-year experience in the conductor fabrication with low-Tc superconductors and the 20-year experience with high-Tc superconductors.

Wires of few kilometer length can be manufacture in single batches, and are commercially available. The main methods to produce wires and tapes are three: the PiT, the diffusion method, and the coating technique.

PiT, described in 4.2, is the most used method to manufacture MgB_2 wires. The powder can be obtained with either techniques, with precursor materials Mg and B, "in situ", or with prereacted MgB_2, "ex situ".

The diffusion method [55] consists of the cold working of a B-filled tube with a Mg rod embedded axially in it. After a final heat treatment, the Mg diffusion leaves a hollow, but produces a dense MgB_2 layer structure with excellent longitudinal and transverse connectivity.

MgB_2 has certain similarities with Nb_3Sn. Once it crystallizes, it is difficult to operate with. The "wind-and-react" procedure has been commonly used, but latest development show that "react-and-wind" techniques can reach the same quality [56].

4.3.1. MgB_2 Modeling

In this case, the manufacturers provide a large set of operating points, so it is not needed to use scaling laws in order to construct the critical surface, although some authors has proposed some [57, 58]. Nevertheless, it is important to notice the strain dependency of MgB_2 as the literature reported [59]. Since no information is provided from the manufacturers addressing this issue, the tests are expected to be done in favorable conditions where no strain was applied. To address this fact a security coefficient is included.

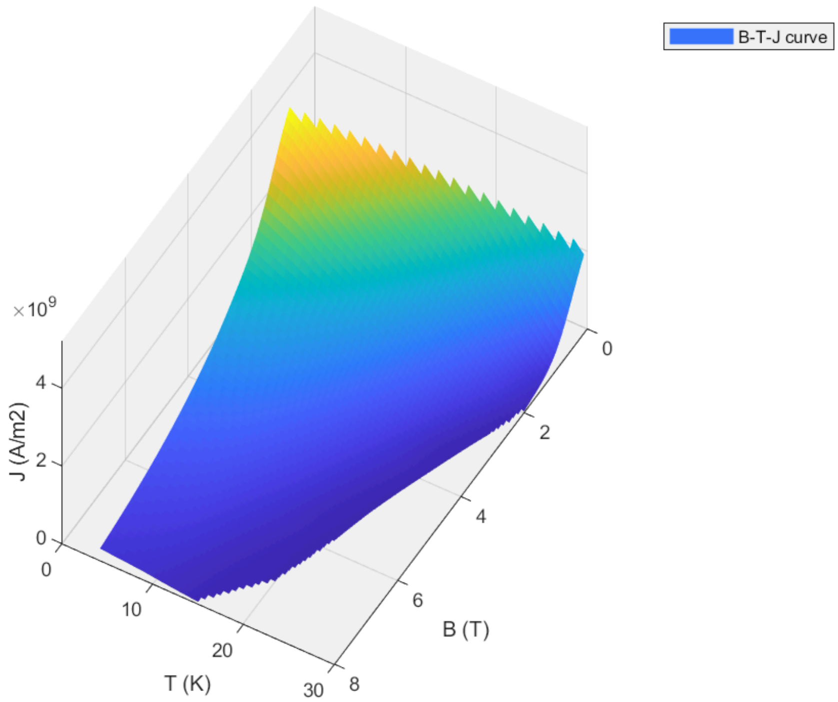


Figure 22. Critical surface from a HyperTech MgB_2 wire

4.4. High temperature superconductors

Since the discovery of superconductivity in 1911 many researches have looked for a superconductor with high transition temperature. In 1986, IBM researchers Bednorz and Muller discovered the first high temperature superconductor, the material they used was La_2CuO_4. For that discovery they were awarded with the Nobel prize in 1987.

Some months later, researches from the University of Alabama and Houston announced the discovery of the first HTS (belonging to the YBCO family) with a transition temperature above 77 K [60]. The main advantage of this type of superconductors is that, working at 77 K they can be cooled with liquid nitrogen. This coolant is easier to work with, than for example liquid helium, used in low temperature superconductors. However, it is also interesting the high critical field they can reach at low temperatures, several tens of Teslas, which could be useful for high field applications.

High temperatures superconductors are not easy to classify, depending on its structure two families of HTS superconductors can be identified: cuprates and iron-based HTS.

4.4.1. Cuprates high temperatures superconductors

Cuprates (Blue diamonds in Figure 13) high-temperature superconductors have been investigated by numerous researches worldwide. Their high critical temperature is the key to extend superconducting applications to new fields.

Their structure consisted on layers of copper oxide, with intermediate layers of rare earth materials, typically yttrium, lanthanum or barium. They are also referred as Rare-earth barium copper oxide (ReBCO), and depending on the Rare-earth we have YBCO (yttrium), or LBCO (lanthanum).

The manufacturing process for cuprates is rather complicated, they are too brittle to be extruded as NbTi. PiT, has been used for BSCCO (Bismuth strontium calcium copper oxide) [61], but it cannot be used for YBCO. Instead, they are usually commercialized as coated tapes or wires. With this process the maximum length achievable is limited to hundreds of meters instead of thousands that are usually needed for magnet applications. Moreover, the price is usually 10-20 times higher than the price for LTS.

4.4.2. Cuprates modeling

The critical surface for an HTS superconductor is shown in Figure 23. In this case, there was no need to use scaling laws, since manufacturers usually test their tapes for a wide range of temperatures.

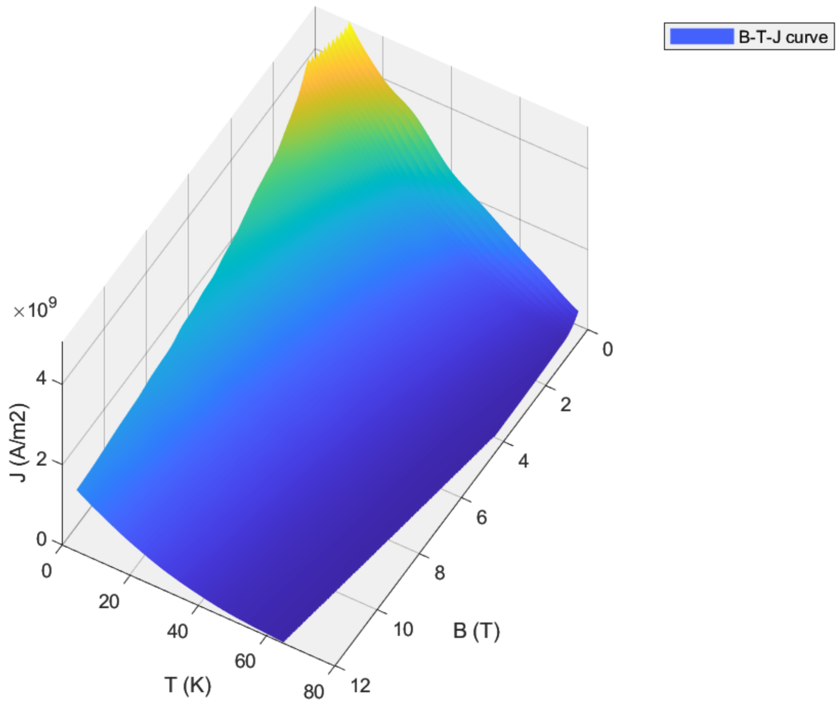


Figure 23. Critical surface from a Fujikura HTS wire

4.4.3. Iron-based high temperatures superconductors

Iron-based superconductors were discovered in 2006. These type of HTS contain layers of iron and pnictogen, typically arsenic or phosphorus. Since their discovery, many families have emerged with temperatures ranging from 20 K to 100 K.

Due to its novelty, they are not available for commercial purposes. Studies have shown that certain types of iron-based superconductors can be manufacture with PiT process [62].

4.5. Superconductors comparison

Comparing superconductors is a multi-variable decision process, and there is no common procedure in order to categorize the superconductors. In this section the author will propose a method to evaluate superconductor viability for a certain operating point.

First the parameters that will be addressed are defined, afterwards the limitations of the method and finally the developed tool to automatize this process.

4.5.1. Parameters

The following categories where superconductors must be evaluated before making a decision were identified:

- Performance: Current density the superconductor is capable to bear without quench risks. It is extremely related with the critical surface of the material, and depends on its structure and composition.
- Price: The price as in most engineering projects is a key variable and cannot be rejected. It is usually given by the manufacturers in $\$/km$.
- Mechanical properties: Ease to manufacture a magnet. As reviewed in the previous section NbTi has high ductility and is prone to solenoid manufacturing while Nb_3Sn, A15 materials or MgB_2 are brittle and require additional actions in order to manufacture a coil.
- Available length: A supply length lower than the required quantity of wire needed will need the design and manufacturing of joints compromising the performance and hindering the design.

To evaluate the superconductor, a cost function will be used, Equation 7, that gives a score for each from 1-10. Only three components of the cost function can be identified, while previously four parameters were defined. The performance and the price are count together using the industry established units $\$/(kA * m)$.

$$Score = A_1 * Score_{Price/Perf} + A_2 * Score_{Length} + A_3 * Score_{Manufacturability} \tag{7}$$

To obtain the $Score_{Price/Perf}$, the price of each superconductor is divided by the current density in the desired operating point obtaining an array of values. The array is normalized and multiply by 10 giving each superconductor a punctuation from 1 to 10 in this category.

Material	*Field (T)*	*Temperature (K)*	*Present* $\$/(kA*m)$	*Projected 5 years* $\$/(kA*m)$ *[63]*
NbTi	6	4.2	0.8	0.8
Nb_3Sn	6	4.2	6.9	4.9
Nb_3Sn	6	10	25	-
MgB_2	6	4.2	19.6	3.5
MgB_2	6	10	26.8	4.9
HTS	6	4.2	16.2	-
HTS	6	10	19	-

Table 10. Present and projected price performance of current available superconductors

The $Score_{Lenght}$ is computed as $10 * e^{-0.5*N_{joints}}$. A non-linear function was chosen to express the non-linear effort needed to elaborate n number of joints, the resources and work needed to move from none to one joint are not the same as moving from one to two joints.

For the $Score_{Manufacturability}$ a score from 1-10 was given to each superconductor based on literature, see previous section, and internal knowledge from CIEMAT researchers.

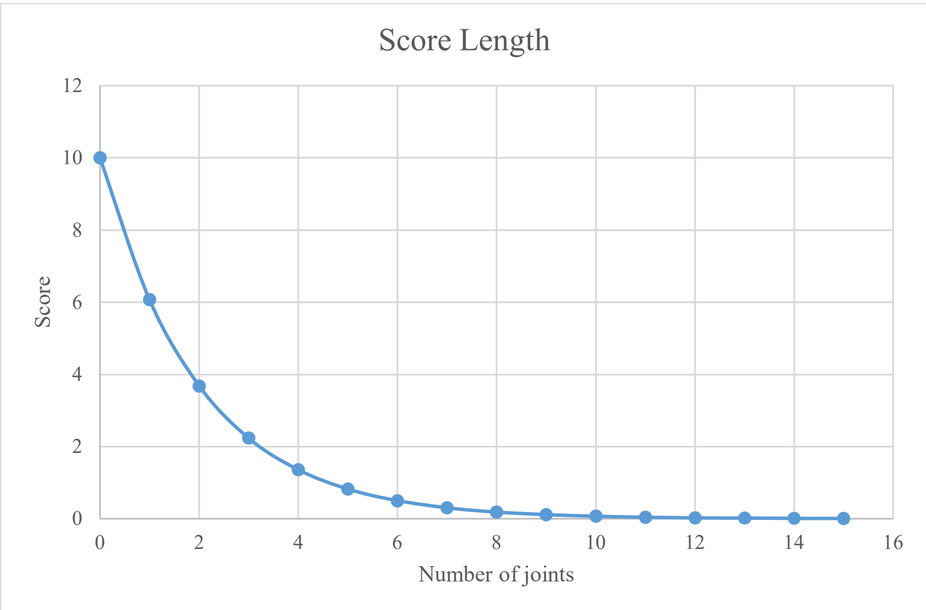


Figure 24. Score versus number of joints

4.5.2. Limitations

The scope of this model is not to give a definitive response of which superconductor is the ideal to use for each situation but to serve as an initial guide on the process. It is important to identify the limitations of the model and areas to improve:

- Minimal quench energy (MQE): It is the minimum energy needed to transition from the superconductivity state to a "normal" state. It is related with the specific heat of the superconductor and the matrix in which it is embedded with. The MQE can increase if the sheath material is replaced with a more thermally and electrical conductive material or when the copper fraction is increased. In this thesis the commercially available tapes and wires are used with the matrix composition given by the manufacturer. Before selecting a certain superconductor, a *quench* study is needed to ensure the viability of the material.
- Geometry characteristics of the magnetic circuit: In a real application the geometry of the coil is linked to the geometry of all the components of the magnetic circuit. The design of the coil and superconducting material cannot be done separately.
- AC applications: This project is focused on superconducting applications for a cyclotron, working in DC. For AC applications it is important to account for the hysteresis losses related to the magnetization of each material.

4.5.3. Comparison tool

A comparison tool was developed and integrated in a MATLAB UI along with the tool developed in chapter 5.

The tool use is very simple, the user define the operating point, the temperature and the magnetic field the magnet is working, and the desire total length of superconductor. The user has the possibility of modifying the weights of each factor, although it is not recommended going to extreme weights and maintain them within the ones proposed by the author to achieve a general view.

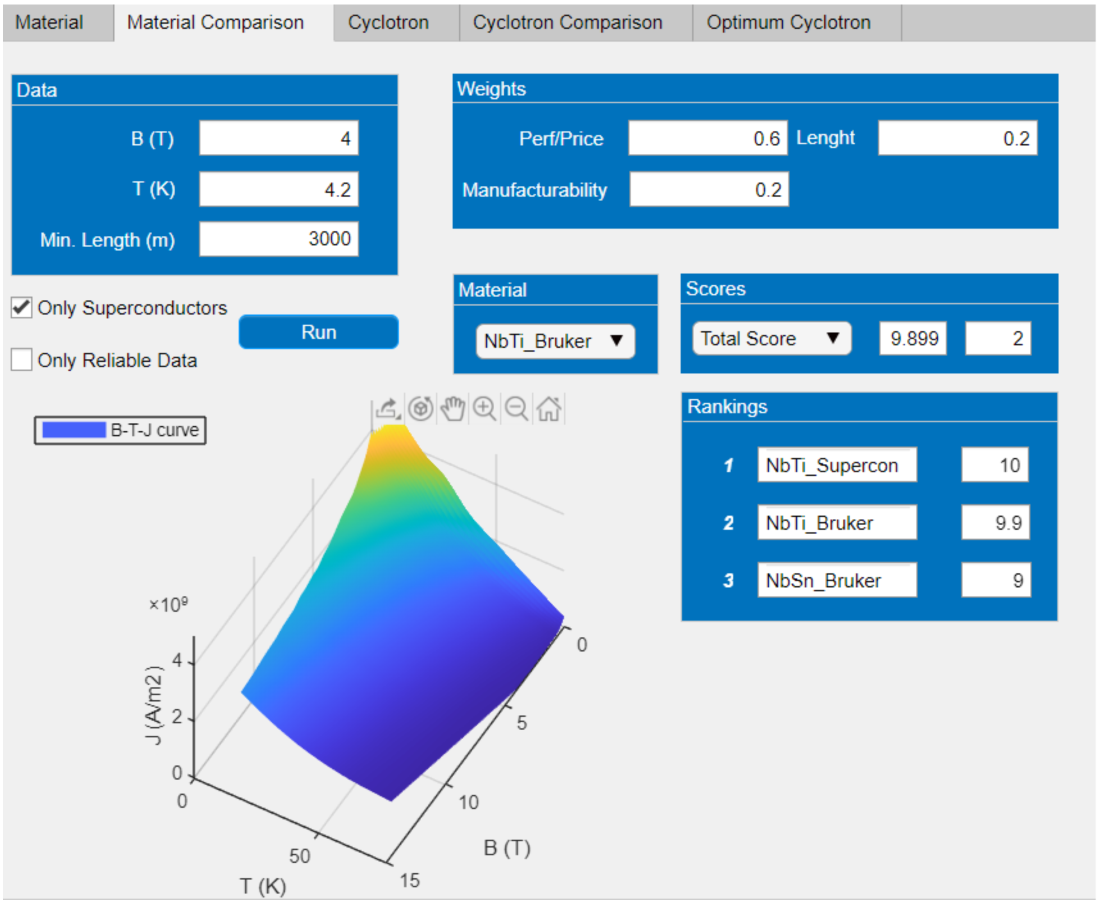


Figure 25. Tab View of the comparison tool developed for an operating point of 4 T and 4.2 K

Once the initial parameters are set up, the user just need to press the "Run" button and the code will do the rest for him. In the rankings panel the user will see the three best performing superconductors for that data input and their score. In the material panel he can change the superconductor in order to see how it performs, changing the score drop down he can change between total score or the score in each of the evaluated sections.

Chapter 5

Automated electromagnetic design of a classical cyclotron

5.1. Introduction

In this section of the project the procedure that was followed to develop a comparison tool for the electromagnetic design of a compact cyclotron will be elaborated. In section 2.5 the steps followed for the current design were reviewed. Three differentiated parts, that required an iterative process, affected the design:

1. Strategic decisions: The objective of the project was to develop a compact cyclotron for radioisotope production. All the decisions taken during the development of the project were aligned with the achievement of this objective. Among which are the decision to use a classical lay-out, or the new refrigeration system developed.
2. Beam dynamics: The beam dynamics analysis provides an input for the magnetic design, the airgap height and the magnetic field shape.
3. Magnetic circuit design: In this phase the materials are selected and the geometry of the cyclotron is fully defined, as well as the operating point.

The process is complex and require a certain number of iterations to reach an optimum or near optimum solution, which is time-consuming and ineffective if the objective is to evaluate the impact of certain changes in the design, e.g., a change in the superconductor material or a new dimension of the height of the airgap provided by the beam analysis. The reasons to automatize are numerous, in which the following were encountered:

- Time: Reduce time spent and obtain a quick solution for a complex problem that could be used as a guide for the line of actions. In other words, have a quick analysis tool that confirms where the efforts must be done to obtain the best solution.
- Superconductor material: Possibility to rapidly analyze the viability of using different superconductors materials and explore the latest developments.
- Yoke material: Analyze the effect of changing the yoke material, e.g., explore the possibility to use a higher saturation field iron which, on the other hand, is more expensive.
- Geometry changes: Examine the effects of geometry changes, e.g., investigate how the iron-coil distance affects the rest of the design or analyze how the aspect ratio of the coil alter the magnetic circuit.
- Wire evaluation: Check that the wire characteristics provided by the manufacturer could be used for this application.
- Conditions changes: Evaluate the effects of changing the temperature and the magnetic field.

The objective is to develop an automated tool for the magnetic circuit design, that will conserve the strategical decisions of the first project and will use the beam dynamics output as a variable input. The advantages were shown above, however it is important to acknowledge the limitations of the developed model:

- Conceptual design: It does not pretend to substitute the whole process, only to guide and serve as a conceptual design.
- The tool is limited by the strategic decisions. The design does not include the option to use an isochronous cyclotron or a different refrigeration configuration.
- Optimum magnetic field shape: The geometry of the pole will be kept as simple as possible and won't reach the ideal magnetic field shape provided by the beam dynamics.

The magnetic design consist on two steps: an analytical solution and a finite element iteration to refine the solution previously reached. The analytical solution will use MATLAB R2019B [64], as a computational software, the details of the magnetic problem will be formulated in section 5.2. The advantages to use an analytical solution are clear, the computing time is very limited and can be useful to analyze changes in the design. A 2D finite element software will be used, Quickfield [34], to evaluate the results obtained with the analytical procedure and automatically iterate to reach a more accurate solution.

The validation of the program will be done with a 3D finite element software, Ansys Maxwell [35], evaluating a certain solution with a 3D model.

5.2. Analytical computation

The 2D magnetic problem consist on an iron yoke with a pot shape and two coils in a Helmholtz lay-out. In Figure 26 a quarter model 2D view of the geometry of the cyclotron is shown, with the defining geometric distances. Due to the non-linearity of iron, the magnetic saturation is about 2 Teslas, the magnetic problem is divided in two.

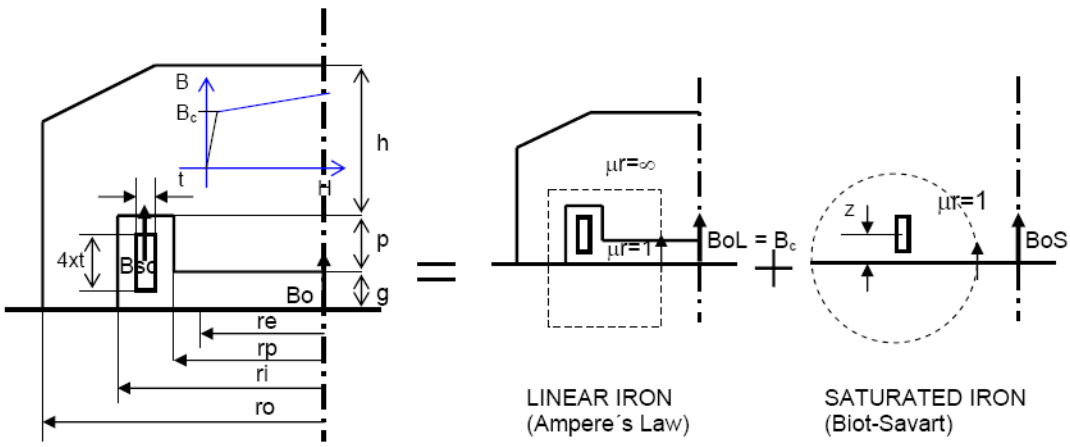


Figure 26. 2D magnetic model and geometry characteristics. Simplification in two cases

As follows, for the resolution of the problem, the following hypotheses are proposed:

- Magnetic field in the airgap will be divided in two. A field equal to the saturation field B_c, and a field created in a non-magnetic medium that would be the difference of the center field and the saturation field.

$$B_o = B_{ol} + B_{os} \tag{8}$$

- The Amperes-Turn necessary for the first field will be computed with the Ampere's circuital law.
- The Amperes-Turn necessary for the second field will be computed with the Biot-Savart law.
- The return path is supposed to be at the saturation limit and the central one is fully saturated.
- The coil will be considered as a filamentary conductor centered in the center of gravity of the coil.
- The extraction radius (r_e), will be considered as an 70% of the pole radius, (r_p).
- The maximum field in the superconductor will follow the expression proposed by the author in Equation 9. The relation between the field at the center and the field at the superconductor is non-linear, depending on geometric parameters (airgap or iron-coil distance), and the superconductor material. This equation was obtained by experimental analysis and as a good balance for the different superconductors involved in the calculation.

$$B_{sc} = B_o * 1.75 * (\frac{2}{B_o})^{0.2} \tag{9}$$

To simplify and provide a clear view of the calculation process, the reader will be guided through a simplified version of it. After, the whole process and the MATLAB implementation will be shown more in detail.

The first step is the input data definition. Among these variables we distinguish three types: initial parameters, geometric parameters and material definition. The initial parameters that can be initially set are:

- Energy reached by the cyclotron. The default value is 8.5 MeV, it is limited to a maximum of 12 MeV when the relativistic effects start to be considerable.
- Operating Point: Magnetic field at the center and temperature of the superconductor.
- Working point: It is initially set to 0.7, and it is recommended to maintain it near to that value.

The geometric parameters that the user can modify as an input are:

- Airgap: This is an input from the beam dynamics, but initially the starting value will be maintained at 74 mm.
- Iron-coil distance: Currently this distance is 32 mm., the minimum technologically possible.
- Aspect ratio of the coil: Relation between the height and the length of the coil.

Finally, the last inputs required from the user are related to the materials used, the user has the possibility to chose between:

- Iron-yoke material: There are four materials in the current database. AISI 1010 [65], AISI 1020 [66], ARMCO steel [67] and Hyperco 50 [68], a cobalt steel.
- Superconductor material: Four superconducting materials are included, NbTi, Nb_3Sn, MgB_2 and HTS from different manufacturers. The database is composed by a total of 11 wires.

After defining the input parameters, the first step is to read the material properties. The steel B-H curve is read and the saturation field of that steel is computed as shown in Equation 10 with the last point of the B-H curve. The superconductor material is read from the database among with a set of points, that with a fitting curve will be used to create the critical surface of the material. The current density *Jc* is calculated evaluating the critical surface at the defined operating point considering the maximum field at the superconductor given by Equation 9.

$$B_{sat} \equiv B_c = B_{steel} - \mu_o * H_{steel} \quad (10)$$

The following steps consist on defining the geometric parameters of the cyclotron. From Equation 3 a relation between the field and the extraction radius of the cyclotron can be derived, that when substituting the proton charge and mass the following expression is obtained:

$$K = 47.89 * (B * r_e)^2 \quad (11)$$

From which the extraction radius (r_e) can be obtained. Then, from the fifth initial hypotheses the pole radius (r_p) can be computed dividing the extraction radius by 0.7. Next the Amperes-Turn necessary to achieve the required field will be evaluated. The total Amperes-Turn are the sum of the Amperes-Turn needed to achieve the saturation field in a linear iron plus the sum of the Amperes-Turn needed to obtain the required field in a non-magnetic medium (First hypotheses).

$$NI = NI_{ol} + NI_{os} \tag{12}$$

From the Ampere circuital law, the following expression is obtained:

$$NI_{ol} = g * \frac{B_c}{\mu_o} \tag{13}$$

To calculate the Amperes-Turn for creating the field in the non-magnetic medium, first the field in the non-magnetic medium is computed and then using a simplified version of the Biot-Savart expression for a coil the Amperes-Turn are calculated.

$$B_{os} = B_o - B_{ol} \tag{14}$$

$$NI_{os} = \frac{r_p * B_{os} * \alpha}{\mu_o} \tag{15}$$

The Biot-Savart expression for a single coil is shown in Equation 16, where the height of the coil, z, and the distance to the center, *a*, are currently unknowns. This is the reason to include the term α in Equation 15, the initial guess for this term is 1.2.

$$B_z = \frac{a^2 * N * I * \mu_o}{2 * (a^2 + z^2)^{\frac{3}{2}}} \tag{16}$$

After evaluating the required Ampere-Turns, the rest of the geometry variables can be calculated. First, with Equation 17, the area of the coil can be obtained, where *WP* is the working point, and *FF* the fill factor (which is the total area of the wires divided by the total area of the coil), with a hexagonal packing a fill factor of 0.9 can be obtained [69].

$$NI = WP * FF * J_q * Area \tag{17}$$

With the aspect ratio provided as an input, the height (l) and width (t) of the coil can be computed. Afterwards, the internal radius of the coil is calculated as in Equation 18, where c is the distance between the coil and the iron.

$$r_i = r_p + t + 2c \tag{18}$$

The external radius (r_o) and height (h) of the cyclotron will be obtained from the fourth hypotheses, where it was supposed that the return path was at the limit of saturation and the central path fully saturated.

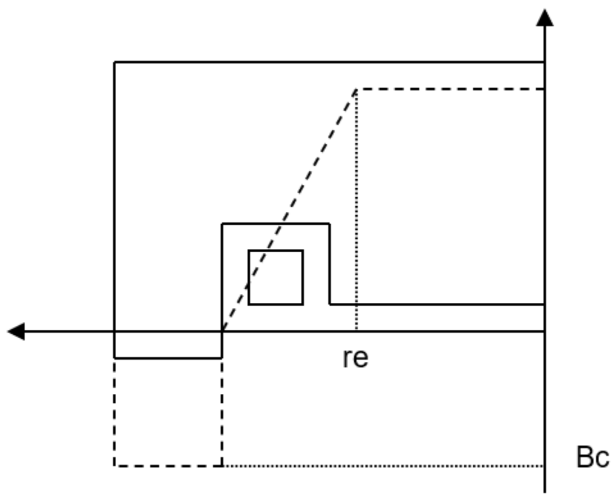


Figure 27. Illustration of the return and central path of the magnetic flux

The magnetic fluxes are represented in Figure 27, and it is assumed that inflow and outflow must be equal. From this supposition the following equations were posed, from which the external radius (r_o) and height (h) are obtained:

$$\pi * r_e^2 * B_o + \frac{\pi * B_o}{2} * [(r_p + c + t)^2 - r_e^2] = \pi(r_o^2 - r_i^2) * B_c * F_{rf} \tag{19}$$

$$\pi * r_e^2 * B_o + \frac{\pi * B_o}{2} * [(r_p + c + t)^2 - r_e^2] = 2 * \pi * r_i * h * B_c \tag{20}$$

In Equation 19, in the second term of the equality, the magnetic flux is multiplied by a factor, F_{rf}. This factor accounts for the hole in the iron-yoke made for the radiofrequency system.

In order to obtain the position of the coil the classical condition of cyclotrons is imposed. This is, that the field at the extraction radius is equal to the field at the center of the cyclotron. This condition is related with the n-index, mentioned in section 2.2, and which value can be computed as in Equation 4. Forcing the field to be equal is equivalent to forcing the n-index to be equal 0. For computational purposes k is defined as the difference between both fields.

$$k \equiv B_o - B_e = 0 \tag{21}$$

Where the field at the extraction radius, B_e is expressed as a function of the elliptic integrals of first and second kind $K(m)$ and $E(m)$.

$$B_e = \frac{2\mu_o * NI * a}{2\pi(2a*y)^{\frac{3}{2}}} * \sqrt{2m} * (a\frac{m}{2-2m}E(m) + yK(m) - y\frac{2-m}{2-2m}E(m)) \tag{22}$$

$$m = \frac{4ay}{(a+y)^2 + z^2} \tag{23}$$

Once the position of the coil (z) is defined, the process could be started again from Equation 16 with a more precise value of z. After a series of iterations an optimum solution should be reached. This is the actual process followed by the code that will be explained at the end of the section.

Next, the height of the pole can be computed as follows:

$$p = z + \frac{L}{2} + c - \frac{g}{2} \tag{24}$$

The procedure to obtain the dimensions of the chamfer is similar to the external radius and height. The returning magnetic flux will follow an elliptic path which radii are the height (h) and the difference between the external radius (r_e) and the internal radius (r_i).

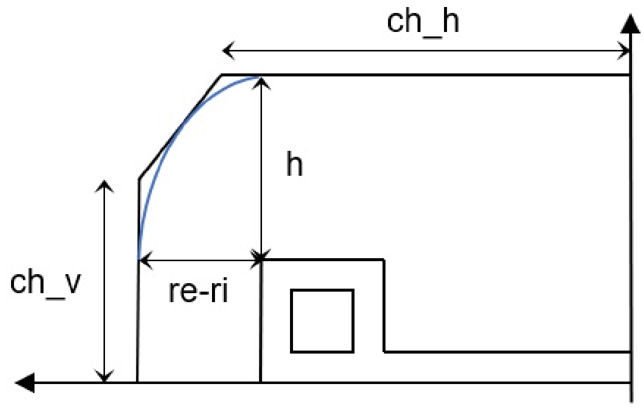


Figure 28. Illustration of the return path of the magnetic flux at the chamfer

To simplify the design and manufacturing instead of an elliptic cut a simple chamfer is drawn. The chamfer is design to be tangent to the ellipsis in the middle point. The coordinates of the chamfer are then obtained as the intersection of the tangent line and the external horizontal and vertical lines of the cyclotron. Two equations define the coordinates of both points.

$$ch_h = h * (\sqrt{2} - 1) + p + \frac{g}{2} \tag{25}$$

$$ch_v = (r_o - r_i) * (\sqrt{2} - 1) + r_i \tag{26}$$

Finally, once defined all the geometry characteristic, the weight, volume of the magnetic circuit as well as the wire length or area can be obtained.

A flow diagram of the code implementation is shown in Figure 30. The process follows the same course described in the previous paragraphs, but an iteration must be done with the position of the coil (z) and the Amperes-Turn necessary (NI) until the variable k reaches a value of 0.

Imposing the condition of $k = 0$ can create convergence problems during the iteration, to sort this out the condition is sometimes relaxed. Moreover, the ideal position for the coils is usually interfering with the mid plane hole for the radiofrequency, so a lower limit is included in the position of the coil.

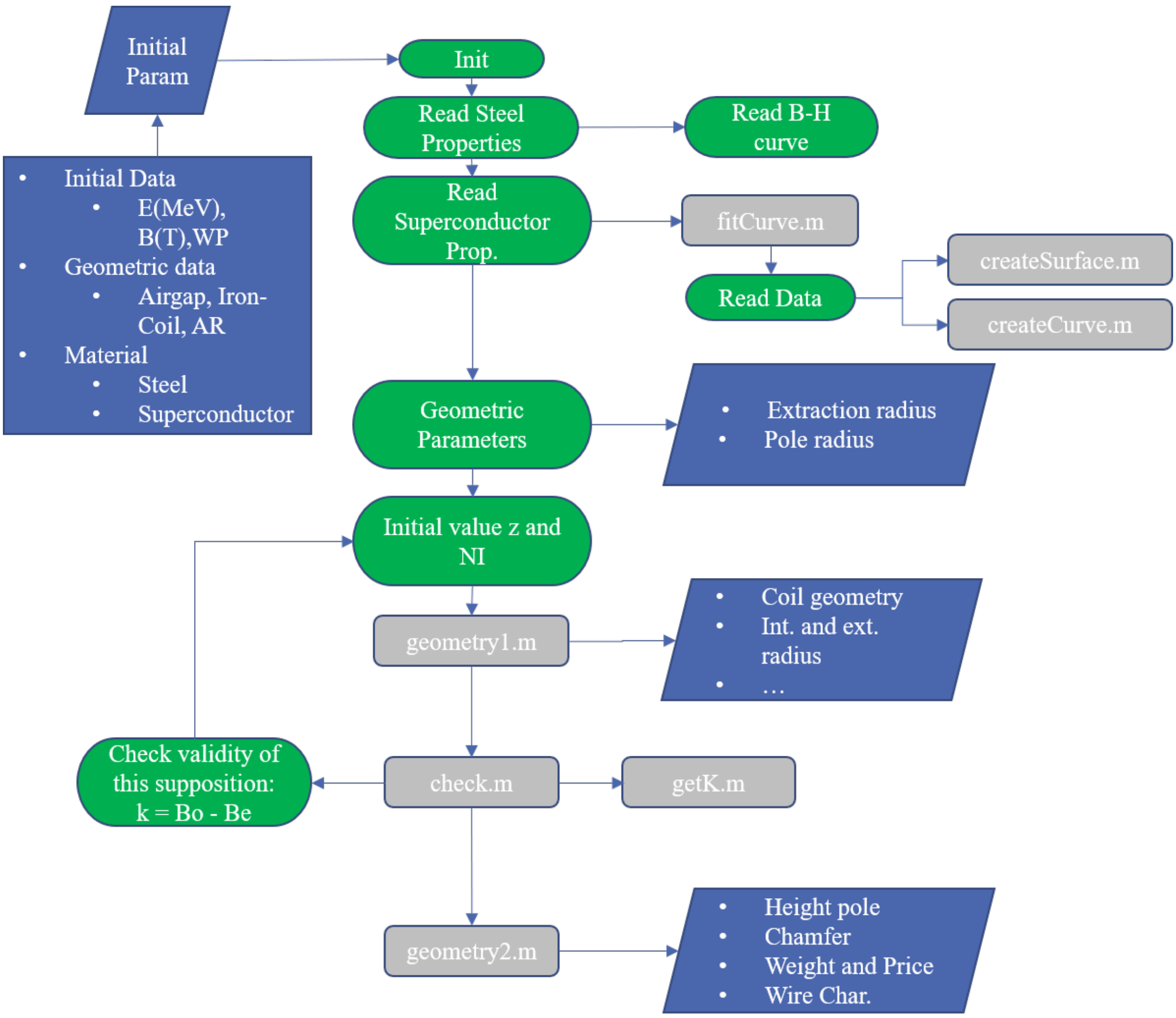


Figure 29. Schematic flow diagram of the custom code to determine the geometry parameters of a cyclotron

5.3. FEM iteration

The analytical process can provide a quick solution in just a few seconds but is based on certain simplifications and hypotheses that if not fulfill could lead to a wrong geometry. In other words, the analytical solution should be taken as a starting point or guess of the real solution. To check the validity of the solution a 2D finite element software, Quickfield, will be used in conjunction with MATLAB. The whole process will be automated from MATLAB.

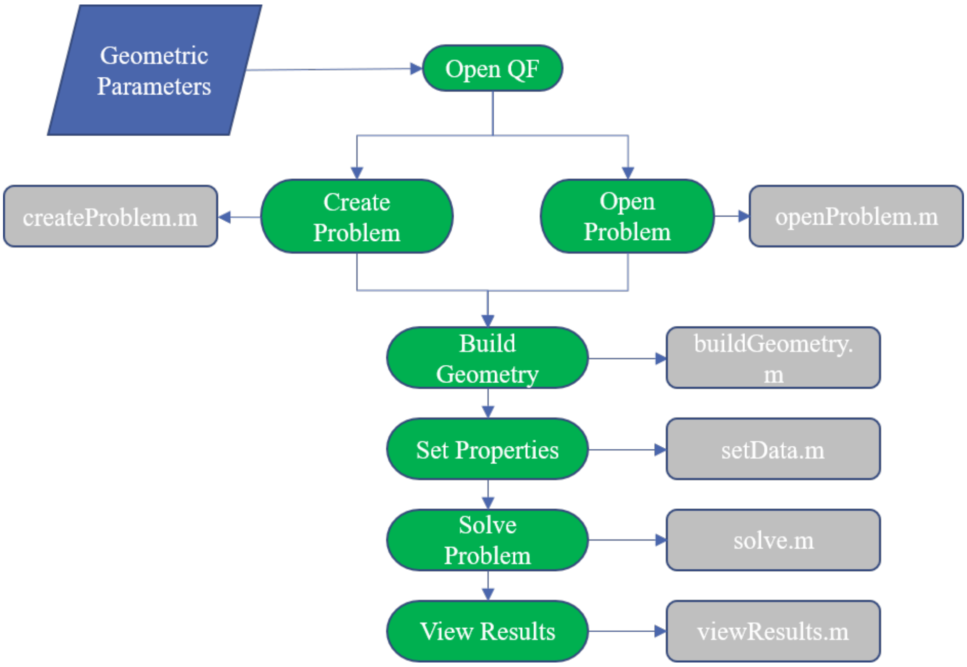


Figure 30. Schematic flow diagram of the RunQF.m function

The solution reached from the analytical procedure serves as an input for Quickfield, afterwards the process is equivalent as if it were done by hand. The problem is defined, the geometry built, the material properties and boundary conditions set and finally the problem is solved.

The problem is defined as a magnetostatic problem with axial symmetry. It is important to notice that the symmetry axis in QF is the X axis of a classical Cartesian coordinates system. The geometry is built with the data from the analytical solution, and afterwards, the mesh is shaped. Certain geometries of the model, the superconductor or the mid plane, need more refinement and have a smaller mesh element. The mesh was refined by a trial and error procedure, where the mesh elements were reduced until no significant differences were found between solutions. A balance between precision and computational time must be carried to reach an optimum point.

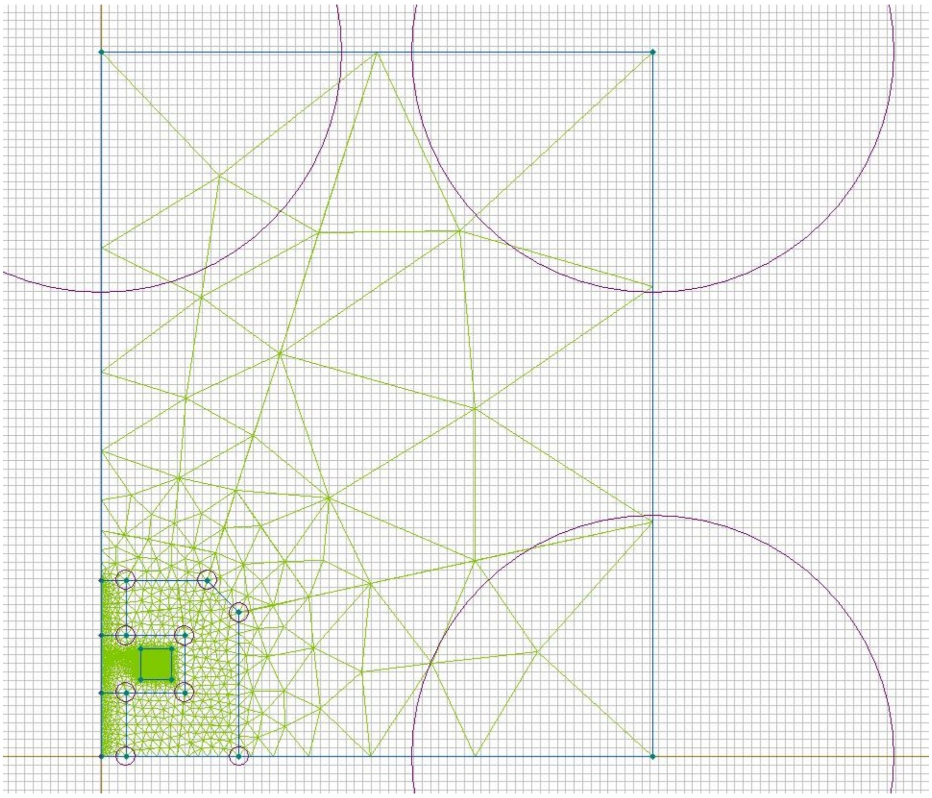


Figure 31. Mesh view of a 2D model cyclotron

Afterwards the material properties and boundary conditions need to be specified. Regarding the materials properties, the permeability and current density were defined for each block:

- Iron-yoke material: The B-H curve of the selected steel was exported to Quickfield.
- Superconductor: The permeability of superconductor materials is usually considered to be equal to the permeability of copper. This is a good approximation for most superconductors but certain materials contain ferromagnetic materials in their matrix. It is the case of MgB_2. The modeling for this type of material will be reviewed in the next section. In addition to the permeability, the current density of the coil is defined.
- Air: A permeability equal to the vacuum permeability is considered.
- Iron-Air block: The objective of this block is to take into account the hole for the radiofrequency in the 2D model. For that the permeability is defined as the weighted average of the iron and air permeability.

The boundary conditions set were the following:

- Symmetry condition along the Y axis.

- Zero magnetic potential in the outer region. The region is defined far enough away for the magnetic field to fully develop.

Once all parameters are characterized, the case can be solved. In Figure 32, the flux density and magnetic field lines of a certain solution can be appreciated.

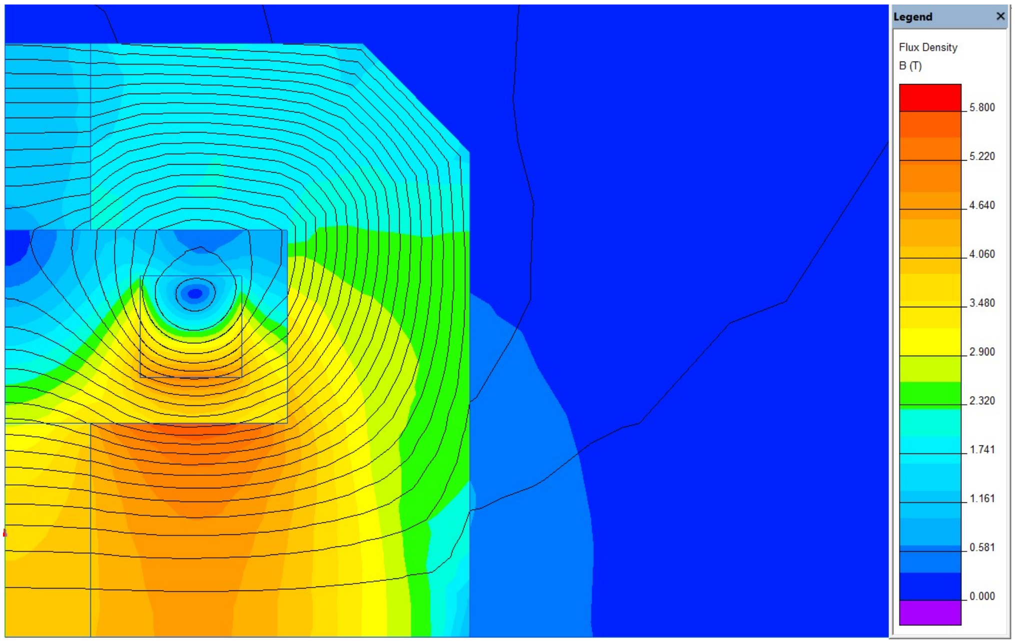


Figure 32. Magnetic flux density of a 4T NbTi cyclotron

The solution reached with QF can be used to corroborate the hypotheses made during the analytical procedure. Moreover, an iterative process using the data from the QF solution could be implemented to reach a more precise solution. The hypotheses that could be check are:

- Maximum magnetic field in the superconductor: Obtain a more precise value of the maximum field in the superconductor to appropriately select the operating point.
- Field characteristics at the mid plane: In this case the magnetic field at the center and the n-index can be checked.

The process is implemented in a MATLAB function, as shown in Figure 33. Two functions are used to obtain the problem results from QF, one to extract the real maximum magnetic field in the superconductor, and another to obtain the magnetic field at the center and the n-index. The conditions that must be fulfilled are that the field at the superconductor is the same as in the previous iteration and the magnetic field at the center is equal to the initial input, everything while minimizing the value of the n-index. For that, the position of the coil (z) and Amperes-Turn (NI) are again iterative variables.

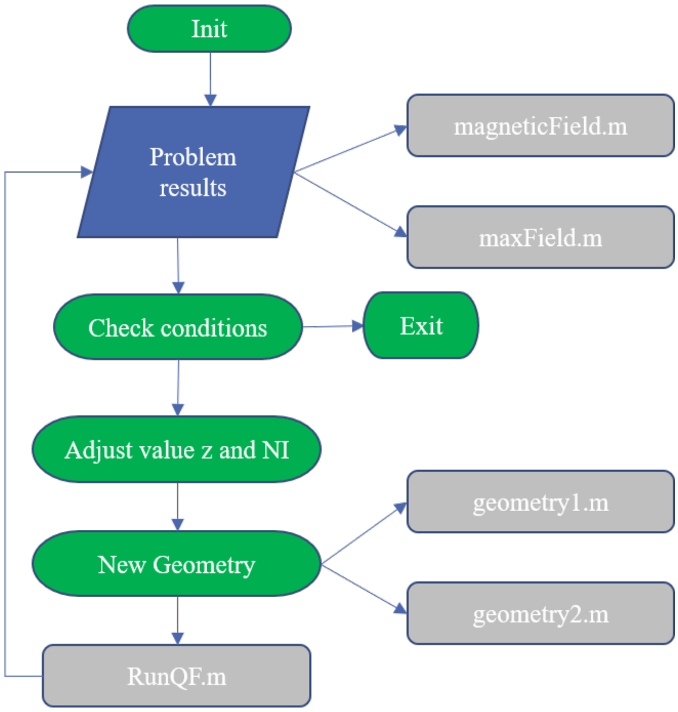


Figure 33. Flow diagram of the MATLAB function OptimizeQF.m

5.4. Implications of superconductors embedded in a ferromagnetic matrix

For the resolution of the magnetic problem it was supposed that the permeability of the superconductor was equal to the vacuum permeability as the constituents (mainly copper) that conforms the matrix do not include ferromagnetic materials. However, the addition of copper to an MgB_2 superconductor is counterproductive, since during the heat treatment a strong reaction between magnesium and copper produces a degradation in the critical current. For that reason is typical to substitute copper with a ferromagnetic material, iron, nickel or monel (A nickel-copper alloy).

The implications of these facts are of extreme importance for the magnet design, as no longer the vacuum permeability could be used as an estimator of the permeability of the matrix. Instead, the problem becomes non-linear and for its resolution a numerical approach is required.

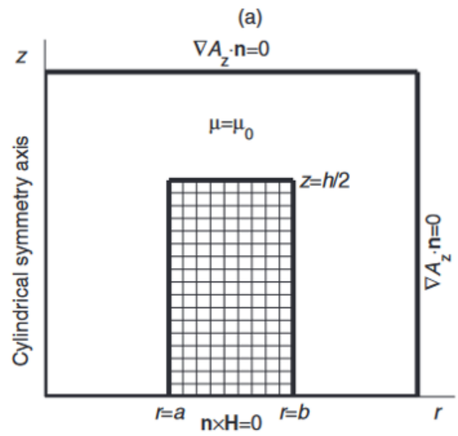


Figure 34. Detailed unit cells coil model [70]

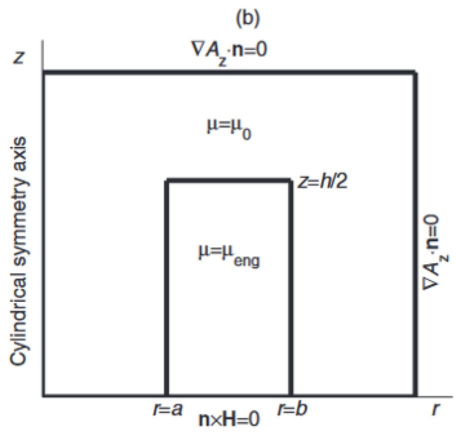


Figure 35. Homogenized unit cell coil model [70]

Typically, for the measurement of a wire, a magnetic flux density, B_{gen}, is generated by an external magnet and applied to the magnet. For most of the conductors, that only contain diamagnetic materials, the generated field B_{gen} is equal to the magnetic field the sample is confronted with, B_{app}. However, when ferromagnetic materials are present, B_{gen} and B_{app} do not need to be equal. The presence of the iron/nickel sheath increases the measured critical current, specially at low fields when the ferromagnetic material is not saturated.

To simplify the problem and maintain the computational time low, the procedure elaborated in [70] will be adopted. First the detailed coil is replaced with a homogenized coil with an engineering permeability and then a mapping for the critical current of the homogenized coil model must be elaborated.

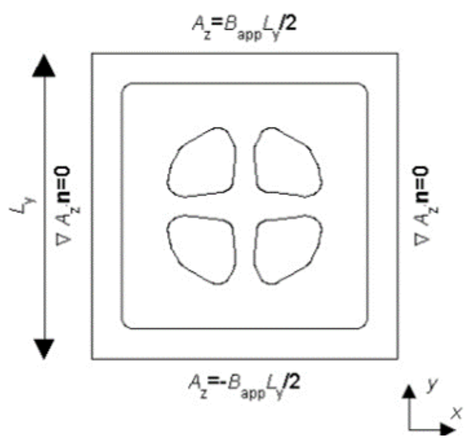


Figure 36. Unit wire with the appropriate boundary conditions for permeability computation [71]

First, as mentioned above, the engineering permeability of the wire must be computed. In Figure 36 the model and the boundary conditions for the problem are shown. On the right and left sides of the unit cell Neumann-type boundary conditions are used to maintain the flux perpendicular to that face. On top and bottom faces Dirichlet-type conditions are used to control the flux density that flows through the unit cell. Once the problem is solved, the engineering permeability can be computed as the average magnitudes of B and H in the x direction:

$$\mu_{eng} = \frac{\int_S B_x ds}{\int_S H_x ds} \tag{27}$$

The relative engineering permeability of a certain wire is presented in Figure 37, where it can be noticed the current and field dependence.

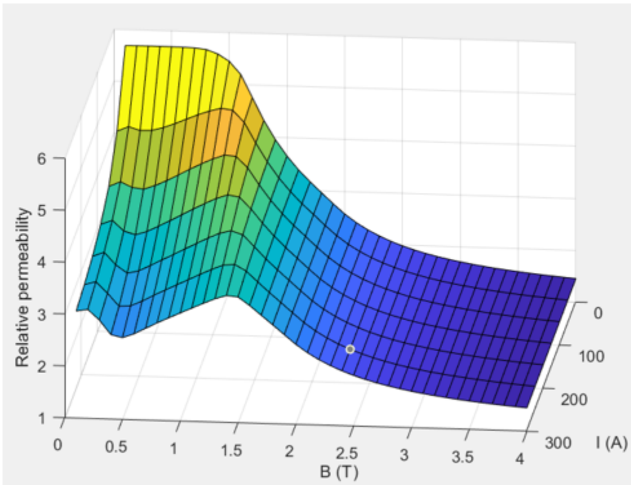


Figure 37. Relative engineering permeability for a certain unit cell

B_{max} is the maximum magnetic field at the unit cell supposing homogenized properties. A model or map to compute the relation between B_{gen} and B_{max} is presented in Figure 38. This model will use the previously obtained engineering density for the homogenized unit cell. Once a mapping is obtained, the critical current of the homogenized unit cell can be computed as:

$$I_{c,coil} = I_c(G(B_{max})) \tag{28}$$

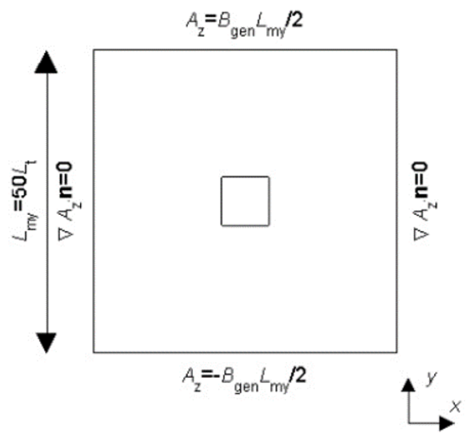


Figure 38. Homogenized cell unit with the appropriate boundary conditions to compute the mapping $G : B_{max} \to B_{gen}$ [71]

The relationship between B_{gen} and B_{max} is show in Figure 39. It is peculiar that the current does not affect the curve. The reason is that the field created by the wire is extremely small when compared with the total field.

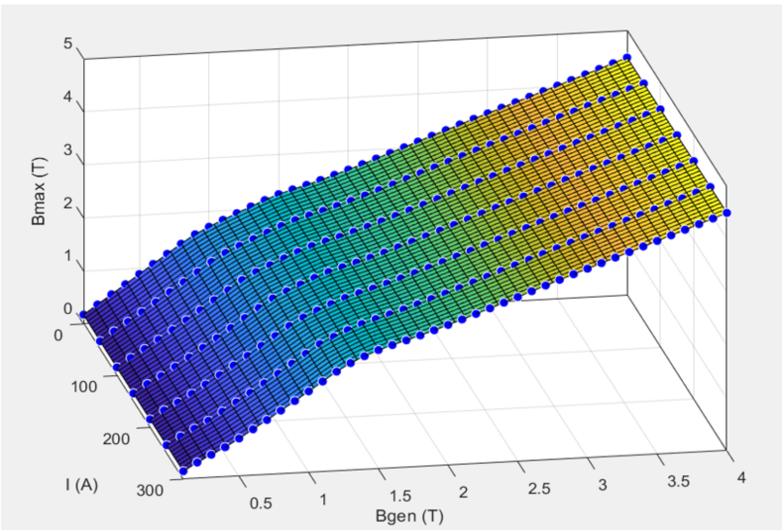


Figure 39. Mapping $G : B_{max} \to B_{gen}$ for a certain MgB_2 wire

A MATLAB code was elaborated to obtain both surfaces for different MgB_2 wires. The input is the matrix structure and composition and the output, the equivalent engineering permeability and the B_{gen} and B_{max} mapping. Both surfaces were introduced to the code developed in the previous sections, but certain modifications were made.

The process starts with the analytical computation of the cyclotron, where a magnetic field at the superconductor is supposed depending on the magnetic field in the center, Equation 9. Then the engineering permeability is obtained from the engineering permeability curve, Figure 37.

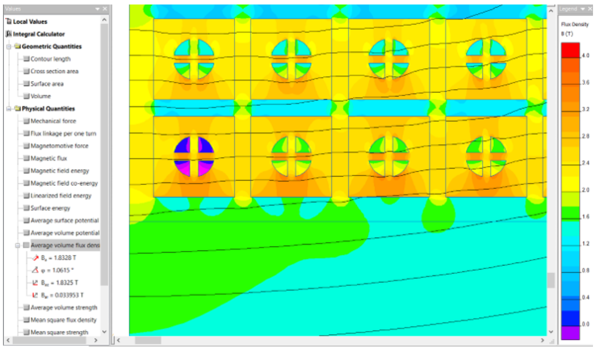


Figure 40. Coil with detailed ferromagnetic unit cells, maximum field in the superconductor is 1.84 T

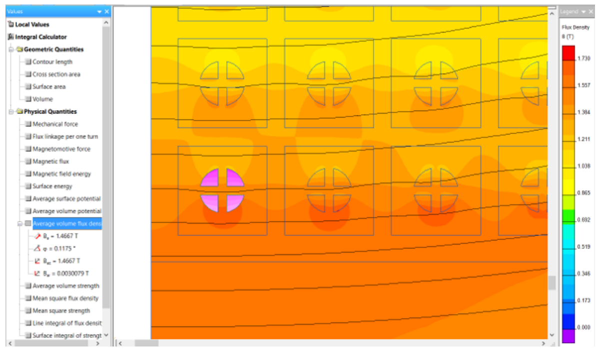


Figure 41. Coil with detailed paramagnetic unit cells, maximum field in the superconductor is 1.47 T

When the case is run in QF, B_{max} is obtained from the results, but is converted to B_{gen} using the mapping function to get the actual magnetic field in the filaments of the superconductor. This process is repeated within the iterative function explained in the previous section.

The results from this process were surprising, as the cyclotron weight was typically a 10% greater than the equivalent with a non-ferromagnetic matrix. The preconceived idea was that the presence of ferromagnetic materials in the matrix would have acted as a shield for the superconductor allowing it to reach higher density currents. However, the opposite effect was observed.

For single wires when applied an external magnetic field the shielding effect can be observed, but when the wires are arranged in a coil the magnetic flux density increases inside the superconductor due to presence of the ferromagnetic materials. The worst case cell has to shield a greater field than the equivalent worst cell for a diamagnetic cell coil as shown in Figure 40 and Figure 41 (The scales in the image are different). In both figures the same coil, with the same current, is tested but the matrix permeability is different.

5.5. Results validation

The solution reached with the system does not pretend to be definitive, although that should not be a limitation to be precise. To check the validity, a certain case will be run in a 3D software simulation program, Ansys Maxwell. In this case, the geometry reached for a NbTi superconductor operating at 4.2 K and with a center magnetic field of 4 T will be translated to this software.

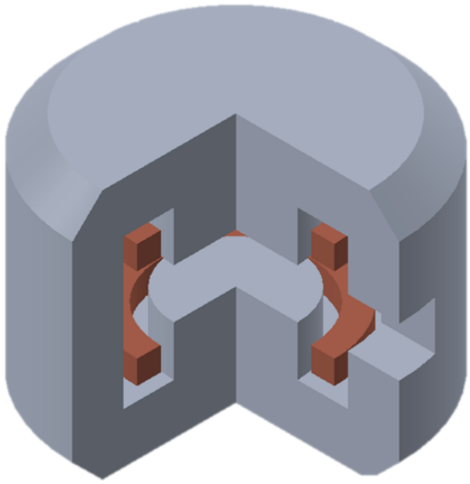


Figure 42. 3D Magnetic circuit model of a NbTi cyclotron

Firstly, a 3D model is built using Solidworks, the result is shown in Figure 42. The geometry is then exported to Ansys Maxwell. The model can be reduced just to an eighth of the total due to symmetry. Once the geometry is ready, the boundary conditions need to be set. In this case the conditions are relatively simple, first the air region, where the problem is going to be analyzed, is set to be sufficiently large that all the magnetic field is solved. The external faces of this region automatically have Neumann conditions so there is no need to add any symmetry conditions to the lateral faces. Although, in the bottom face (in the mid plane), a symmetry condition is needed.

Afterwards, the excitations are defined, in this case is opted for defining the current density at the superconductor, although the Amperes-Turn could have alternatively been used. This value is translated from the QF solution. Ansys Maxwell uses an energy convergence method, where the mesh is refined automatically after each iteration until the energy percent error trespasses certain value input from the user. For this case, an energy error below 0.005 % was reached. Furthermore, certain mesh operations were defined initially to obtain a more precise image of the mid-plane and the superconductor. The final mesh, after 14 iterations, is shown in Figure 43.

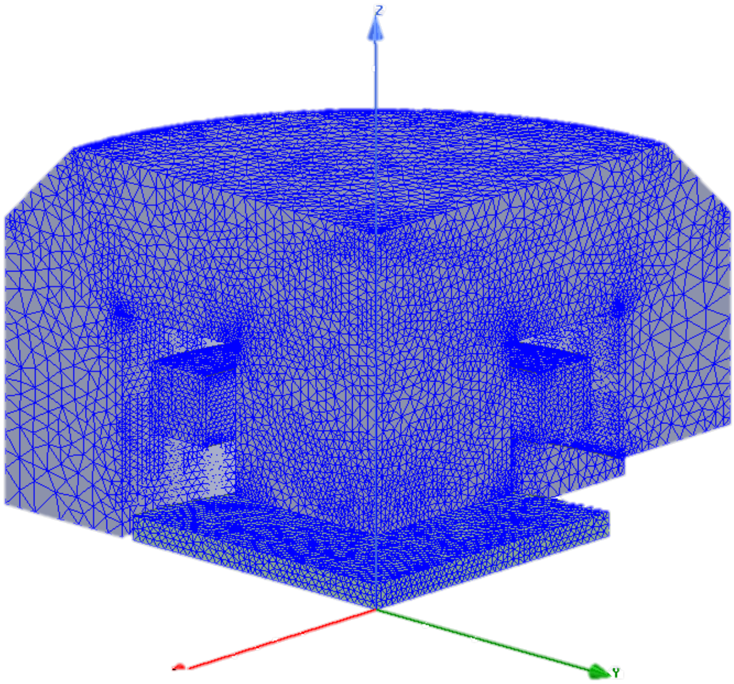


Figure 43. Image of the mesh from Ansys Maxwell, a quarter cylinder is defined at the mid-plane to improve precision

Once the simulation reaches a solution, the validation is conducted comparing the magnetic field at the mid plane and the magnetic field at the superconductor obtained in QF and Maxwell. In Figure 45 and Figure 47 the differences between both can be noticed. The maximum field in the 3D model is 4.07 compared to the 4.01 T obtained in Quickfield. The difference is below 2%.

The field in the superconductor given by Ansys is 4.77 T, greater than the 4.72 T obtained with Quickfield. In this occasion the difference is around an 1%. The differences are not especially notable, but it is important to remember that a higher field in the superconductor implies a change in the operating point and a decrease in the current density. Nevertheless, if the current density is adjusted in order to obtain a magnetic field of 4.00 T in the center the field at the superconductor will decrease within the same order.

The results are extremely similar, although certain difference can be appreciated. The reason of this difference were thought to be related with the hole left for the radiofrequency. The cases were run again, without the hole, in Quickfield and then in Ansys and this thought was confirmed. In this case both cases converge to the same magnetic field value, as show in Figure 46 and Figure 49. It should be taken into consideration that as the airgap increases the difference between the 2D and 3D model would be larger.

It is important to notice that although the 3D model could give a more precise solution, the computational time required to reach a solution is significant greater than with the 2D model. While with the 2D a solution can be reached through various iterations in under 2 minutes, the 3D simulation takes more than 180 minutes just to run a single case.

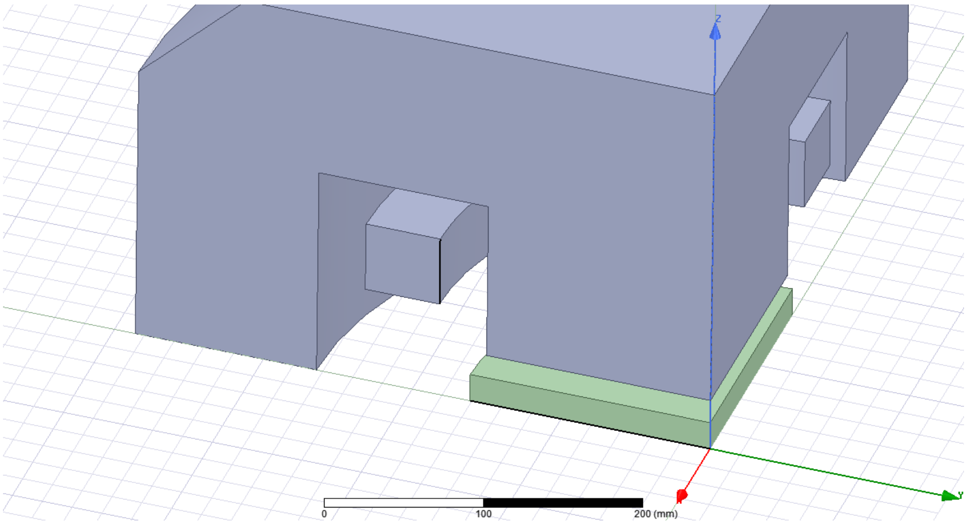


Figure 44. Image of the lines (in black) where the measurements where done

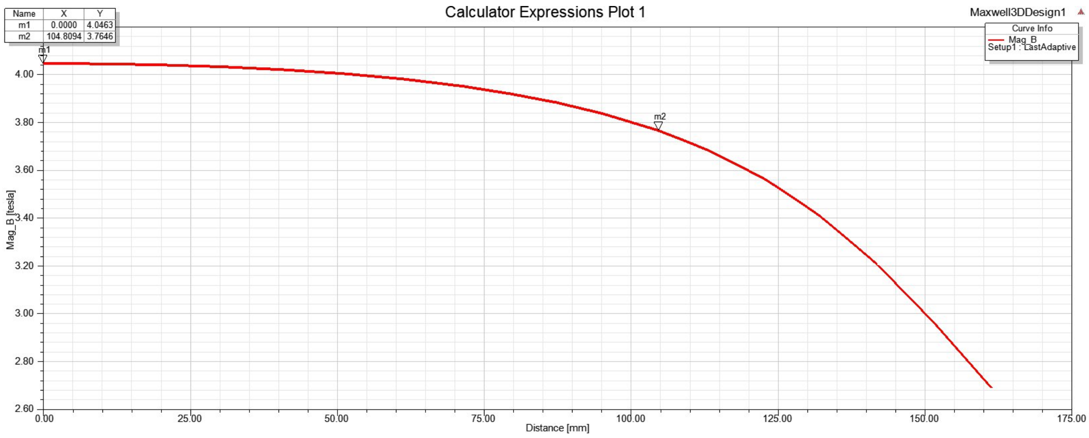


Figure 45. Mid plane field in a 3D model solved with Maxwell, the maximum field is 4.07 T

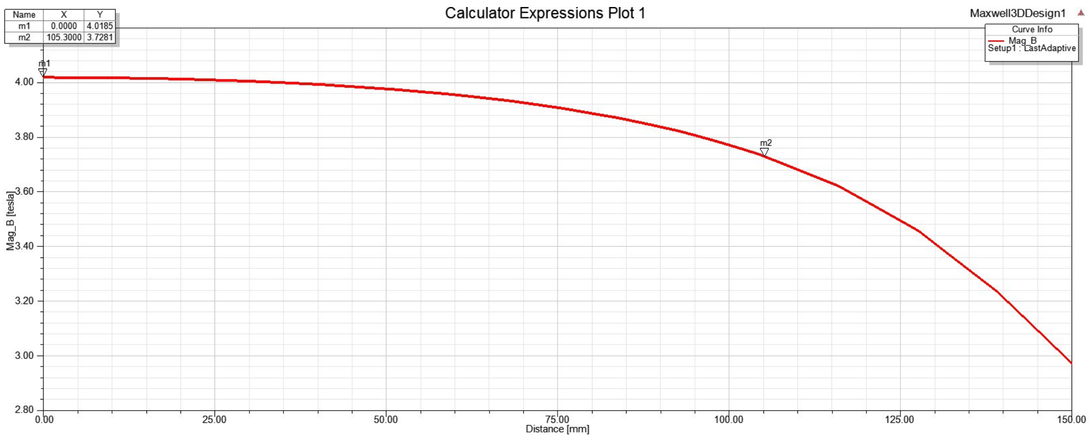


Figure 46. Mid plane field in a 3D model without radiofrequency hole solved with Maxwell, the maximum field is 4.01 T

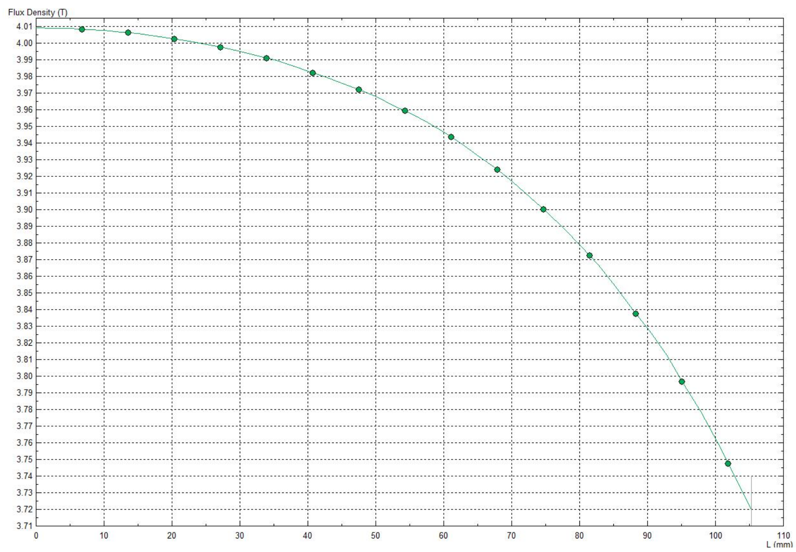


Figure 47. Mid plane field in a 2D model solved with Quickfield, the maximum field is 4.01 T

Figure 48. Internal face field of the superconductor obtained with Maxwell, maximum field is 4.77

Figure 49. Internal face field of the superconductor without radiofrequency hole obtained with Maxwell, maximum field is 4.73 T

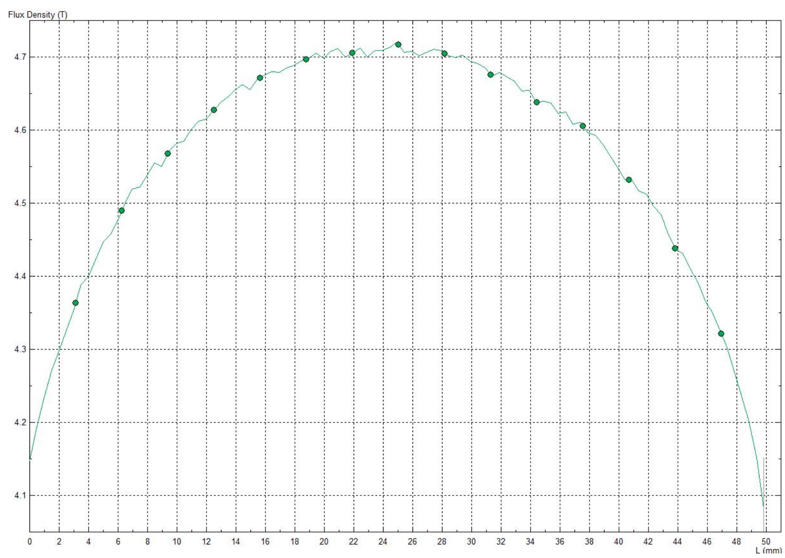


Figure 50. Internal face field of the superconductor obtained with Quickfield, maximum field is 4.72

5.6. User Interface Tool

A user interface (UI) was developed to facilitate the human-computer interaction and allow an effective operation and clear understanding of the modifications.

A total of seven different tabs were developed within the same application. The first two were mentioned in subsection 4.5.3, an allows the user to make a quick comparison of the different superconducting materials. The other five were developed to allow the user compare different configurations for the design of a cyclotron. They are based on the code explained in the previous sections.

In the next sections the five tabs and the motivation to use each one will be reviewed.

5.6.1. Cyclotron geometry

In this tab the user has the possibility to change or modify all the inputs, listed in section 5.2, and see the results of these modifications to the cyclotron design.

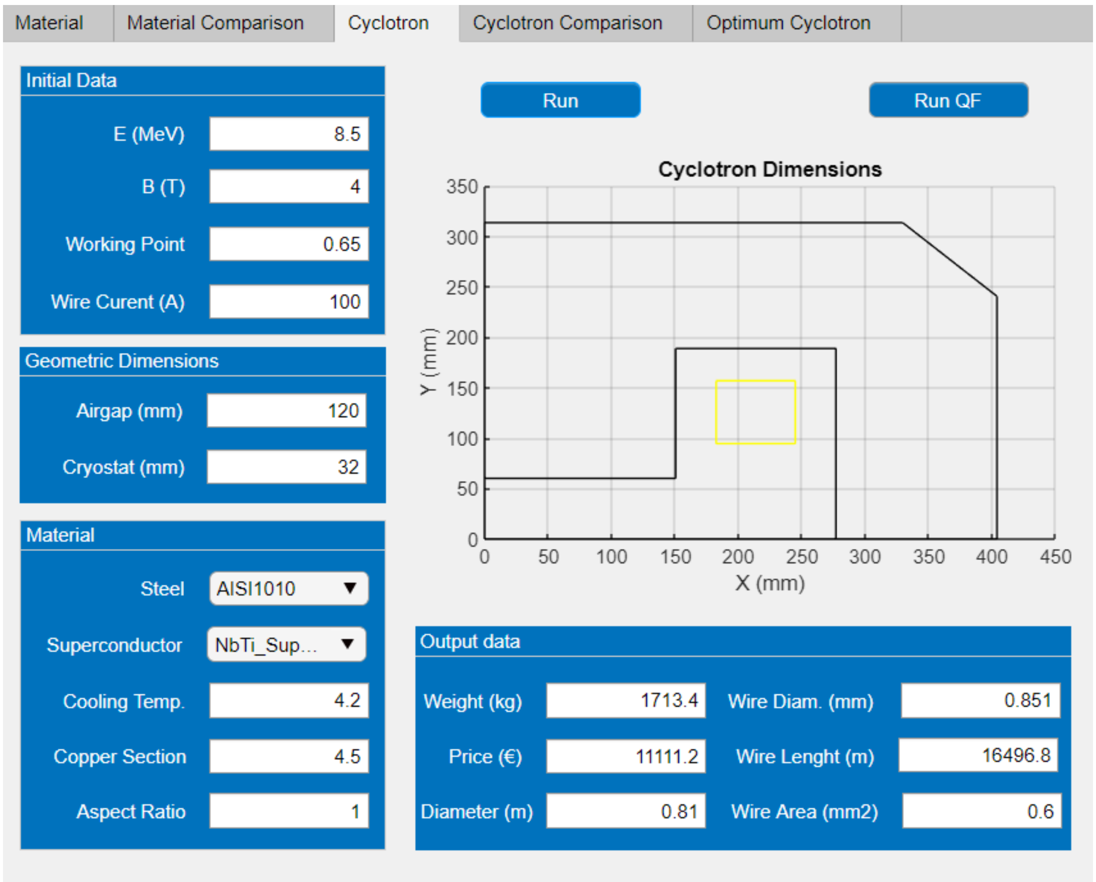


Figure 51. View of the cyclotron design tab

The inputs can be modified at the left part of the UI. Once the user has finished with the modifications he can press the "Run" button and the problem will be solved analytically. The cyclotron geometry will be shown in the X and Y axes and the output data (Weight, price/ length of the superconductor, etc.) will be shown in the lower panel.

If the user presses the "Run QF" button the case will be solved analytically and using the FEM iteration function, for a more precise solution. The results will be shown similarly.

5.6.2. Optimum cyclotron

The previous tab can give the user a quick idea of the dimensions of a cyclotron but it needed to run several trials to reach the optimum cyclotron e.g., the cyclotron with minimum weight or the cyclotron with minimum price. To facilitate the analysis in this new tab the program iterates through all the possible magnetic field values and store the output values, weight and price.

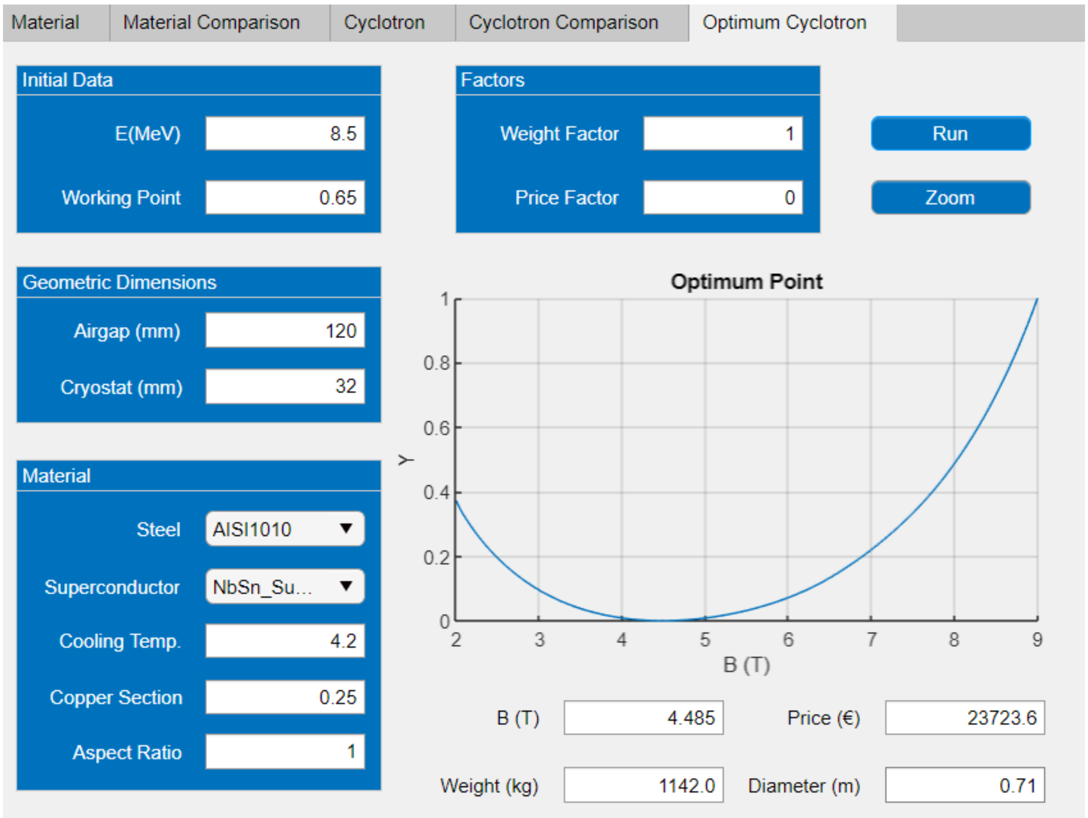


Figure 52. View of the optimum cyclotron tab

Once the program has iterated through all the possibilities, a cost function is used to compare the set of points. The weight and price of each point are normalized and multiply for a user-defined factor. E.g. in Figure 52, the minimum weight cyclotron for a Nb_3Sn wire is obtained at 4.485 T.

$$Score = A_1 * Weight_{Normalize} + A_2 * Price_{Normalize} \tag{29}$$

The whole iteration is done solving each case analytically, for a more precise solution the "Zoom" button can be pressed. In this case the user could define the starting and ending magnetic field value and the step size, and the program will do a FEM iteration. It takes a longer time, proportional to the number of cases, so it has to be used cautiously. Usually, the case could be solved analytically obtaining a curve as in Figure 52, and "Zooming" between 4 and 5 T for a more accurate solution.

5.6.3. Cyclotron score

The magnetic circuit of the cyclotron cannot be evaluated only in terms of weight and price, there are more factors to take into account in the design. Some of them were addressed in chapter 4, where the superconductors were evaluated in terms of price/performance, manufacturability and length.

$$Score = A_1 * Score_{Weight} + A_2 * Score_{Price} + A_3 * Score_{Length} + A_4 * Score_{Manufacturability} \tag{30}$$

The objective of this tab is to translate this method to the cyclotron design, in order to obtain a 1-10 score for each superconductor. In this case the price/performance evaluation will be done separately in price and weight. The resulting cost function is shown in Equation 30. The weight of the price/performance factor is divided equally between price and weight (0.3 each). To compute both scores, the weights and price are normalized and a score is obtained using the following formula: $10 * (1 - Normalized factor)$.

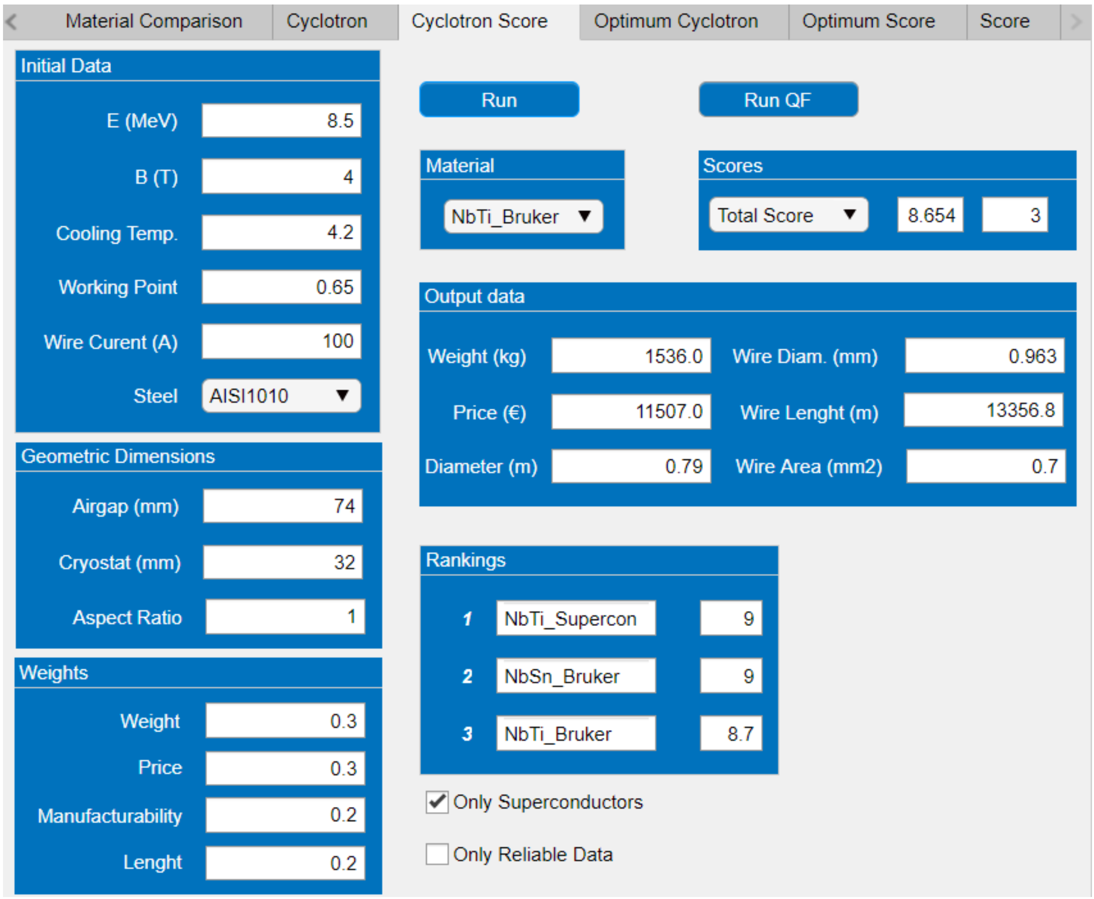


Figure 53. View of the cyclotron score tab

In this case the user will define the initial data similarly to the first tab and the program will solve the magnetic problem for all the superconductor materials. The user has the option to solve the problem analytically for a quick analysis or using FEM iteration for a more precise evaluation. For each case the weight, price and needed length are stored in an array. Once it has evaluated all the materials, the program proceeds just as the superconductors material comparison program, where it gives a score for each section. Finally, the total score is obtained by using Equation 30.

5.6.4. Optimum cyclotron score

This tab is a combination of the last two tabs. With the tab described in subsection 5.6.3, the user can compare different cyclotrons designs in the same operating point, however the optimum operating point for NbTi is different from the optimum operating point for MgB_2, resulting in various simulations until the optimum for both is reached. With the tab described in subsection 5.6.2 the user can evaluate the optimum operating point for a certain superconductor. Although he cannot compare between them.

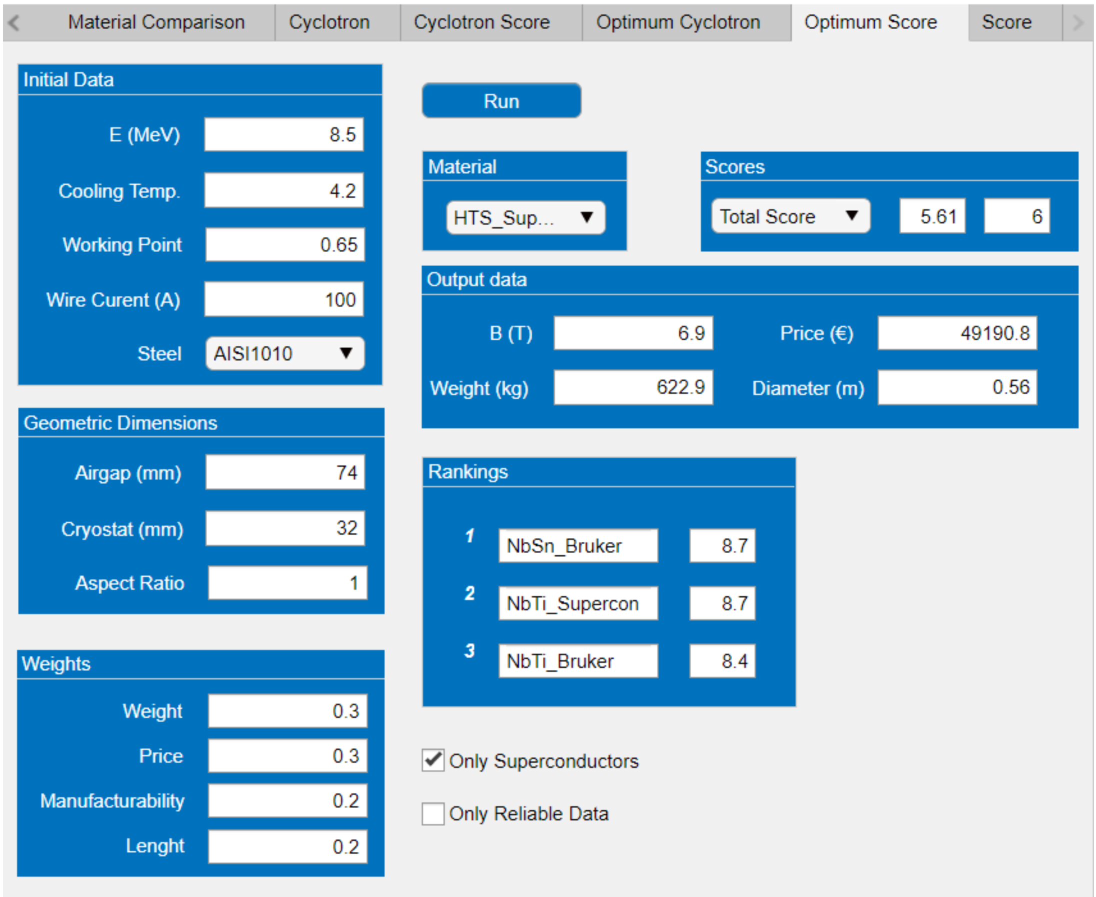


Figure 54. View of the optimum cyclotron score tab

In this tab the program iterates through all the materials and magnetic field values. It starts with the first superconductor and find its optimum point, afterwards it continues the iteration with the second material. Once it has iterated through all the materials and found their optimum operating point, it uses the same evaluation method used in subsection 5.6.3, generating a score for each material.

5.6.5. Score

One last tab was designed in order to obtain the score between four different materials. Usually the user will compare between NbTi, Nb_3Sn, MgB_2 and HTS, from which he will select the best performing wire.

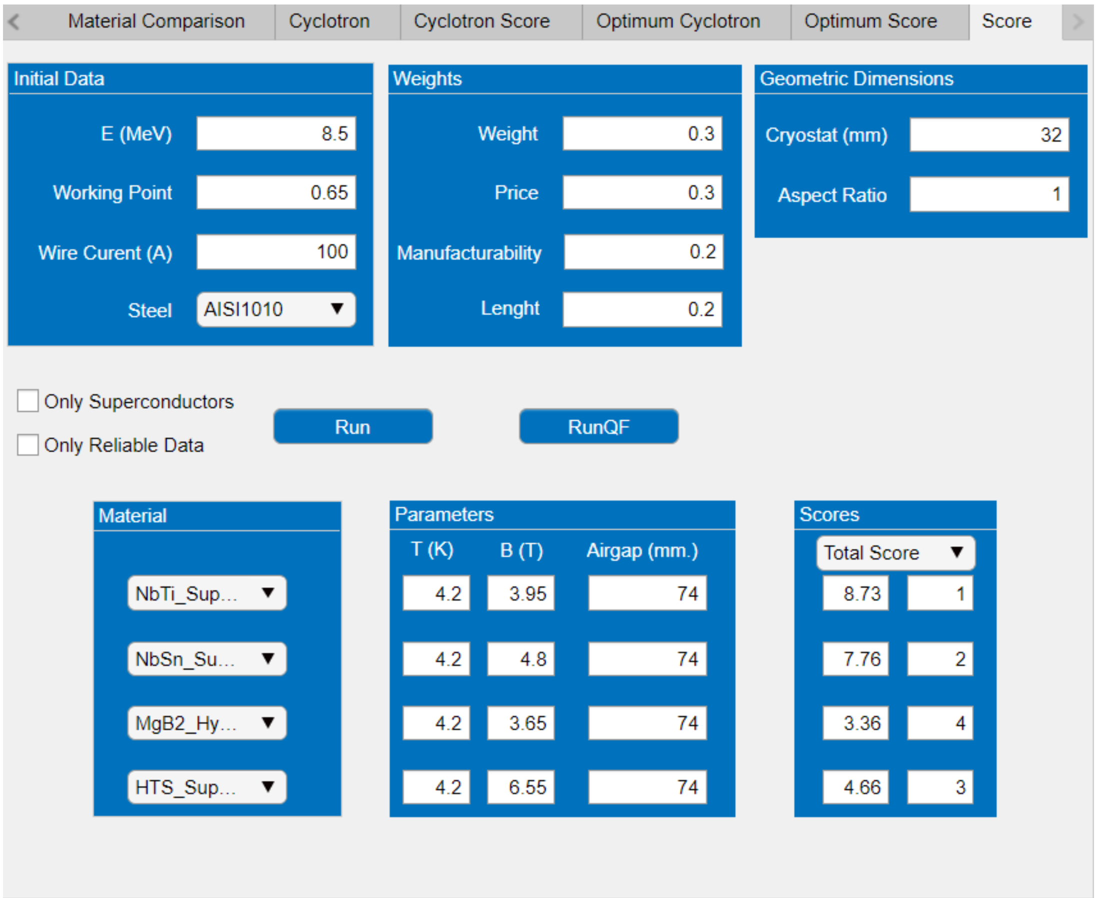


Figure 55. View of the score tab

This tab allows comparing between wires at different operating points solving the case analytically or using the FEM iteration function, and obtaining a score for each one. The score is obtained comparing the performance, weight and price for each solution, so the user needs to be aware of the conditions he is evaluating each material. In other words, it is a relative comparison and the score changes when comparing the same material with a different one.

Chapter 6

Analysis of results

This section will serve as a guide on how to use the comparison tool developed and explained in the previous section, and as an actual analysis for the future developments for the CYCLOMED company.

In the first section an updated analysis with the same conditions as the current cyclotron will be done. All materials will be again evaluated, contrasting if NbTi is still the best option. A second analysis will be done at 10 K, this time in line with CYCLOMED's proposal to raise the operating temperature of the cyclotron.

Finally, in the last section it will be explored how the election of different initial parameters affects the design.

6.1. Copper coil cyclotron

The copper coil cyclotron will serve as the base comparison for the alternative superconducting materials. The initial conditions are shown in Table 14, and ignoring the temperature are the same as for the superconductor analysis.

Material	*Weight (kg.)*	*Diameter (m.)*	*Price (€)*	*Center Field (T)*
Copper	14,526	1.62	28,446	1.94

Table 11. Minimum weight copper coil cyclotron

The optimum operating point for copper is near 2 T, the magnetic saturation of steel. The weight is near 15 tons, while with the use of superconductor materials it is usually below 2 tons (as shown in next section). Confirming the weight reduction by a factor of 8-12 addressed in subsection 2.4.2.

Diameter, instead of volume, will be used to compare cyclotrons, since usually, the space restriction does not come from the height of the room but its wide and length.

6.2. Superconductors analysis working at 4.2 K

In this section an analysis of superconductors working at 4.2 K will be carried. To maintain sensitive data from manufacturers secret, the superconductors materials will be referred only as such and the original manufacturer will not be mentioned. Moreover, only the best performing wire for each material will be considered.

The first step to start the analysis is to define the initial parameters. In this case, the analysis will be done with the same parameters of the actual cyclotron. These parameters are shown in Table 14.

Parameter	*Value*
Energy	8.5 MeV
Cooling Temperature	4.2 K
Load factor	65 %
Nominal current	100 A
Steel	AISI 1010
Airgap	74 mm.
Iron-coil distance	32 mm
Aspect Ratio	1

Table 12. Initial parameters for superconductor analysis at 4.2 K

One of the key parameters in the design is the copper to non-copper proportion. For NbTi, the value of this parameter will be maintained at 4.5, the same as in the current cyclotron. For the other materials a *quench* study should be done to obtain a precise number, but it is out of the scope of this project. For the rest of materials, the standard copper to non-copper proportion provided by the manufacturers will be used.

Once the initial parameters are defined, the analysis can proceed. It is a good idea to start with the "Optimum Score" tab (subsection 5.6.4), to obtain a broad overview of the behavior of each superconductor. In this case the minimum weight cyclotron was obtained for each.

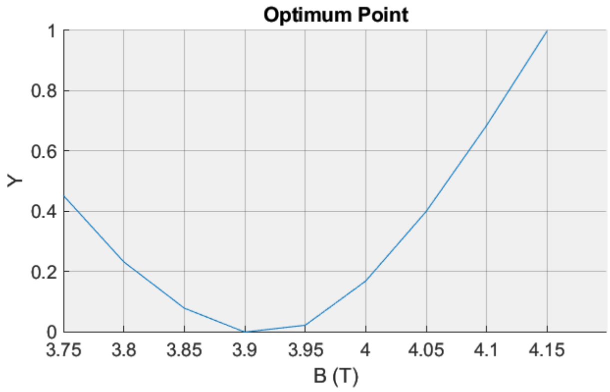
The superconductor with the highest score is NbTi, although with Nb_3Sn and HTS a lighter cyclotron can be obtained. The difference on the scores is related to price, which in this case NbTi is the cheapest and to the manufacturability of the coil and the length of the wire. In chapter 4 all the addressed superconductors were reviewed, in which it was found that coil manufacturing with Nb_3Sn or HTS was tedious as it implies using a "wind-and-react" process. Moreover, HTS superconductors have a very limited length, which means it would need a certain number of joints.

Material	*Score*	*Weight (kg.)*	*Price (€)*	*Center Field (T)*
NbTi	8.7	1,473	7,855	3.4
Nb_3Sn	8.05	930	20,595	4.7
MgB_2	3.62	1,510	93,941	3.1
HTS	5.61	622	49,190	6.9

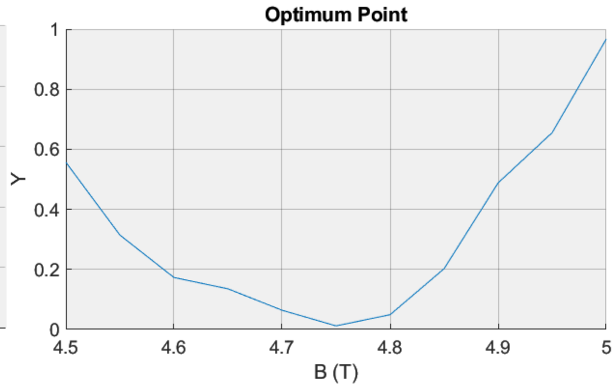
Table 13. Minimum weight cyclotron for each superconductor material working at 4.2 K

NbTi, seems at first sight as the best option, but it needs to be contrasted. Next, using the "Optimum" tab (subsection 5.6.2), and running the cases in QF, the optimum operating point will be confirmed for each case.

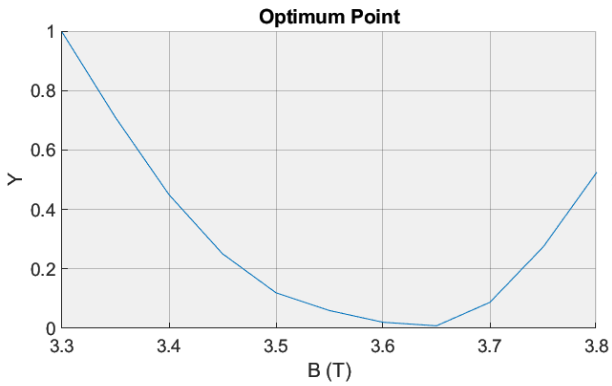
Certain differences can be appreciated, the optimum point for NbTi is reached at 3.9 T (which corresponds with the 4 T of the current design), while with the analytical solution the optimum was reached at 3.4 T. Similarly for MgB_2 the magnetic field was undervalued, however for the HTS coil it was overvalued. It is always important to confirm the values obtained with a FEM software, if not it can lead to wrong solutions. The differences are related to the initial suppositions made at the beginning of the analytical computation, specially related to the limitations of the proposed equation for the maximum magnetic field at the superconductor, Equation 9.



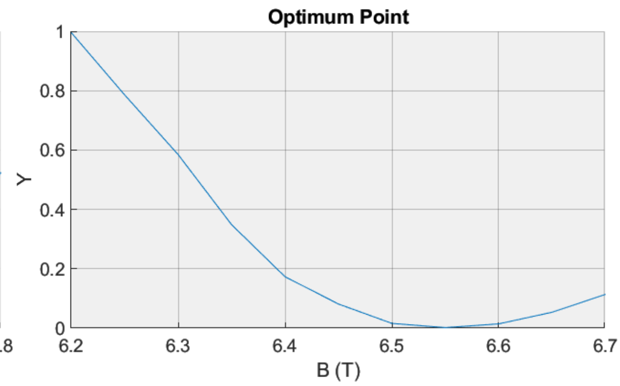
(a) NbTi



(b) Nb_3Sn



(c) MgB_2



(d) HTS

Figure 56. Normalized weight versus field for different superconductors working at 4.2 K

With the new optimum values of the magnetic field, Table 13 is updated and the minimum weight cyclotrons for each material have the following characteristics.

Material	*Weight (kg.)*	*Diameter (m.)*	*Price (€)*	*Center Field (T)*
NbTi	1,197	0.76	5,385	3.9
Nb_3Sn	926	0.66	21,126	4.8
MgB_2	1,259	0.76	76,128	3.65
HTS	665	0,56	59,634	6.55

Table 14. Updated Table, minimum weight cyclotron for each superconductor material working at 4.2 K

MgB_2 can be quickly discarded as, for this temperature, the heavier and most expensive cyclotron is obtained. The other three options offered certain advantages in certain points, NbTi is the cheapest, easier to manufacture and the coil could be done in one piece. With Nb_3Sn, a lighter cyclotron can be obtained at a reasonable price (The total price of the cyclotron is around 1.5M€), with a single piece of wire, on the other hand the manufacturability of the coil is problematic. With an HTS cyclotron the weight reaches its minimum value, on the other hand, the price is 10 times the price of NbTi, moreover it needs around 15-20 joints due to its limited length and the coil manufacturing is as problematic as with Nb_3Sn.

To obtain a rational answer, the user could use the "Score" tab (subsection 5.6.5) and compare the solutions reached in Table 14.

Material	*Score*
NbTi	8.73
Nb_3Sn	7.76
MgB_2	3.36
HTS	4.66

Table 15. Final score for each superconductor working at 4.2 K

The order is not altered from what was initially obtained. MgB_2 is not a real alternative for this temperature, and HTS superconductors are still an immature technology but with great potential. The decision is between NbTi and Nb_3Sn, having the first a slight advantage due to its ease to manufacture and low price.

6.3. Superconductors analysis working at 10 K

6.3.1. Advantages of working at 10 K

One of the possibilities for the future designs of CYCLOMED, is developing a cyclotron working at higher temperatures. This possibility allows to relax the operating conditions of the refrigeration system since a higher quantity of heat could be extracted. To quantify the increased performance, the COP for the whole system could be computed for both temperatures. COP, or coefficient of performance, is the ratio between heat (Extracted or supplied) and the work done by the system. It is a good indicator of the operating expenses of the system.

$$COP = \frac{Q}{W} \tag{31}$$

According to the Carnot cycle, it can be shown that an ideal machine working at maximum efficiency follows the following relation $\frac{Q_H}{T_H} = \frac{Q_C}{T_C}$, resulting in the following expression for a cooling machine:

$$COP = \frac{Q_C}{Q_H - Q_C} = \frac{T_C}{T_H - T_C} \tag{32}$$

The COP, can be now computed for both temperatures. In both cases a temperature of 300 K was selected as the hot heat reservoirs.

Temperature	*COP*
4.2 K	1.45%
10 K	3.45%

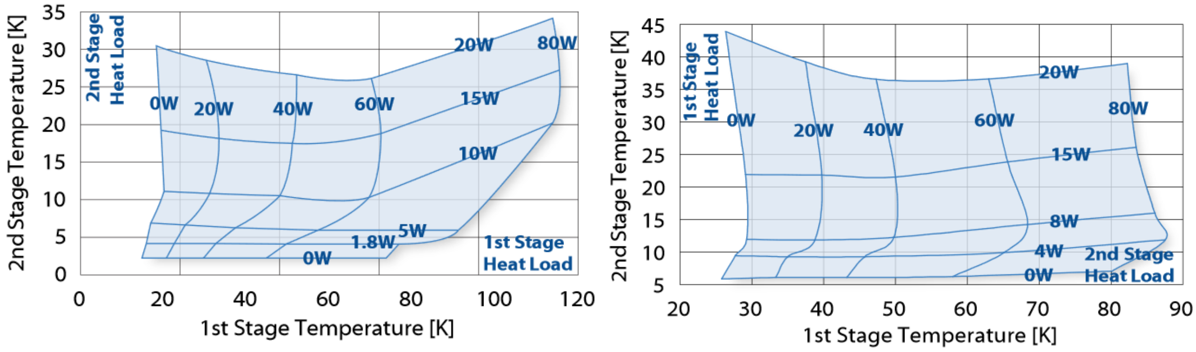
Table 16. COP (in %) for the refrigeration system working at 4.2 K and 10 K

Typically, the COP can reach values over 1 (100% if expressed in percentage), since it converts to heat the work applied and the pumps in additional heat from the heat source. Although, in this case due to the low value of the temperature from the cold source, little heat can be extracted, resulting in low values of COP.

Nevertheless, even if the COP have a low value, it almost tripled just by increasing the temperature to 10 K. Another option to analyze the impact of increasing the operating temperature is to examine the heat capacities of the available cryocoolers working at the mentioned temperatures.

The cryocoolers are based on Gifford-McMahon cycle. In Figure 57 the heat capacity map for the most recent and powerful cryocoolers is represented. The first one has a heat extraction capacity at the second stage of 1.8 W while working at 4.2 K [72]. The second cryocooler, has a heat extraction capacity of 5.4 W [73]. The heat extraction capacity is tripled, which the corresponds with the previous results. Even the same cryocooler (RDK-418D), increases its heat extraction capacity to 10 W at 10 K, as shown in Figure 57.

The main disadvantage of increasing the temperature is the loss of the liquid state of helium, which worsens the thermal homogeneity of the coil.



(a) RDK-418D 4 K Cryocooler Series **(b)** RDK-408S 10K Cryocooler Series

Figure 57. Heat map capacity for two cryocoolers working at 50 Hz

6.3.2. Superconductor analysis

In this section, the superconductor alternatives for working at 10 K will be analyzed. The procedure will be similar to the developed in the previous section. First, the initial conditions will be evaluated. All the initial conditions will be kept the same except the initial temperature.

Parameter	*Value*
Energy	8.5 MeV
Cooling Temperature	10 K
Load factor	70 %
Nominal current	100 A
Steel	AISI 1010
Airgap	74 mm.
Iron-coil distance	32 mm
Aspect Ratio	1

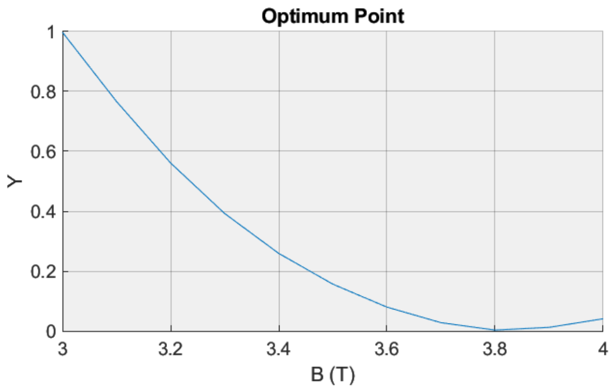
Table 17. Initial parameters for superconductor analysis at 10 K

Using the "Optimum Score" tab an initial score and characteristics of the different cyclotrons is obtained.

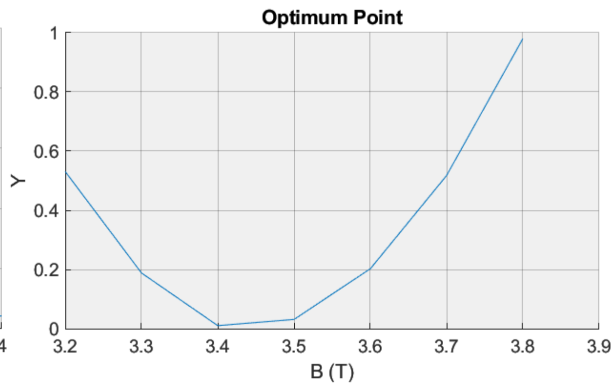
Material	*Score*	*Weight (kg.)*	*Price (€)*	*Center Field (T)*
NbTi	2	-	-	-
Nb_3Sn	8.45	1,375	41,354	3.4
MgB_2	5.02	1,630	104,929	2.9
HTS	6.2	655	58,420	6.8

Table 18. Initial parameters for superconductor analysis at 4.2 K

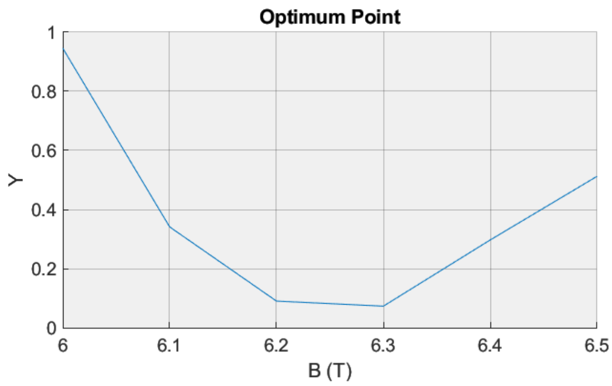
NbTi cannot operate at 10 K, which is the reason of the low score and the lack of defining characteristics. At first sight, Nb_3Sn seems to be the material better positioned, but it has to be confirmed.



(a) Nb_3Sn



(b) MgB_2



(c) HTS

Figure 58. Normalized weight versus field for different superconductors working at 10 K

With the new optimum values of the magnetic field, Table 13 is updated and the minimum weight cyclotrons for each material have the following characteristics.

Material	*Weight (kg.)*	*Diameter (m.)*	*Price (€)*	*Center Field (T)*
Nb_3Sn	1,193	0.77	32,863	3.8
MgB_2	1,342	0.83	79,590	3.45
HTS	694	0.58	65,703	6.2

Table 19. Updated Table, minimum weight cyclotron for each superconductor material working at 10 K

Once again, the selection is not immediate. MgB_2 offers similar characteristics to Nb_3Sn, although at double the price. An HTS cyclotron would have lower weight but at the cost of increasing the number of joints. Again, the "Score" tab was used to reach a final decision.

Material	*Score*
Nb_3Sn	7.87
MgB_2	4.49
HTS	5.56

Table 20. Final score for each superconductor working at 10 K

Nb_3Sn appears to be the superconductor better positioned to work at 10 K. HTS superconductors suffer from the aforementioned defects, it is still an immature technology with great possibilities but still some advances need to be done in wire fabrication and coil manufacturing, to use them in a greater scale. The analyzed MgB_2 wires does not seem to fit for this application, the required high fields discard them as viable options. The use of MgB_2 wires in the near future seems to be limited to low-field applications where it can be a good substitute of NbTi. For CYCLOMED it will be interesting to explore a development of an MgB_2 isochronous cyclotron working below 3 T.

6.3.3. Nb_3Sn cyclotron working at 10 K

Next, the Nb_3Sn solution will be explored more in detail. The initial parameters were exposed in Table 19. The magnet characteristics, including geometry, current and number of turns reached with the program are shown in Figure 59 and Table 21.

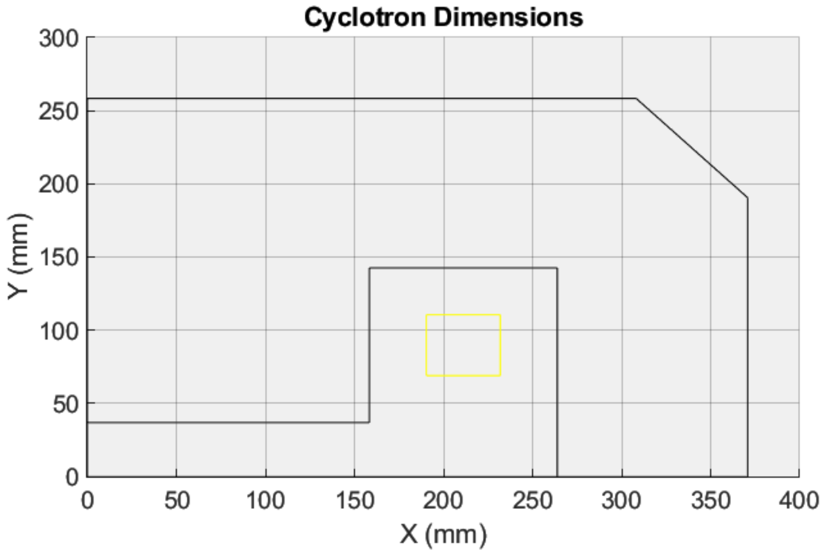


Figure 59. 2D view of the cyclotron geometry

These results were reached with a Supercon Nb_3Sn, with the characteristics shown in Table 21.

Parameter	*Value*
Nominal current	110.4 A
Number of turns	3876
Wire diameter (bare/insulated)	0.6/0.75 mm
Coil cross section	41.45x41.45 mm^2
Cu/nCu ratio	0.25
Distance iron to coil	32 mm

Table 21. Magnet parameters defined from 2D model

Once that the 2D design is completed, a 3D model was developed in SolidWorks and run in Maxwell. With the 3D simulation it is possible to corroborate the results obtained with the 2D model. An image of the model is presented in Figure 60.

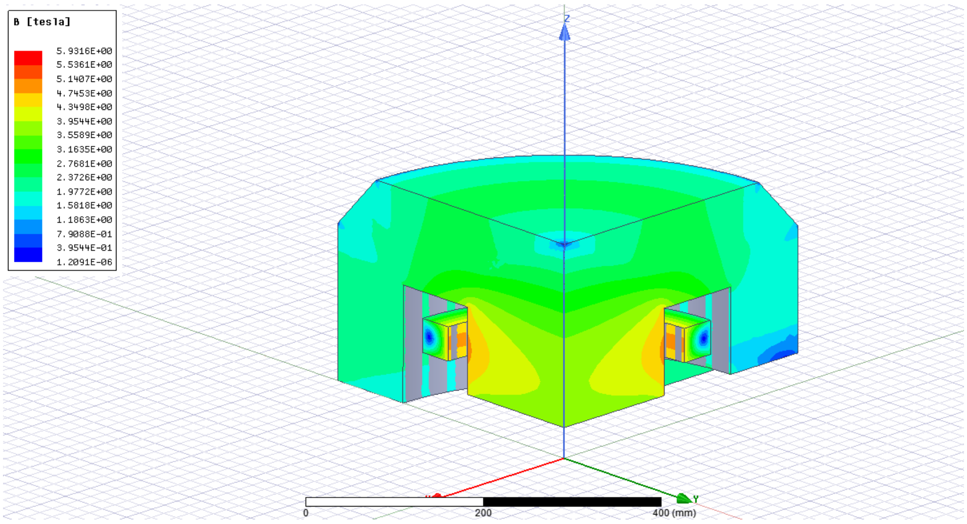
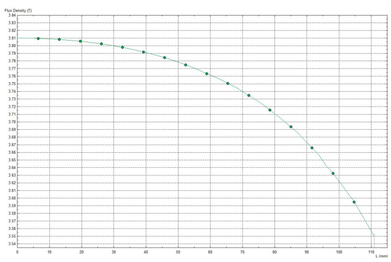
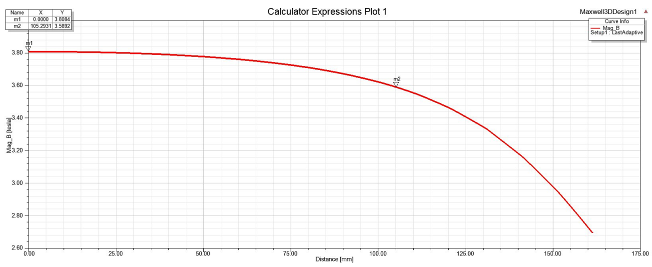


Figure 60. 3D view of the cyclotron, the magnetic field is represented in its surface

The procedure to run the cases in Maxwell is the same as the explained in section 5.5. To validate the solution reach with QF, the magnetic field at the midplane and the magnetic field in the superconductor are compared between the programs. The comparison is shown in Figure 61 and Figure 62, in this case the reader can appreciate the resemblance between both solutions.

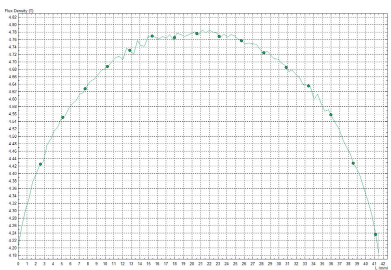


(a) Quickfield, center field of 3.81 T

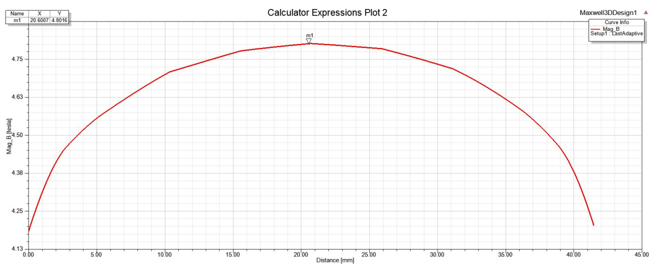


(b) Maxwell, center field of 3.81 T

Figure 61. Magnetic field shape at the mid-plane



(a) Quickfield, maximum field of 4.78 T



(b) Maxwell, maximum field of 4.8 T

Figure 62. Magnetic field shape in the superconductor

6.4. Design differences: operating at 4.2 K vs 10 K

Increasing the temperature from 4.2 K to 10 K has a clear impact in the design of the cyclotron. This change mainly affects two subsystems: the refrigeration system and the magnetic circuit.

The main change in the magnetic circuit is the use of another superconducting material, from NbTi to Nb_3Sn. This change won't have a great impact in the cyclotron design, as the optimum fields for both materials operating at the mentioned temperatures are extremely similar, 3.9 T and 3.8 T respectively. This implies that the size of the cyclotron would not change significantly, as it can be appreciated in Table 22.

On the other hand, the major change will come in the magnet fabrication. As reviewed in section 4.2, Nb_3Sn is very fragile when reacted, in contrast with NbTi, which complicates the winding process of the coil.

T_K	*Material*	*Center Field (T)*	*Weight (kg.)*	*Diameter (m.)*	*Price (€)*
4.2 K	NbTi	3.9	1,197	0.76	5,385
10 K	Nb_3Sn	3.8	1,193	0.77	32,863

Table 22. Magnetic circuit design differences for two different operating temperatures

The main beneficiary of this modification would be the refrigeration system. Operating at a higher temperature allows the system to extract a higher quantity of heat from the cyclotron. In subsection 6.3.1, it was shown that the heat extraction capacity of the most recent cryocooler operating at 10 K, triples the equivalent cryocooler operating at 4.2 K

T_K	*Heat load 2nd stage*
4.2 K	1.8 W
10 K	5.4 W

Table 23. Available heat for the second stage from two different commercial cryocoolers

6.5. Design effects of the initial inputs

In the previous sections the effects of changing different coil materials were explored, while maintaining the same conditions for the rest of the parameters. However, these parameters have a great effect in the final design and should not be chosen without much forethought. In the followings sections the effects on changing these parameters will be investigated.

6.5.1. Iron-yoke materials

Four types of steel for magnetic applications are included in the database shown in Table 24.

Steel	*Saturation Field (T)*
AISI 1010	2.056
AISI 1020	2.19
ARMCO	2.14
Hyperco 50	2.265

Table 24. Steel database, with the correspondent magnetic saturation field

Higher saturation fields implies that less Amperes-Turn are needed in order to obtain the same field, reducing the size of the coil. The coil reduction has a direct impact in the dimensions of the cyclotron.

Four simulations, one for each iron-yoke material were done with a 4 T NbTi cyclotron working at 4.2 K. The differences can be appreciated in Table 25.

Steel	*Weight (kg.)*	*Diameter (m.)*
AISI 1010	1198	0.76
AISI 1020	1105	0.75
ARMCO	1138	0.75
Hyperco 50	1058	0.74

Table 25. Cyclotron characteristics differences using a different steel

As it was expected, with a higher saturation steel a lighter cyclotron can be obtained. This weight reduction also impacts in the superconductor price.

The iron-yoke material election is not trivial, and its as important as the election the coil material. Steels with enhanced properties must be considered as an option for future developments.

6.5.2. Geometry factors: Airgap and iron-coil distance

Upon the resolution of the case the user has to define two geometric factors: the airgap and the iron-coil distance. The airgap as in most electric machines is one of the most important designing parameters but in this case is a variable dependent on the beam dynamics, so the user cannot modify it to his liking.

The iron-coil distance is kept to the minimum technologically feasible. In Table 26, it can be appreciated the effects of changing this distance 2 mm.

Iron-coil distance (mm.)	*Weight (kg.)*	*Diameter (m.)*	*Price (€)*
30	1149	0.75	5302
32	1198	0.76	5542
34	1248	0.77	5765

Table 26. Cyclotron characteristics varying iron-coil distance $\pm 2mm.$

Approximately, with a 1 mm variation a weight reduction/increase of 25 kg. can be achieved. Is important to maintain this distance at the minimum possible value, for the direct impact it has in the magnetic circuit design. Nevertheless, between the iron and the coil, the thermal shield, supporting structure, or the casing need to be accommodated. If the distance is too small, it is more probable that certain issues with the refrigeration system appear, so a balance between both systems must be reached.

6.5.3. Aspect ratio of the coil

Another geometry parameter that affects the design of the cyclotron is the aspect ratio of the coil. It is not easy to foresee which is the optimum value to minimize the cyclotron weight, for that several cases were run while maintaining the rest of the parameters constant (4T NbTi cyclotron).

Aspect ratio of the coil	*Weight (kg.)*	*Diameter (m.)*	*Price (€)*
0.5	1287	0.83	5309
0.75	1225	0.79	5429
1	1198	0.76	5542
1.5	1186	0.75	5825
2	1180	0.74	5815
2.5	1189	0.72	5924
3	1438	0.71	6146

Table 27. Cyclotron characteristics varying the aspect ratio of the coil

The optimum value of the aspect ratio appears to be around 2, although the difference in weight is not significant in either case. However, the diameter is reduced in a higher proportion than weight. Using a high aspect ratio could make the cyclotron more compact and suitable for space restricted rooms benefiting from a smaller diameter. Although it is out of the scope of this project, it is worth mentioning the importance of the aspect ratio in the mechanical design of the coil, as stresses and therefore the support structure will differ depending on the selected value.

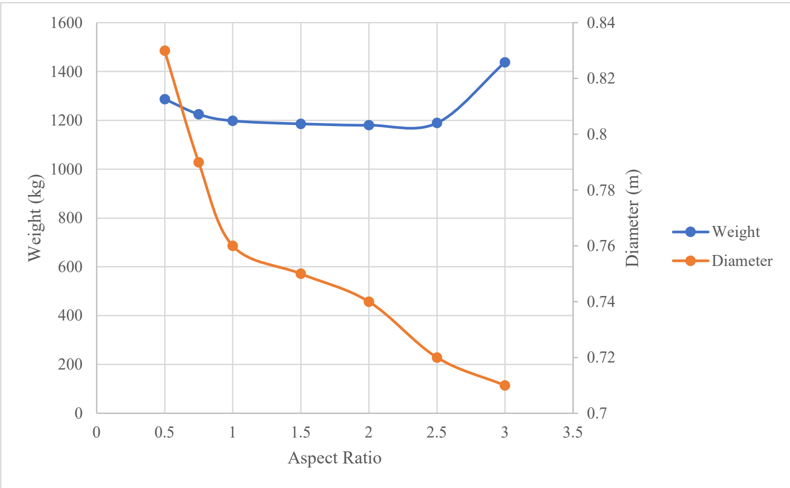


Figure 63. Weight (left axis) and diameter (right axis) versus the aspect ratio of the coil

Chapter 7

Conclusions

Superconducting materials manufacturing techniques have developed rapidly during the last 20 years. From a limited use, mostly in academic applications at the beginning of the century, to commercial applications that ranges from particle accelerators to magnetic trains or power applications.

CYCLOMED, as a company, must have a good oversight and be aware of the latest developments in the superconducting field, in order to maintain its competitive advantage and ensure its business success.

The objective of this thesis is aligned with this reasoning, exploring the possibilities CYCLOMED has in future developments regarding superconducting technology. For this purpose, a series of primary objectives were defined. The first objective was to review the current state of the art of commercially available superconductors and create a database with the properties of this superconducting materials, in terms of critical surface and price.

The second objective was to create a software program with which the main geometry and characteristics of a classical cyclotron could be obtained. The aim of this objective was to have a software tool to facilitate the last objective: doing and analysis and comparison of the impact on using alternative superconducting materials in the cyclotron geometry.

All these objectives have been successfully accomplished. Moreover, a series of enhanced objectives were defined during the realization of this thesis. The first three objectives are a natural continuation of the primary objectives. First the superconducting wire fabrication techniques and coil manufacture procedures were reviewed to give not only a quantitative impact, but a qualitative impact on the effects of using each superconductor in future developments. Second, a UI was developed to facilitate the analysis and comparison of the alternatives. In this UI the author included a method, in form of a score from 1 to 10, to evaluate each design. Moreover, this UI is easily upgradeable, and more materials could be included without making major modifications. In the development of the UI, it was included the possibility to modify not only the superconducting materials but also the yoke material and certain geometric parameters that have an impact on the magnetic design. The third enhanced objective was to perform an analysis of the impact of the variations of these inputs.

Two additional enhanced objectives were successfully conducted. In the first one, a method for modeling superconductors with ferromagnetic materials was implemented. The second objective was the validation of the developed program using a third party FEM software.

All these objectives are summarized in Table 28.

Primary Objectives	Enhanced Objectives
Creation of a database with the commercially available superconducting materials	Detailed analysis of available superconductors for magnet fabrication
Development of cyclotron sizing software	Elaboration of an intuitive software to facilitate the analysis to the user
Analysis and comparison on the impact of using different superconducting materials in the cyclotron geometry	Analysis and comparison of the effects of the varying the initial inputs
	Magnetic design with ferromagnetic materials in its matrix composition
	Validation with a 3D simulation software.

Table 28. Thesis objectives completed

CYCLOMED future possibilities in the design and development of future cyclotrons are multiple. If the option is to maintain the concept, certain improvements can be made in addition to the mentioned in [4], e.g., use of a better performing wire or modify the yoke material. If the strategy is more aggressive the decisions would mainly affect two variables, the operating temperature and the cyclotron energy. The first one was addressed in this thesis.

In the thesis the possibility of increasing the operating temperature to 10 K was explored. The available cooling power for the refrigeration system triples from 1.8 W to 5.4 W, while the magnetic circuit suffered certain changes. The main consequence is the use of Nb_3Sn superconductive coils. The working field would not suffer an extreme variation (from 4 T to 3.8 T), so there are not significant differences in terms of weight and size between the current solution and the proposed one. On the other hand, the coil fabrication would be much complex due to the fragility of reacted Nb_3Sn.

MgB_2 and specially HTS superconductors cannot be discarded as future possibilities as the manufacturing methods improve.

The selected temperature of 10 K seems reasonable and feasible, although further study needs to be done to evaluate the advantages and disadvantages of this option and select the optimum temperature for the system.

If the decision is to raise the energy (which has not been addressed in this project), to be capable of producing a higher quantity of radioisotopes, the classical cyclotron solution is not longer a possibility and it is necessary to go with an isochronous cyclotron.

These decisions are not exclusive to each other, and could be conducted at the same time. Further investigation and evaluation are needed to valuate these possibilities. Two line of actions are proposed for each one:

- Elaboration of a procedure to determine the optimum operating temperature, evaluating the advantages and disadvantages for each of the systems involved, the refrigeration system and magnetic circuit.
- Elaboration of an equivalent isochronous cyclotron design program, to obtain the geometric differences and analyze the possibility of developing such type of cyclotron.

Sustainable Development Goals

Population aging is a global phenomenon, the number of 60-year-olds will triple by 2050 [74]. This trend has been present in most developed countries during the 20th century, Japan life expectancy now tops the 82 years old. But the movement is emerging around the globe, in developing countries, the number of people aged >65 will triple from 2000 to 2030. In spite of the recent events and the unfortunate appearance of COVID-19 the world has experienced a transition in the leading causes of death, from infectious diseases to chronic and degenerative illness. Cardiovascular diseases, Alzheimer and other dementia are now the leading causes of death, according to WHO [2].

The Sustainable Development Goals (SDGs) are a compilation of 17 global objectives designed to be a "blueprint to achieve a better and more sustainable future for all" [34], developed by the United Nations and intended to be achieved by 2030. Addressing the aforementioned problem we have the SDG3, which promotes well-being and ensure a healthy live to all ages, for that new techniques allowing early diagnosis and giving better treatment must be developed. Radiopharmaceuticals can play a key role in both, diagnosis and treatment.

With the development and implantation of new compact cyclotrons the accessibility and availability to new forms of radiopharmaceuticals (^{18}F and ^{11}C), the diagnosis and treatment with a more personalized medicine will be improved.

The compactness of the solution facilitates the on-site production making this technology accessible to small cities or poor connected areas. This consequence is aligned with the SDG9, enhancing the technological and scientific capabilities of less developed areas.

Finally, the use of cyclotrons in a larger scale substituting nuclear reactors as the main producers of radioisotopes could greatly benefit the biosphere. Cyclotrons produced high specific activity radioisotopes in contrast with the radioisotopes produced in nuclear reactors, that significantly reduce the amount of nuclear waste.

SDG13, invites taking action against climate change and its impact. In [75] the author addresses the nuclear waste disposal problem and the environmental effects a wrong action could have. It becomes clear that the use of cyclotrons can help to address this problem and reduce the risks of a catastrophic event.

SDG Dimension	*SDG Identified*	*Role*	*Goal*
Biosphere	SDG 13: Take urgent action to combat climate change and its impacts	Secondary	Strengthen resilience and adaptive capacity to climate-related hazards and natural disasters in all countries
Society	SDG 3: Ensure healthy lives and promote well-being for all at all ages	Primary	Achieve universal health coverage, including financial risk protection, access to quality essential health-care services and access to safe, effective, quality, and affordable essential medicines and vaccines for all
Economy	SDG 9: Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation	Secondary	Enhance scientific research, upgrade the technological capabilities of industrial sectors in all countries, in particular developing countries, including, by 2030, encouraging innovation and substantially increasing the number of research and development workers per 1 million people and public and private research and development spending

Table 29. Sustainable development goals addressed in the paper

Bibliography

[1] United Nations. *World Population Ageing 2017 Highlights*. 2018, p. 42. DOI: https://doi.org/https://doi.org/10.18356/10e32e81-en. URL: https://www.un-ilibrary.org/content/publication/10e32e81-en.

[2] *World Health Organization*. URL: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[3] *"AMIT - Tecnologias de Imagen Molecular Avanzadas"*. eprint: http://www.upm.es/observatorio/vi/index.jsp?pageac=actividad.jsp&id_actividad=133556.

[4] Javier Munilla Lopez. "Development of a Novel Concept of Efficient Superconducting Magnet for Radioisotopes Production Cyclotron". PhD thesis. Universidad Pontificia de Comillas, 2019.

[5] *Mind the Gap*. Fundacion Botin. eprint: https://www.fundacionbotin.org/ciencia-contenidos/programa-inversion-en-transferencia-tecnologica-mind-the-gap.html.

[6] American Nuclear Society. *Medical Use of Radioisotopes*.

[7] Vasken Dilsizian. "SPECT and PET Techniques". In: *Atlas of Nuclear Cardiology*. London: Current Medicine Group, 2003, pp. 19–46. ISBN: 978-1-4615-6496-6. DOI: 10.1007/978-1-4615-6496-6_2. URL: https://doi.org/10.1007/978-1-4615-6496-6_2.

[8] M.-M. BE et al. *Table of Radionuclides*.

[9] Alberto Sardina del Dedo. "ACELERADORES COMPACTOS PARA LA PRODUCCION DE RADIOISOTOPOS". MA thesis. MINA, 2018.

[10] GM Currie et al. "Radionuclide production". In: *The Radiographer*. 2011.

[11] Timothy Antaya. “Cyclotron Basics”. In: *Unit 10 - Lectures 14*. MIT Plasma Science and Fusion Center.

[12] Frederic Chautard. “Beam dynamics for cyclotrons”. In: *GANIL and Caen and France*.

[13] Mike Seidel. “Introduction to Cyclotrons”. In: *WE-Heraeus-Seminar on accelerator physics for intense ion beams*. Paul Scherrer Institut. 2012.

[14] Siyong Kim and John W. Wong. *Advanced and emerging technologies in radiation oncology physics*. CRC Press, 2018.

[15] IAEA. *Cyclotron produced radionuclides: principles and practice*. Tech. rep. International atomic energy agency, 2008.

[16] Paul-Emmanuel Goethals. *Cyclotrons used in nuclear medicine report & directory*. Tech. rep. MEDrays Intell, 2020.

[17] Mateusz Adam Synowiecki, Lars Rutger Perk, and J. Frank W. Nijsen. “Production of novel diagnostic radionuclides in small medical cyclotrons”. In: *EJNMMI Radiopharmacy and Chemistry* 3.1 (Feb. 2018). DOI: 10.1186/s41181-018-0038-z.

[18] *GE Healthcare*. URL: http://www3.gehealthcare.es/es-es/productos/categorias/imagenes_moleculares/pet-radiofarmacia/tracer_center_equipment/minitrace_cyclotron#tabs/tab9A7DD68D95BA43D1B4513C

[19] M. ABS et al. “THE IBA PET- DEDICATED CYCLOTRONS MAIN FEATURES AND IMPROVEMENTS”. In: *Proceedings of the 14th International Conference on Cyclotrons and their Applications, Cape Town, South Africa*. 1348.

[20] *Design of IBA Cyclone 11 Cyclotron Magnet*. 2010.

[21] *Eclypse RD cyclotron*. Healthcare Siemens. URL: https://www.siemens-healthineers.com/en-uk/molecular-imaging/cyclotron-chemistry-solution/eclipse-rd-cyclotron/technical-details.

[22] *"TR-19"*. ACSI. URL: https://www.advancedcyclotron.com/cyclotron-solutions/pet/tr19.

[23] *Best Cyclotrons Systems*. URL: http://www.bestcyclotron.com/product_15p.html.

[24] *Sumitomo Heavy Industries*. URL: https://www.shi.co.jp/quantum/eng/product/pet/hm12.html..

[25] *ABT Molecular Imaging*. URL: http://www.bestabt.com/our-solutions/overview/.

[26] Bernd Seeber. *Handbook of Applied Superconductivity*. IOP Publishing Ltd, 1998.

[27] F.B. Silsbee and John Washington. In: *Academic Science* 6 (1916), p. 597.

[28] Alexander Gerbershagen et al. "The advantages and challenges of superconducting magnets in particle therapy". In: *Home Search Collections Journals About Contact us My IOPscience*. 2016.

[29] V. Smirnov and S. Vorozhtsov. "Modern compact accelerators of cyclotron type for medical applications". In: *Physics of Particles and Nuclei* 47.5 (Sept. 2016), pp. 863–883. DOI: 10.1134/s1063779616050051.

[30] Victor Smirnov et al. "Innovative 20-MeV Superconducting Cyclotron for Medical Applications". In: *Physics of Particles and Nuclei Letters* 16 (Nov. 2019), pp. 899–906. DOI: 10.1134/S1547477119060529.

[31] Malay Kanti Dey, Anjan Dutta Gupta, and Alok Chakrabarti. "Novel compact superconducting cyclotron for medical applications". In: *PHYSICAL REVIEW SPECIAL TOPICS - ACCELERATORS AND BEAMS* (2013). DOI: 10.1103/PhysRevSTAB.16.040101.

[32] *iMiTRACE Cyclotron*. iMiGine. URL: https://www.pmb-alcen.com/sites/pmb-alcen/files/pdf/data_sheet_iMiTRACE_eng_imp_02042020_1.pdf.

[33] Xiaoyu Wu. "The ION-12SC Compact Superconducting Cyclotron for Production of Medical Isotopes". In:

[34] *Quickfield Manual*. URL: http://quickfield.com/.

[35] *Ansys Maxwell*. URL: https://www.ansys.com/products/electronics/ansys-maxwell.

[36] Pedram Elhaminia et al. "Optimal Electromagnetic Design of a Nonsalient Magnetic-Cored Superconducting Synchronous Machine Using Genetic Algorithm". In: *IEEE Transactions on Applied Superconductivity* 25 (Feb. 2015), pp. 1–9. DOI: 10.1109/TASC.2014.2360874.

[37] D. Gavela et al. “Calculation and Design of a RF Cavity for a Novel Compact Superconducting Cyclotron for Radioisotope Production”. In: ().

[38] C. Oliver et al. “Optimizing the Radioisotope Production with a Weak Focusing Compact Cyclotron”. In: ().

[39] P. Calvo et al. “Optimization of the ion source for the compact superconducting AMIT cyclotron for radioisotope production - CIEMAT-Technology Department”. In: ().

[40] Hiroshi Ueda et al. “Conceptual Design of Compact HTS Cyclotron for RI Production”. In: *IEEE Transactions on Applied Superconductivity* 29.5 (Aug. 2019), pp. 1–5. DOI: 10.1109/tasc.2019.2903538.

[41] Tanvir Baig et al. “Conceptual designs of conduction cooled MgB and magnets for 1.5 and 3.0 T and full body MRI”. In: *Home Search Collections Journals*. 2017.

[42] Jacob Kelly and Hywel Owen. “Compact Rare Earth Superconducting Cyclotron”. In: *School of Physics and Astronomy*. 2019.

[43] *The National High Magnetic Field Laboratory*. URL: https://nationalmaglab.org/.

[44] Pia Jensen Ray. *Master’s thesis: Structural investigation of La(2-x)Sr(x)CuO(4+y) - Following staging as a function of temperature*. Feb. 2016. DOI: 10.6084/m9.figshare.2075680.v2. URL: https://figshare.com/articles/Structural_investigation_of_La_2_x_Sr_x_CuO_4_y_Following_staging_as_a_function_of_temperature/2075680/2.

[45] “Applied Superconductivity Handbook on Devices and Applications”. In: ed. by Paul Seidel. Wiley-VCH, 2015, pp. 105–193.

[46] Yukikazu Iwasa. “Case Studies in Superconducting Magnets Design and Operational Issues”. In: Springer. Chap. Chapter 3: Magnets, fields and forces, pp. 136–137.

[47] L. Bottura. *A PRACTICAL FIT FOR THE CRITICAL SURFACE OF NBTI*. Tech. rep. EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH, 1999.

[48] Martin N. Wilson. “SUPERCONDUCTING MATERIALS FOR MAGNETS”. In:

[49] Kyoji ZAITSU Kazuyoshi SAITO. “Nb3Sn Wire Technology for High-field Superconducting Magnet”. In: 2016.

[50] R. Gupta et al. “React Wind Nb3SnCommon Coil Dipole”. In: 2006.

[51] Luca Bottura and Bernardo Bordini. "Jc (B,T,eps) Parameterization for the ITER Nb3SN Production". In: *IEEE TRANSACTIONS ON APPLIED SUPERCONDUCTIVITY* (2009).

[52] Jack Ekin et al. "Extrapolative Scaling Expression: A Fitting Equationfor Extrapolating Full Ic(B,T,eps) Data Matrixes From Limited Data". In: *IEEE TRANSACTIONS ON APPLIED SUPERCONDUCTIVITY* (2017).

[53] DhalleMMJ. *Microsoft PowerPoint - Dhalle [Compatibility Mode].*

[54] L. Bottura L. Rossi. "EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCHCERN - ACCELERATORS AND TECHNOLOGY SECTOR". In: 2013.

[55] G Giunchi et al. "High performance new MgB2superconducting hollow wires". In: *Superconductor Science and Technology* 16.2 (Jan. 2003), pp. 285–291. DOI: 10.1088/0953-2048/16/2/328. URL: https://doi.org/10.1088%2F0953-2048%2F16%2F2%2F328.

[56] Mike Sumption et al. "Wind and React and React and Wind MgB2 Solenoid, Racetrack and Pancake Coils". In: *Applied Superconductivity, IEEE Transactions on* 17 (July 2007), pp. 2286–2290. DOI: 10.1109/TASC.2007.898039.

[57] Giunchi Giovanni. "High performance new MgB2 superconducting hollow wires". In: ().

[58] G. Li et al. "Transport Critical Current Densities and n -Values of Multifilamentary MgB_2 Wires at Various Temperatures and Magnetic Fields". In: *IEEE Transactions on Applied Superconductivity* 24.3 (2014), pp. 1–5.

[59] W.X. Li et al. "Dependence of superconducting properties on lattice strain in MgB2". In: *Physica C: Superconductivity and its Applications* 470 (2010). Proceedings of the 9th International Conference on Materials and Mechanisms of Superconductivity, S629–S630. ISSN: 0921-4534. DOI: https://doi.org/10.1016/j.physc.2009.11.053. URL: http://www.sciencedirect.com/science/article/pii/S0921453409007709.

[60] M. K. Wu et al. "Superconductivity at 93 K in a new mixed-phase Y-Ba-Cu-O compound system at ambient pressure". In: *Phys. Rev. Lett.* 58 (9 Mar. 1987), pp. 908–910. DOI: 10.1103/PhysRevLett.58.908. URL: https://link.aps.org/doi/10.1103/PhysRevLett.58.908.

[61] Timothy P. Beales et al. "Comparison of the powder-in-tube processing properties of two (Bi2-xPbx)Sr2Ca2Cu 3O10+Î´powders". In: *J. Mater. Chem.* 7 (4 1997), pp. 653–659. DOI: 10.1039/A606896K. URL: http://dx.doi.org/10.1039/A606896K.

[62] Yanwei Ma et al. "Fabrication and characterization of iron pnictide wires and bulk materials through the powder-in-tube method". In: *Physica C: Superconductivity* 469.9 (2009). Superconductivity in Iron-Pnictides, pp. 651–656. ISSN: 0921-4534. DOI: https://doi.org/10.1016/j.physc.2009.03.024. URL: http://www.sciencedirect.com/science/article/pii/S092145340900104X.

[63] Matt Rindfleisch. "Commercial Rules for Making Money with Superconducting Applications". In:

[64] *MATLAB R2019B*. URL: https://es.mathworks.com/products/matlab.html.

[65] *AISI 1010*. URL: https://www.azom.com/article.aspx?ArticleID=6539.

[66] *AISI 1020*. URL: http://www.matweb.com/search/datasheettext.aspx?matguid=10b74ebc27344380ab16b1b69f1cffbb.

[67] *ARMCO*. URL: https://www.aksteel.es/productos/armco-hierro-puro/.

[68] *Hyperco 50*. URL: https://edfagan.com/hiperco-50-coil.php.

[69] Hai-Chau Chang and Lih-Chung Wang. "A Simple Proof of Thue's Theorem on Circle Packing". In: (Sept. 2010).

[70] A. Stenvall et al. "Home Search Collections Journals About Contact us My IOPscience". In: *Superconductor science and technology* (2005).

[71] A. Stenvall et al. "Discrepancies in Modeling Magnets Utilizing MgB2 Conductor With Ferro- and Non-Magnetic Matrix Configurations". In: *IEEE Transactions on Applied Superconductivity* 17.2 (June 2007), pp. 2369–2372. DOI: 10.1109/tasc.2007.899269.

[72] *RDE-418D4 4K Cryocooler Series*. URL: http://www.shicryogenics.com/products/4k-cryocoolers/rde-418d4-4k-cryocooler-series/.

[73] *RDK-408S 10K Cryocooler Series*. URL: http://www.shicryogenics.com/products/10k-cryocoolers/rdk-408s-10k-cryocooler-series/.

[74] S A Tabish. "Population aging is a global phenomenon". In: Apr. 2012.

[75] Patsy T. Mink. "Nuclear Waste: The Most CompellingEnvironmental Issue Facing The World Today". In: *Fordham Environmental Law Review* (2011).

PART II

SOURCE CODE

Chapter 1

UI code

```
classdef ciclotronapp_v5_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                        matlab.ui.Figure
        TabGroup                        matlab.ui.container.TabGroup
        MaterialTab                     matlab.ui.container.Tab
        MaterialPanel_3                 matlab.ui.container.Panel
        SuperconductorDropDown_3Label   matlab.ui.control.Label
        SuperconductorDropDown_3        matlab.ui.control.DropDown
        UIAxes_3                        matlab.ui.control.UIAxes
        AddMaterialButton               matlab.ui.control.Button
        DataPanel_Material              matlab.ui.container.Panel
        BTEditField_2Label_3            matlab.ui.control.Label
        BT_Material                     matlab.ui.control.NumericEditField
        TKEditFieldLabel_2              matlab.ui.control.Label
        TK_Material                     matlab.ui.control.NumericEditField
        FillFactorLabel                 matlab.ui.control.Label
        Fill_Factor                     matlab.ui.control.NumericEditField
        JAm2Label_2                     matlab.ui.control.Label
        J_Material                      matlab.ui.control.NumericEditField
        MaterialComparisonTab           matlab.ui.container.Tab
        DataPanel                       matlab.ui.container.Panel
        BTEditField_2Label_2            matlab.ui.control.Label
        BTComparison                    matlab.ui.control.NumericEditField
        TKEditFieldLabel                matlab.ui.control.Label
        TKComparison                    matlab.ui.control.NumericEditField
        MinLengthmEditFieldLabel        matlab.ui.control.Label
        MinLength                       matlab.ui.control.NumericEditField
        WeightsPanel                    matlab.ui.container.Panel
        PerfPriceLabel                  matlab.ui.control.Label
```

```
Performance                       matlab.ui.control.NumericEditField
LenghtLabel                       matlab.ui.control.Label
Length                            matlab.ui.control.NumericEditField
ManufacturabilityEditFieldLabel_2  matlab.ui.control.Label
Manufacturability                 matlab.ui.control.NumericEditField
MaterialPanelComparison           matlab.ui.container.Panel
SuperconductorDropDownComparison  matlab.ui.control.DropDown
UIAxesComparison                  matlab.ui.control.UIAxes
ScoresPanel                       matlab.ui.container.Panel
ScoreNumber                       matlab.ui.control.NumericEditField
ScoreDropDown                     matlab.ui.control.DropDown
ScoreNumber_2                     matlab.ui.control.NumericEditField
RankingsPanel                     matlab.ui.container.Panel
Label                             matlab.ui.control.Label
Name1                             matlab.ui.control.EditField
Label_2                           matlab.ui.control.Label
Name2                             matlab.ui.control.EditField
Label_3                           matlab.ui.control.Label
Name3                             matlab.ui.control.EditField
Score1                            matlab.ui.control.NumericEditField
Score2                            matlab.ui.control.NumericEditField
Score3                            matlab.ui.control.NumericEditField
RunButtonComparison               matlab.ui.control.Button
OnlySuperconductorsCheckBox       matlab.ui.control.CheckBox
OnlyReliableDataCheckBox          matlab.ui.control.CheckBox
CyclotronTab                      matlab.ui.container.Tab
InitialDataPanel                  matlab.ui.container.Panel
EMeVLabel                         matlab.ui.control.Label
EMeVEditField                     matlab.ui.control.NumericEditField
BTLabel                           matlab.ui.control.Label
BTEditField                       matlab.ui.control.NumericEditField
WorkingPointEditFieldLabel        matlab.ui.control.Label
WorkingPointEditField             matlab.ui.control.NumericEditField
WireCurentAEditFieldLabel         matlab.ui.control.Label
WireCurentAEditField              matlab.ui.control.NumericEditField
GeometricDimensionsPanel          matlab.ui.container.Panel
AirgapmmEditFieldLabel            matlab.ui.control.Label
AirgapmmEditField                 matlab.ui.control.NumericEditField
CryostatmmEditFieldLabel          matlab.ui.control.Label
CryostatmmEditField               matlab.ui.control.NumericEditField
RunButton                         matlab.ui.control.Button
RunQFButton                       matlab.ui.control.Button
UIAxes                            matlab.ui.control.UIAxes
MaterialPanel                     matlab.ui.container.Panel
SteelDropDownLabel                matlab.ui.control.Label
```

```
SteelDropDown                      matlab.ui.control.DropDown
CopperSectionEditField_2Label      matlab.ui.control.Label
CopperSectionEditField             matlab.ui.control.NumericEditField
AspectRatioEditFieldLabel          matlab.ui.control.Label
AspectRatioEditField               matlab.ui.control.NumericEditField
CoolingTempEditFieldLabel          matlab.ui.control.Label
CoolingTempEditField               matlab.ui.control.NumericEditField
SuperconductorDropDown_4Label      matlab.ui.control.Label
SuperconductorDropDown             matlab.ui.control.DropDown
OutputdataPanel                    matlab.ui.container.Panel
WireDiammmEditFieldLabel           matlab.ui.control.Label
WireDiammmEditField                matlab.ui.control.NumericEditField
WireLenghtmEditFieldLabel          matlab.ui.control.Label
WireLenghtmEditField               matlab.ui.control.NumericEditField
WeightkgEditFieldLabel             matlab.ui.control.Label
WeightkgEditField                  matlab.ui.control.NumericEditField
PriceLabel                         matlab.ui.control.Label
PriceEditField                     matlab.ui.control.NumericEditField
WireAreamm2Label                   matlab.ui.control.Label
WireAreammEditField                matlab.ui.control.NumericEditField
DiametermLabel                     matlab.ui.control.Label
DiametermEditField_Custom          matlab.ui.control.NumericEditField
CyclotronScoreTab                  matlab.ui.container.Tab
InitialDataPanel_Comparison        matlab.ui.container.Panel
EMeVEditField_2Label_2             matlab.ui.control.Label
EMeVEditField_Comparison           matlab.ui.control.NumericEditField
BTEditField_3Label                 matlab.ui.control.Label
BTEditField_Comparison             matlab.ui.control.NumericEditField
WorkingPointEditField_2Label_2     matlab.ui.control.Label
WorkingPointEditField_Comparison  matlab.ui.control.NumericEditField
WireCurentAEditField_2Label        matlab.ui.control.Label
WireCurentAEditField_Comparison  matlab.ui.control.NumericEditField
CoolingTempEditField_2Label        matlab.ui.control.Label
CoolingTempEditField_Comparison  matlab.ui.control.NumericEditField
SteelDropDown_2Label_2             matlab.ui.control.Label
SteelDropDown_Comparison           matlab.ui.control.DropDown
GeometricDimensionsPanel_Comparison  matlab.ui.container.Panel
AirgapmmEditField_2Label_2         matlab.ui.control.Label
AirgapmmEditField_Comparison       matlab.ui.control.NumericEditField
CryostatmmEditField_2Label_2       matlab.ui.control.Label
CryostatmmEditField_Comparison     matlab.ui.control.NumericEditField
AspectRatioEditField_2Label_2      matlab.ui.control.Label
AspectRatioEditField_Comparison  matlab.ui.control.NumericEditField
RunButton_Comparison               matlab.ui.control.Button
RunQFButton_Comparison             matlab.ui.control.Button
```

```
RankingsPanel_Comparison        matlab.ui.container.Panel
Label_4                         matlab.ui.control.Label
Name1_Comparison                matlab.ui.control.EditField
Label_5                         matlab.ui.control.Label
Name2_Comparison                matlab.ui.control.EditField
Label_6                         matlab.ui.control.Label
Name3_Comparison                matlab.ui.control.EditField
Score1_Comparison               matlab.ui.control.NumericEditField
Score2_Comparison               matlab.ui.control.NumericEditField
Score3_Comparison               matlab.ui.control.NumericEditField
MaterialPanelComparison_2       matlab.ui.container.Panel
SuperconductorDropDownComparison_2  matlab.ui.control.DropDown
ScoresPanel_Comparison          matlab.ui.container.Panel
ScoreNumber_Comparison          matlab.ui.control.NumericEditField
ScoreDropDown_Comparison        matlab.ui.control.DropDown
ScoreNumber2_Comparison         matlab.ui.control.NumericEditField
OutputdataPanel_Comparison      matlab.ui.container.Panel
WireDiammmEditField_2Label      matlab.ui.control.Label
WireDiammmEditField_Comparison  matlab.ui.control.NumericEditField
WireLenghtmEditField_2Label     matlab.ui.control.Label
WireLenghtmEditField_Comparison  matlab.ui.control.NumericEditField
WeightkgEditField_2Label_2      matlab.ui.control.Label
WeightkgEditField_Comparison    matlab.ui.control.NumericEditField
PriceEditField_2Label           matlab.ui.control.Label
PriceEditField_Comparison       matlab.ui.control.NumericEditField
WireAreamm2Label_2              matlab.ui.control.Label
WireAreammEditField_Comparison  matlab.ui.control.NumericEditField
DiametermLabel_2                matlab.ui.control.Label
DiametermEditField_Custom_Comparison  matlab.ui.control.NumericEditField
WeightsPanel_Comparison         matlab.ui.container.Panel
WeightLabel                     matlab.ui.control.Label
Performance_Comparison          matlab.ui.control.NumericEditField
LenghtLabel_2                   matlab.ui.control.Label
Length_Comparison               matlab.ui.control.NumericEditField
ManufacturabilityEditFieldLabel_3  matlab.ui.control.Label
Manufacturability_Comparison    matlab.ui.control.NumericEditField
PriceLabel_3                    matlab.ui.control.Label
Price_Comparison                matlab.ui.control.NumericEditField
OnlySuperconductorsCheckBox_2   matlab.ui.control.CheckBox
OnlyReliableDataCheckBox_2      matlab.ui.control.CheckBox
OptimumCyclotronTab             matlab.ui.container.Tab
InitialDataPanel_Optimum        matlab.ui.container.Panel
EMeVEditField_2Label            matlab.ui.control.Label
EMeVEditField_Optimum           matlab.ui.control.NumericEditField
WorkingPointEditField_2Label    matlab.ui.control.Label
```

```
WorkingPointEditField_Optimum   matlab.ui.control.NumericEditField
GeometricDimensionsPanel_Optimum  matlab.ui.container.Panel
AirgapmmEditField_2Label        matlab.ui.control.Label
AirgapmmEditField_Optimum       matlab.ui.control.NumericEditField
CryostatmmEditField_2Label      matlab.ui.control.Label
CryostatmmEditField_Optimum     matlab.ui.control.NumericEditField
MaterialPanel_Optimum           matlab.ui.container.Panel
SteelDropDown_2Label            matlab.ui.control.Label
SteelDropDown_Optimum           matlab.ui.control.DropDown
CopperSectionEditField_2Label_2  matlab.ui.control.Label
CopperSectionEditField_Optimum  matlab.ui.control.NumericEditField
AspectRatioEditField_2Label     matlab.ui.control.Label
AspectRatioEditField_Optimum    matlab.ui.control.NumericEditField
CoolingTempEditField_3Label     matlab.ui.control.Label
CoolingTempEditField_Optimum    matlab.ui.control.NumericEditField
SuperconductorDropDown_4Label_2  matlab.ui.control.Label
SuperconductorDropDown_Optimum  matlab.ui.control.DropDown
FactorsPanel                    matlab.ui.container.Panel
WeightFactorEditFieldLabel      matlab.ui.control.Label
WeightFactorEditField           matlab.ui.control.NumericEditField
PriceFactorEditFieldLabel       matlab.ui.control.Label
PriceFactorEditField            matlab.ui.control.NumericEditField
UIAxes_Optimum                  matlab.ui.control.UIAxes
RunButton_Optimum               matlab.ui.control.Button
RunQFButton_Optimum             matlab.ui.control.Button
WeightkgEditField_2Label        matlab.ui.control.Label
WeightkgEditField_Optimum       matlab.ui.control.NumericEditField
PriceLabel_2                    matlab.ui.control.Label
PriceEditField_Optimum          matlab.ui.control.NumericEditField
BTEditField_2Label              matlab.ui.control.Label
BTEditField_Optimum             matlab.ui.control.NumericEditField
DiametermEditFieldLabel         matlab.ui.control.Label
DiametermEditField_Optimum      matlab.ui.control.NumericEditField
Zomm_Optimum                    matlab.ui.control.Button
OptimumScoreTab                 matlab.ui.container.Tab
InitialDataPanel_OptimumScore   matlab.ui.container.Panel
EMeVEditField_2Label_3          matlab.ui.control.Label
EMeVEditField_OptimumScore      matlab.ui.control.NumericEditField
WorkingPointEditField_2Label_3  matlab.ui.control.Label
WorkingPointEditField_OptimumScore  matlab.ui.control.NumericEditField
WireCurentAEditField_2Label_2   matlab.ui.control.Label
WireCurentAEditField_OptimumScore  matlab.ui.control.NumericEditField
CoolingTempEditField_2Label_2   matlab.ui.control.Label
CoolingTempEditField_OptimumScore  matlab.ui.control.NumericEditField
SteelDropDown_2Label_3          matlab.ui.control.Label
```

```
SteelDropDown_OptimumScore      matlab.ui.control.DropDown
GeometricDimensionsPanel_OptimumScore  matlab.ui.container.Panel
AirgapmmEditField_2Label_3      matlab.ui.control.Label
AirgapmmEditField_OptimumScore  matlab.ui.control.NumericEditField
CryostatmmEditField_2Label_3    matlab.ui.control.Label
CryostatmmEditField_OptimumScore  matlab.ui.control.NumericEditField
AspectRatioEditField_2Label_3   matlab.ui.control.Label
AspectRatioEditField_OptimumScore  matlab.ui.control.NumericEditField
RunButton_OptimumScore          matlab.ui.control.Button
RankingsPanel_OptimumScore      matlab.ui.container.Panel
Label_7                         matlab.ui.control.Label
Name1_OptimumScore              matlab.ui.control.EditField
Label_8                         matlab.ui.control.Label
Name2_OptimumScore              matlab.ui.control.EditField
Label_9                         matlab.ui.control.Label
Name3_OptimumScore              matlab.ui.control.EditField
Score1_OptimumScore             matlab.ui.control.NumericEditField
Score2_OptimumScore             matlab.ui.control.NumericEditField
Score3_OptimumScore             matlab.ui.control.NumericEditField
MaterialPanelOptimumScore       matlab.ui.container.Panel
SuperconductorDropDownOptimumScore  matlab.ui.control.DropDown
ScoresPanel_OptimumScore        matlab.ui.container.Panel
ScoreNumber_OptimumScore        matlab.ui.control.NumericEditField
ScoreDropDown_OptimumScore      matlab.ui.control.DropDown
ScoreNumber2_OptimumScore       matlab.ui.control.NumericEditField
WeightsPanel_OptimumScore       matlab.ui.container.Panel
WeightLabel_2                   matlab.ui.control.Label
Performance_OptimumScore        matlab.ui.control.NumericEditField
LenghtLabel_3                   matlab.ui.control.Label
Length_OptimumScore             matlab.ui.control.NumericEditField
ManufacturabilityEditFieldLabel_4  matlab.ui.control.Label
Manufacturability_OptimumScore  matlab.ui.control.NumericEditField
PriceLabel_4                    matlab.ui.control.Label
Price_OptimumScore              matlab.ui.control.NumericEditField
OnlySuperconductorsCheckBox_3   matlab.ui.control.CheckBox
OnlyReliableDataCheckBox_3      matlab.ui.control.CheckBox
OutputdataPanel_OptimumCyclotron  matlab.ui.container.Panel
BTLabel_2                       matlab.ui.control.Label
BTEditField_OptimumScore        matlab.ui.control.NumericEditField
WeightkgEditField_2Label_3      matlab.ui.control.Label
WeightkgEditField_OptimumCyclotron  matlab.ui.control.NumericEditField
PriceEditField_2Label_2         matlab.ui.control.Label
PriceEditField_OptimumCyclotron  matlab.ui.control.NumericEditField
DiametermLabel_3                matlab.ui.control.Label
DiametermEditField_OptimumCyclotron  matlab.ui.control.NumericEditField
```

```
ScoreTab                          matlab.ui.container.Tab
InitialDataPanel_Score            matlab.ui.container.Panel
EMeVEditField_2Label_4            matlab.ui.control.Label
EMeVEditField_Score               matlab.ui.control.NumericEditField
WorkingPointEditField_2Label_4    matlab.ui.control.Label
WorkingPointEditField_Score       matlab.ui.control.NumericEditField
WireCurentAEditField_2Label_3     matlab.ui.control.Label
WireCurentAEditField_Score        matlab.ui.control.NumericEditField
SteelDropDown_2Label_4            matlab.ui.control.Label
SteelDropDown_Score               matlab.ui.control.DropDown
GeometricDimensionsPanel_Score    matlab.ui.container.Panel
CryostatmmEditField_2Label_4      matlab.ui.control.Label
CryostatmmEditField_Score         matlab.ui.control.NumericEditField
AspectRatioEditField_2Label_4     matlab.ui.control.Label
AspectRatioEditField_Score        matlab.ui.control.NumericEditField
RunButton_Score                   matlab.ui.control.Button
MaterialPanelScore                matlab.ui.container.Panel
SuperconductorDropDown_Score_1    matlab.ui.control.DropDown
SuperconductorDropDown_Score_2    matlab.ui.control.DropDown
SuperconductorDropDown_Score_3    matlab.ui.control.DropDown
SuperconductorDropDown_Score_4    matlab.ui.control.DropDown
ScoresPanel_Score                 matlab.ui.container.Panel
ScoreNumber_Score_1               matlab.ui.control.NumericEditField
ScoreDropDown_Score_1             matlab.ui.control.DropDown
ScoreNumber2_Score_1              matlab.ui.control.NumericEditField
ScoreNumber_Score_2               matlab.ui.control.NumericEditField
ScoreNumber2_Score_2              matlab.ui.control.NumericEditField
ScoreNumber_Score_3               matlab.ui.control.NumericEditField
ScoreNumber2_Score_3              matlab.ui.control.NumericEditField
ScoreNumber_Score_4               matlab.ui.control.NumericEditField
ScoreNumber2_Score_4              matlab.ui.control.NumericEditField
WeightsPanel_Score                matlab.ui.container.Panel
WeightLabel_3                     matlab.ui.control.Label
Performance_Score                 matlab.ui.control.NumericEditField
LenghtLabel_4                     matlab.ui.control.Label
Length_Score                      matlab.ui.control.NumericEditField
ManufacturabilityEditFieldLabel_5  matlab.ui.control.Label
Manufacturability_Score           matlab.ui.control.NumericEditField
PriceLabel_5                      matlab.ui.control.Label
Price_Score                       matlab.ui.control.NumericEditField
OnlySuperconductorsCheckBox_Score  matlab.ui.control.CheckBox
OnlyReliableDataCheckBox_Score    matlab.ui.control.CheckBox
Parameters_Score                  matlab.ui.container.Panel
AirgapmmEditField_Score_1         matlab.ui.control.NumericEditField
CoolingTempEditField_Score_1      matlab.ui.control.NumericEditField
```

```
        TKLabel_Score                   matlab.ui.control.Label
        BTLabel__Score                  matlab.ui.control.Label
        AirgapmmLabel_Score             matlab.ui.control.Label
        BTEditField_Score_1             matlab.ui.control.NumericEditField
        AirgapmmEditField_Score_2       matlab.ui.control.NumericEditField
        CoolingTempEditField_Score_2    matlab.ui.control.NumericEditField
        BTEditField_Score_2             matlab.ui.control.NumericEditField
        AirgapmmEditField_Score_3       matlab.ui.control.NumericEditField
        CoolingTempEditField_Score_3    matlab.ui.control.NumericEditField
        BTEditField_Score_3             matlab.ui.control.NumericEditField
        AirgapmmEditField_Score_4       matlab.ui.control.NumericEditField
        CoolingTempEditField_Score_4    matlab.ui.control.NumericEditField
        BTEditField_Score_4             matlab.ui.control.NumericEditField
        RunQFButton_Score               matlab.ui.control.Button
    end


    properties (Access = private)
        E;              % Energy of the Cyclotron (MeV)
        Bo;             % Magnetic Field in the center (T)
        Bos;            % Magnetic Field in non-magnetic medium (T)
        Bsc;            % Magnetic Field in the coil (T)
        Supercond;      % Type of Superconductor
        Supercond_name;% Name of Superconductor
        Steel;          % Type of steel
        Tk;             % Working Temperature (K)
        Cu_SC;          % Copper-Section Relation
        Jc;             % A/m2 in the coil (A/m2)
        Jeq;            % A/m2 in the coil (A/m2)
        NI;             % A/rev (A/rev)
        Muo;            % Magnetic Permeability in vacuum
        V;              % Volume of the superconductor (m3)
        P;              % Price of the Superconductor (ÿ)
        p_unit;         % Price of the Superconductor (ÿ/m3)
        W;              % Weight of the magnetic circuit (kg)
        rho_fe;         % Steel density (kg/m3)
        rho_sc;         % Superconductor density (kg/m3)
        steel_data;     % Variable that contains steel properties
        WF;             % Weight Factor
        PF;             % Price Factor
        Curve;          % B-T-J surface of the superconductor
        T;              % Range of working temperatures from superconductor
        B;              % Range of the available magnetic field
        wire_diameter;  % Diameter of the superconductor (mm)
        wire_lenght;    % meters of superconductor
```

```
wire_area;      % Wire area (mm2)
Ff;             % Fill factor
Frf;            % RadioFrequency factor
I;              % Current through each wire
dat;            % text file with superconductor information
dat_steel;      % text file with steel information

% Geometry Data

re;             % Extraction Radius (mm)
rp;             % Pole Radius (mm)
y;              % Pole Radius (m)
t;              % Coil Width (mm)
L;              % Coil Height (mm)
g;              % Thickness of the Air Gap (mm)
c;              % Thickness of the Cryostat (mm)
ri;             % Internal Radius Steel Yoke (mm)
ro;             % External Radius Steel Yoke (mm)
h;              % Height Steel Yoke (mm)
w;              % Core Width (mm)
z;              % Height of the coil (mm)
p;              % Height of the pole (mm)
ch_v;           % Vertical distance to chamfer (mm)
ch_h;           % Horizontal distance to chamfer (mm)

% Initial Assumptions

WP;             % Working Point
Bc;             % Magnetic Saturation Field (T)
AR;             % Aspect Ratio of the Coil (mm)
Rep;            % Relation between extraction radius and pole radius
Rosc;           % Relation between field in the center with field in the coil
alpha;          % Correction parameter to calculate the field
ext;            % Distance from the ciclotron to the 0 magnetic potential edge

% QF data

answer;         % QF problem name
QF;             % QF program

% Material comparison

% Scores
score_perf_price;
score_length;
```

```
        score_manufacturability;
        score_total;
        score_weight;
        score_price;

    end

    methods (Access = public)

        function DetermineGeometry(app)
            % Function: Determine the geometry of the cyclotron

            % Read steel data from .csv file
            app.steel_data = csvread('Acero.csv',3,0);

            % Saturation Field

            % Length of the column (Number of points in the B-H curve)
            a = size(app.steel_data,1);

            app.Bc = 0;
            % Saturation field
            % Approximate with a straight line y = mx + n, being n = Bc, m = Muo, and x
            % and y a point of the B-H curve
            while app.Bc == 0
                app.Bc = app.steel_data(a,3*(app.Steel - 1) + 2) -
                    app.Muo*app.steel_data(a,3*(app.Steel - 1) + 3);
                a = a - 1;
            end

            % Extraction Radius
            app.re = 1000*(app.E/(47.89*app.Bo^2))^0.5; %Change from m to mm

            % Pole Radius
            app.rp = app.re/app.Rep;
            app.y = app.rp/1000;

            % Field in non-magnetic medium
            app.Bos = app.Bo - app.Bc;
            if app.Bos < 0
                app.Bos = 0;
            end

            % Maximum Field in the coil
            app.Bsc = app.Rosc*((2/app.Bo)^0.2)*app.Bo;
```

```
    % A/m2 in the coil
    % Select Temperature
    % In case we only have data of one Temp
    if app.T(1) == app.T(end)
        app.Jc = (app.Curve(app.Bsc))/(app.Cu_SC + 1);
    else
        app.Jc = (app.Curve(app.Bsc,app.Tk))/(app.Cu_SC + 1);
    end



    % Position of the coil
    z_c = 0;

    % In case Bc is too high for the field
    if app.Jc <= 0
        msgbox('Material not valid for that field');
        return;
    end
    % A/rev in the coil
    NI_c = (app.g*app.Bc/(1000*app.Muo) + 1.2*app.Bos*app.rp/(1000*app.Muo));
        %Divided by 1000 to convert mm to m

    % Calculate the basic dimensions with this supposition
    Geometry1(app,NI_c,z_c);

    % Checks the validity of the suposition (value of z_c)
    % To check this suposition check Bo = Be, field at the center equals the
    % field at the extraction radius
    CheckGeometry(app);

    % Calculate the rest of the dimensions
    Geometry2(app);

end
function [fitresult,y,x] = fitCurve(app)
    % %FitCurve()
    %  Create a fit curve/surface for each material.
    %
    %  Data for fit:
    %      Supercond: Type of superconductor
    %  Output:
    %      fitresult : a fit object representing the fit.
    %      y : range of temperatures [K].
```

```
    %% Fit surface to superconductor data

    if strcmp(app.Supercond_name,'Copper')
        x = 0:0.1:15;
        y = 0:0.1:100;
        z = 3000000*ones(size(y,2),size(x,2));
        row = 2;
    else
        % Read superconductor properties
        dat = csvread(strcat(app.Supercond_name,'.csv'));

        % Magnetic field range (T)
        x = dat(1,2:size(dat,2));

        % Temperature Range (K)
        y = dat(2:size(dat,1),1)';

        % Current density (J/mm^2)
        z = zeros(size(y,2),size(x,2));

        for row = 1:size(y,2)
            for col = 1:size(x,2)
                val = dat(row+1,col+1);
                if val == 0
                    val = NaN;
                end
                z(row,col) = val;
            end
        end
    end



    % Create Surface or Curve depending on the data
    if row > 1
        [fitresult, gof] = createSurfaceFit(app,x, y, z);
    else
        [fitresult,gof] = createCurveFit(app,x,z);
    end
end
function [fitresult, gof] = createCurveFit(app,x, z)
    %CREATEFIT(X_OUT,Y_OUT,Z_OUT)
    %  Create a fit.
    %
    %  Data for fit:
    %      X Input : x_out
```

```
            %      Y Input : y_out
            %  Output:
            %      fitresult : a fit object representing the fit.
            %      gof : structure with goodness-of fit info.
            %
            %  See also FIT, CFIT, SFIT.

            %% Fit: 'B-T-J curve'.
            [xData, zData] = prepareCurveData(x,z);

            % Set up fittype and options.
            ft = fittype( 'poly2' );
            opts = fitoptions( 'Method', 'LinearLeastSquares' );
            opts.Robust = 'Off';

            % Fit model to data.
            [fitresult,gof] = fit(xData,zData,ft, opts);

%             % Plot fit with data.
%             figure( 'Name', 'Surface Response' );
%             h = plot( fitresult, xData, zData );
%             legend(h, 'B-T-J curve', 'J vs. B, T', 'Location', 'NorthEast', 'Interpreter',
    'none' );
%             % Label axes
%             xlabel( 'B (T)', 'Interpreter', 'none' );
%             ylabel( 'J (A/m2)', 'Interpreter', 'none' );
%             grid on
        end
        function [fitresult, gof] = createSurfaceFit(app,x, y, z)
            %CREATESURFACEFIT(X_OUT,Y_OUT,Z_OUT)
            %  Create a fit.
            %
            %  Data for fit:
            %      X Input : x_out
            %      Y Input : y_out
            %      Z Output: z_out
            %  Output:
            %      fitresult : a fit object representing the fit.
            %      gof : structure with goodness-of fit info.
            %
            %  See also FIT, CFIT, SFIT.


            %% Fit: 'B-T-J curve'.
            [xData, yData, zData] = prepareSurfaceData(x,y,z);
```

```
%             % Set up fittype and options.
%             ft = fittype( 'poly32' );
%             opts = fitoptions( 'Method', 'LinearLeastSquares' );
%             opts.Robust = 'Off';
%
%             % Fit model to data.
%             [fitresult, gof] = fit( [xData, yData], zData, ft, opts );
            ft = 'cubicinterp';

            % Fit model to data.
            [fitresult, gof] = fit( [xData, yData], zData, ft, 'Normalize', 'on' );

            [X,Y] = meshgrid(x(1):0.1:xData(end),y(1):0.1:y(end));
            Z = fitresult(X,Y);

            % Plot fit with data.
            surf(app.UIAxes_3,X,Y,Z,'EdgeColor',"texturemap");
            zlim(app.UIAxes_3, [0,inf]);
            legend( app.UIAxes_3, 'B-T-J curve', 'J vs. B, T', 'Location', 'NorthEast',
                'Interpreter', 'none' );
            % Label axes
            xlabel( app.UIAxes_3,'B (T)', 'Interpreter', 'none' );
            ylabel( app.UIAxes_3,'T (K)', 'Interpreter', 'none' );
            zlabel( app.UIAxes_3,'J (A/m2)', 'Interpreter', 'none' );
            view(app.UIAxes_3, 120, 50 );
            grid(app.UIAxes_3,"on");

            % PLot data in Axes 4
            surf(app.UIAxesComparison,X,Y,Z,'EdgeColor',"texturemap");
            zlim(app.UIAxesComparison, [0,inf]);
            legend( app.UIAxesComparison, 'B-T-J curve', 'J vs. B, T', 'Location',
                'NorthWest', 'Interpreter', 'none' );
            % Label axes
            xlabel( app.UIAxesComparison,'B (T)', 'Interpreter', 'none' );
            ylabel( app.UIAxesComparison,'T (K)', 'Interpreter', 'none' );
            zlabel( app.UIAxesComparison,'J (A/m2)', 'Interpreter', 'none' );
            view(app.UIAxesComparison, 120, 50 );
            grid(app.UIAxesComparison,"on");

        end
        function Geometry1(app,NI_c,z_c)
            % Function that computes some geometric parameters of the
            % cyclotron
            % A/rev in the coil
```

```
        app.NI = NI_c;

        % Current density in the coil
        app.Jeq = app.Jc*app.WP*app.Ff;

        % Width of the coil
        app.t = 1000*((app.NI)/(app.AR*app.Jeq))^0.5;

        % Height of the coil
        app.L = app.t*app.AR;

        % Interior radius of the yoke
        app.ri = app.rp + app.t + 2*app.c;

        % Exterior radius of the yoke
        app.ro = ((app.Bo*app.re^2)/(2*app.Bc*app.Frf) + (app.Bo*(app.rp + app.c +
            app.t)^2)/(2*app.Bc*app.Frf) + app.ri^2)^0.5;

        % Steel Yoke height
        app.h = (app.re^2)*app.Bo/(2*app.ri*app.Bc) + app.Bo*((app.rp + app.c + app.t)^2
            -app.re^2)/(4*app.Bc*app.ri);

        % Core width
        app.w = 2*app.ro;

        % Height of the coil
        app.z = z_c*1000;

    end
    function CheckGeometry(app)
        % Check the validity of z_c supposition
        % Classical condition of cyclotrons
        % Be = Bo
        % Field at the extraction radius equals the field at the center

        % Specified time limit
        % 10 seconds to solve the case analytically
        % If not pass
        time0 = tic;
        timeLimit = 10;

        z_c = app.z/1000;
        % Parameter of the iteration. How much z increases/decreases
        e = 0.001;
        i = 0;
```

```
j = 0;
% k = Be - Bo
k = GetK(app,z_c);

while 1
    while k ~= 0

        if (k < 0)
            z_c = z_c - e;
            e = e/2;
        elseif (k  > 0)
            z_c = z_c + e;
        end

        % Recalculate geometry with new value of z_c
        Geometry1(app, app.NI,z_c);
        i = i + 1;
        if z_c < (app.g/2 + app.L/2 + app.c)/1000
            z_c = (app.g/2 + app.L/2 + app.c)/1000;
            break;
        end

        if toc(time0) > 5
            break;
        end
        % Check k again
        k = GetK(app,z_c);

    end
    j = j + 1;
    a = (app.rp + app.t/2 + app.c)/1000;

    % Recalculate NI necessary to obtain Bo T in the center
    NI_ol = app.g*app.Bc/(1000*app.Muo);
    NI_os = app.Bos*((z_c^2 + a^2)^(1.5))/(app.Muo*(a^2));
    NI_c = (NI_ol + NI_os);

    % Recalculate geometry with new value of z_c
    Geometry1(app,NI_c,z_c);

    if z_c <= (app.g/2 + app.L/2 + app.c)/1000 && (i > 20 || j > 20)
        z_c = (app.g/2 + app.L/2 + app.c)/1000;
        break;
    end
```

```
        % If timelimit is passed break
        if toc(time0) > timeLimit
            msgbox('Unable to solve case analytically');
            break
        end

        % Check k again
        k = GetK(app,z_c);
        if k == 0 && z_c >= (app.g/2 + app.L/2 + app.c)/1000
            break;
        end
    end
    alpha1 = (1.111 - 0.01674*app.Bo + 0.1996*app.AR - 0.06951*app.AR^2 -
        0.003248*app.Bo*app.AR);
    app.NI = app.NI/alpha1;

    Geometry1(app,app.NI,z_c);
end
function k = GetK(app,z_c)
    % Function to determine k
    % k = Be - Bo
    % Convert to m the measures that defines the position of the coil
    a = (app.rp + app.t/2 + app.c)/1000;

    % Calculate Be, which is the magnetic field at a distance y from the center
    % Need to solve the eliptic integral:
        http://www.sc.ehu.es/sbweb/fisica/elecmagnet/campo_magnetico/espira/espira.html
    m = (4*a*app.y)/((a+app.y)^2+z_c^2);

    [K, E]= ellipke(m);
    alpha = (1.958 - 0.3378*app.Bo + 0.03548*app.AR + 0.03203*app.Bo^2 -
        0.01369*app.Bo*app.AR);



    Be = (2*app.Muo*app.NI*a*((2*m)^0.5)*(a*m*E/(2-2*m) + app.y*K -
        app.y*(2-m)*E/(2-2*m))/(2*pi*(2*a*app.y)^(1.5)) + app.Bc)/alpha;

    k = round(Be - app.Bo,2);
end
function Geometry2(app)
    % Determines some geometric parameters of the cyclotron
    % Height of the pole
    app.p = app.z + app.L/2 + app.c - app.g/2;
```

```
        % Vertical distance to chamfer (mm)
        app.ch_v = app.h*((2^0.5)-1)+ app.p + app.g/2;

        % Horizontal distance to chamfer (mm)
        app.ch_h = (app.ro-app.ri)*((2^0.5)-1) + app.ri;

        % Superconductor Volume
        app.V = 2*app.t*app.L*pi*(app.ri + app.rp)/(1000^3);

        % Superconductor Price
        app.P = app.V*app.Ff*app.p_unit;

        % Weight of the magnetic circuit
        app.rho_fe = app.steel_data(1,3*(app.Steel - 1) + 1);       % (kg/m3)

        app.W = 2*(app.rho_fe*(pi*(app.ro^2)*(app.g/2 + app.p + app.h) -
            pi*(app.ri^2)*app.g/2 - pi*(app.ri^2 - app.rp^2)*app.p) +
            app.rho_sc*app.t*app.L*(app.ri + app.rp)*pi-2*app.rho_fe*pi*(2*app.ro/3 +
            app.ch_h)*(app.ro - app.ch_h)*(app.g/2 + app.p + app.h - app.ch_v))/(10^9);

        % Wire properties
        app.wire_diameter = 2*(app.I*1000000/(app.Jc*app.WP*pi))^0.5;
        % I = Jc*WP*pi*((0.896/2)^2)/1000000;
        app.wire_area = app.I*1000000/(app.Jc*app.WP);
        N = app.NI/app.I;
        % N_1 = (t*L)*Ff/(pi*((0.896/2)^2));
        app.wire_lenght = 2*2*N*pi*(app.rp + app.t/2 + app.c)/1000;
    end
    function PlotCyclotron(app)
        % Function that plots the cyclotron
        % Plot steel yoke
        plot(app.UIAxes,[0; app.ro],[0; 0],'-k');
        hold (app.UIAxes);
        plot(app.UIAxes,[0; 0],[0; app.g/2 + app.p + app.h],'-k');
        plot(app.UIAxes,[app.ro; app.ro],[0; app.ch_v],'-k');
        plot(app.UIAxes,[0; app.ch_h],[app.g/2 + app.p + app.h; app.g/2 + app.p +
            app.h],'-k');
        plot(app.UIAxes,[app.ch_h; app.ro],[app.g/2 + app.p + app.h; app.ch_v],'-k');

        % Plot Airgap
        plot(app.UIAxes,[0; app.rp],[0; 0],'-k');
        plot(app.UIAxes,[0; 0],[0; app.g/2],'-k');
        plot(app.UIAxes,[0; app.rp],[app.g/2; app.g/2],'-k');

        % Plot the cryostat
```

```
            plot(app.UIAxes,[app.rp; app.ri],[0; 0],'-k');
            plot(app.UIAxes,[app.rp; app.rp],[app.g/2; app.g/2 + app.p],'-k');
            plot(app.UIAxes,[app.rp; app.ri],[app.g/2 + app.p; app.g/2 + app.p],'-k');
            plot(app.UIAxes,[app.ri; app.ri],[0; app.g/2 + app.p],'-k');

            % Plot the superconductor
            plot(app.UIAxes,[app.rp + app.c; app.rp + app.c + app.t],[app.z - app.L/2; app.z
                - app.L/2],'-y');
            plot(app.UIAxes,[app.rp + app.c; app.rp + app.c],[app.z - app.L/2; app.z +
                app.L/2],'-y');
            plot(app.UIAxes,[app.rp + app.c; app.rp + app.c + app.t],[app.z + app.L/2; app.z
                + app.L/2],'-y');
            plot(app.UIAxes,[app.rp + app.c + app.t; app.rp + app.c + app.t],[app.z -
                app.L/2; app.z + app.L/2],'-y');
            hold (app.UIAxes,"off");
        end
        function prb = RunQF(app)
            %% Function that runs QF
            %% 1- Open Quickfield
%             try
%                 % Close QF if open and it opens it again
%                 app.QF.Quit (); % Finish QuickField program
%                 app.QF.release;
%                 try
%                     app.QF = actxserver ('QuickField.Application');
%                 catch
%                     error ('QuickField cannot start');
%                 end
%                 count = 1;
%             catch
%                 % Opens QF
%                 try
%                     app.QF = actxserver ('QuickField.Application');
%                 catch
%                     error ('QuickField cannot start');
%                 end
%                 count = 0;
%             end
            try
                app.QF = actxserver ('QuickField.Application');
            catch
                error ('QuickField cannot start');
            end

            app.QF.MainWindow.Visible = true;
```

```
    % make the QuickField main window visible
    % display the welcome message

    fprintf ('\n QuickField %s started', app.QF.Version);
    %% 2- Create/Open Problem
    % If QF is open overwrite the archive
    if isempty(app.answer)
        prb = CreateProblem(app);
    else
        prb = OpenProblem(app);
    end

    %% 3- Build Geometry

    BuildGeometry (app,prb.get ('ReferencedFile', 0)); % Build the geometry model
        with FE mesh

    %% 4- Set Properties

    SetData (app,prb); % Set media properties and boundary conditions
    %% 5- Solve Problem

    Solve (app,prb); % Run solver

    %% 7- View results

    ViewResults (app,prb); % Show the field picture
end
function prb = CreateProblem(app)
    % Creates a problem

    prompt = {'Under what name would you want to save the file?'};
    dlgtitle = 'File name';
    definput = {'ciclotronv1'};
    opts.Interpreter = 'None';
    app.answer = inputdlg(prompt,dlgtitle,[1 60],definput,opts);

    app.QF.DefaultFilePath = fullfile(strcat(pwd,'\QF'));

    prb = app.QF.Problems.Add (); % Create an empty problem
    prb.ProblemType = 'qfMagnetostatics'; % Set the analysis type
    prb.Class = 'qfAxisymmetric'; % Set the geometry class
    prb.LengthUnits = 'qfMillimeters'; % Set length units
    prb.Coordinates = 'qfCartesian'; % Set coordinate system
```

```
    % Set geometry and physical data files the problem refers to
    prb.set ('ReferencedFile', 0, string(strcat(app.answer,'.mod')));
    prb.set ('ReferencedFile', 1, string(strcat(app.answer,'.dms')));
    prb.SaveAs (string(strcat(app.answer,'.pbm'))); % Save the new problem
end
function prb = OpenProblem(app)
    % Opens a problem
    app.QF.DefaultFilePath = fullfile(strcat(pwd,'\QF'));
    prb = app.QF.Problems.Open (string(strcat(app.answer,'.pbm')));
end
function BuildGeometry(app, modelFileName)
    mdl = app.QF.Models.Add (); % Create an empty geometry model
    mdl.SaveAs (modelFileName); % Save it immediately to establish link to the problem


    % Create the Air
    shp = mdl.Shapes.AddEdge (app.QF.PointXY(0, 0), app.QF.PointXY(app.ext*(app.g/2 +
        app.p + app.h), 0));
    mdl.Shapes.AddEdge (app.QF.PointXY (0, 0), app.QF.PointXY(0, app.ext*app.ro));
    mdl.Shapes.AddEdge (app.QF.PointXY (app.ext*(app.g/2 + app.p + app.h), 0),
        app.QF.PointXY(app.ext*(app.g/2 + app.p + app.h), app.ext*app.ro));
    mdl.Shapes.AddEdge (app.QF.PointXY (0, app.ext*app.ro),
        app.QF.PointXY(app.ext*(app.g/2 + app.p + app.h), app.ext*app.ro));
    shp.Left.Item (1).Label = 'Air';

    % Create the Steel Yoke
    shp = mdl.Shapes.AddEdge (app.QF.PointXY(0, 0), app.QF.PointXY(0, app.ro));
    mdl.Shapes.AddEdge (app.QF.PointXY(0, 0), app.QF.PointXY(app.g/2 + app.p + app.h,
        0));
    mdl.Shapes.AddEdge (app.QF.PointXY (0, app.ro), app.QF.PointXY(app.ch_v, app.ro));
    mdl.Shapes.AddEdge (app.QF.PointXY (app.g/2 + app.p + app.h, 0),
        app.QF.PointXY(app.g/2 + app.p + app.h, app.ch_h));
    mdl.Shapes.AddEdge (app.QF.PointXY (app.ch_v, app.ro), app.QF.PointXY(app.g/2 +
        app.p + app.h, app.ch_h));
    shp.Right.Item (1).Label = 'Steel';

    % Create the Space for RF
    shp = mdl.Shapes.AddEdge (app.QF.PointXY(0, app.ri), app.QF.PointXY(0, app.ro));
    mdl.Shapes.AddEdge (app.QF.PointXY(0, app.ri), app.QF.PointXY(app.g/2, app.ri));
    mdl.Shapes.AddEdge (app.QF.PointXY (0, app.ro), app.QF.PointXY(app.g/2, app.ro));
    mdl.Shapes.AddEdge (app.QF.PointXY (app.g/2, app.ri), app.QF.PointXY(app.g/2,
        app.ro));
    shp.Right.Item (1).Label = 'Steel_air';

    % Create the Superconductor
```

```
shp = mdl.Shapes.AddEdge (app.QF.PointXY(app.z - app.L/2, app.rp + app.c),
    app.QF.PointXY(app.z + app.L/2, app.rp + app.c));
mdl.Shapes.AddEdge (app.QF.PointXY(app.z - app.L/2, app.rp + app.c),
    app.QF.PointXY(app.z - app.L/2, app.rp + app.c + app.t));
mdl.Shapes.AddEdge (app.QF.PointXY (app.z + app.L/2, app.rp + app.c),
    app.QF.PointXY(app.z + app.L/2, app.rp + app.c + app.t));
mdl.Shapes.AddEdge (app.QF.PointXY (app.z - app.L/2, app.rp + app.c + app.t),
    app.QF.PointXY(app.z + app.L/2, app.rp + app.c + app.t));
shp.Left.Item (1).Label = 'Superconductor';

% Create the Cryostat
shp = mdl.Shapes.AddEdge (app.QF.PointXY(0, app.rp), app.QF.PointXY(app.g/2 +
    app.p, app.rp));
mdl.Shapes.AddEdge (app.QF.PointXY(0, app.rp), app.QF.PointXY(0, app.ri));
mdl.Shapes.AddEdge (app.QF.PointXY (app.g/2 + app.p, app.rp),
    app.QF.PointXY(app.g/2 + app.p, app.ri));
mdl.Shapes.AddEdge (app.QF.PointXY (0, app.ri), app.QF.PointXY(app.g/2 + app.p,
    app.ri));
shp.Left.Item (1).Label = 'Cryostat';

% Create the Airgap
shp = mdl.Shapes.AddEdge (app.QF.PointXY(0, 0), app.QF.PointXY(app.g / 2, 0));
mdl.Shapes.AddEdge (app.QF.PointXY (0, 0), app.QF.PointXY(0, app.rp));
mdl.Shapes.AddEdge (app.QF.PointXY (app.g / 2, 0), app.QF.PointXY(app.g / 2,
    app.rp));
mdl.Shapes.AddEdge (app.QF.PointXY (0, app.rp), app.QF.PointXY(app.g/2, app.rp));
shp.Left.Item (1).Label = 'AirGap';

outerEdges = mdl.Shapes.get ('Boundary', 0);

% Symmetry axis
outerEdges.Item (2).Label = 'Symmetry';
outerEdges.Item (8).Label = 'Symmetry';
outerEdges.Item (7).Label = 'Symmetry';
outerEdges.Item (5).Label = 'Symmetry';



% Zero Magnetic Potential
outerEdges.Item (3).Label = 'Zero';
outerEdges.Item (4).Label = 'Zero';

 % Set Spacing
 pre = 1;
 sp = pre*app.L/3; % The spacing value
 mdl.Shapes.get ('LabeledAs', '', '', 'Cryostat').Spacing = sp;
```

```
        mdl.Shapes.get ('LabeledAs', '', '', 'Superconductor').Spacing = pre*app.L/100;
        mdl.Shapes.get ('LabeledAs', '', '', 'Air').Spacing = pre*app.L*8;
        mdl.Shapes.get ('LabeledAs', '', '', 'Steel').Spacing = sp;
        mdl.Shapes.get ('LabeledAs', '', '', 'Air Gap').Spacing = pre*app.L/20;
        mdl.Shapes.get ('LabeledAs', '', 'Symmetry', '').Spacing = pre*app.L/20;
        mdl.Shapes.get ('LabeledAs', '', 'Zero', '').Spacing = pre*app.L*8;

    % Generate the mesh
    try
        mdl.Shapes.BuildMesh ();
    catch
        error ('Mesh Problems');
    end

    % Save the complete model
    mdl.Save ();
end
function SetData(app,prb)
    % Set the media properties, filed sources and boundary conditions

    % First set properties for block labels

    blockLabels = prb.get ('Labels', 3);

    for i = 1 : blockLabels.Count

        lab = blockLabels.Item (i);
        cntBlock = lab.Content;

        switch lab.Name

            case 'Air'
                cntBlock.set('Kxx', false, 1);
                cntBlock.set('Kyy', false, 1);
            case 'AirGap'
                cntBlock.set('Kxx', false, 1);
                cntBlock.set('Kyy', false, 1);
            case 'Cryostat'
                cntBlock.set('Kxx', false, 1);
                cntBlock.set('Kyy', false, 1);
            case {'Steel'}
                spl = cntBlock.CreateBHCurve ();
                [a, b] = size(app.steel_data);
                j = 1;
                % Create BH curve for steel
```

```
                while j ~= a
                    spl.Add (app.QF.PointXY(app.steel_data(j,3*(app.Steel - 1) + 2),
                        app.steel_data(j,3*(app.Steel - 1) + 3)));
                    j = j + 1;
                    % In case the points for each steel are different
                    if app.steel_data(j,3*(app.Steel - 1) + 2) == 0
                        j = a;
                    end
                end
                cntBlock.Spline = spl;
            case {'Steel_air'}
                 spl = cntBlock.CreateBHCurve ();
                 [a, b] = size(app.steel_data);
                 j = 1;
                 % Create BH curve for steel
                 while j ~= a
                     spl.Add (app.QF.PointXY(app.steel_data(j,3*(app.Steel - 1) + 2),
                         (app.Frf*app.steel_data(j,3*(app.Steel - 1) + 3)) + (1 -
                         app.Frf)*app.steel_data(j,3*(app.Steel - 1) + 2)/app.Muo));
                     j = j + 1;
                     % In case the points for each steel are different
                     if app.steel_data(j,3*(app.Steel - 1) + 2) == 0
                         j = a;
                     end
                 end
                 cntBlock.Spline = spl;
            case {'Superconductor'}
                cntBlock.set('TotalCurrent', true);
                cntBlock.Loading = app.NI;
                % We will suppose the permeability of the superconductor is 1
                cntBlock.set('Kxx', false, 0.999991);
                cntBlock.set('Kyy', false, 0.999991);

        end % of switch
        lab.Content = cntBlock;
    end % of the loop

    % Magnetic Potential of zero at the exterior
    cntEdge = prb.get ('Labels' , 2).Item ('Zero').Content;
    cntEdge.Dirichlet = 0;

    prb.get ('Labels' , 2).Item ('Zero').Content = cntEdge;

    % Symmetry condition
    cntEdge = prb.get ('Labels' , 2).Item ('Symmetry').Content;
```

```
    cntEdge.Neumann = 0;

    prb.get ('Labels' , 2).Item ('Symmetry').Content = cntEdge;

    % Saving data document
    prb.DataDoc.Save;
end
function Solve(app,prb)
    % Solves the problem

    if prb.CanSolve
        prb.SolveProblem;
    end
end
function ViewResults(app, prb)
    if prb.Solved
        prb.AnalyzeResults;
    end
    res = prb.Result;

    if ~isinterface (res)
        disp ('error: Cannot get problem result');
        return;
    end

    fieldWin = res.GetFieldWindow (1);
    fieldWin.WindowState = 'qfMaximized';
    fieldWin.Height = fieldWin.Width + 10;

    bottomLeft = app.QF.PointXY (-50, -50);
    topRight = app.QF.PointXY (1.5*app.ext*(app.g/2 + app.p + app.h),
        1.5*app.ext*app.ro);

    fieldWin.Zoom (bottomLeft, topRight);

    cont = fieldWin.Contour;
    cont.AddLineTo (app.QF.PointXY (0,0));
    cont.AddLineTo (app.QF.PointXY (0,app.re));

    res.GetXYPlot(cont);
    % Clean contour
    cont.Delete;
    cont.release;
    fieldWin.release;
end
```

```
function closeQF(app,prb)
    %% closeQF
    % Close QF windows
    %  Input:
    %      global QF application
    %%
     prb.LoadModel;
     mdl = prb.Model; % get the geometry model

     mdl.Close ();
     mdl.release;

     % Closing data document
     prb.DataDoc.Close;

     res = prb.Result;

     res.Close;
end
function OptimizeQF(app,prb)

    i = 0;
    % Minimum number of iterations
    j = 5;
    e = 10;
    C = 1;

    % Minimize Area & check gradient C
    while ((i <= j) || (C > 0))
        % Call function, magnetic field
        [B0,A,C] = MagneticField(app,prb);
        app.Bsc = 1*maxField(app,prb);
        closeQF(app,prb);
        % In case we only have data of one Temp
        if app.T(1) == app.T(end)
            app.Jc = (app.Curve(app.Bsc))/(app.Cu_SC + 1);
        else
            app.Jc = (app.Curve(app.Bsc,app.Tk))/(app.Cu_SC + 1);
        end

        % Adjust the NI value
        NI_c = ((app.Bo/B0)^1.1)*app.NI;

        if C == 1
            z_c = (app.z + 3*e/4)/1000;
```

```
            e = e/2;
        elseif (C == 0) && (app.z > (app.g/2 + app.L/2 + app.c + e))
            z_c = (app.z - e)/1000;
        else
            z_c = (app.g/2 + app.L/2 + app.c)/1000;
        end

        % Change the geometry with the new values of NI and z
        Geometry1(app,NI_c,z_c);
        Geometry2(app);

        % Run QF and check the magnetic field
        prb = RunQF(app);

        % If this condition is fulfill break, limit the number of iterations
         if (round(app.z,1) == round((app.g/2 + app.L/2 + app.c),1)) && (i > 2) &&
             (round(app.Bo-B0,1)== 0)
                break;
         end

        i = i + 1;

    end
end
function [B0,A,C] = MagneticField(app,prb)

    res = prb.Result;
    if ~isinterface (res)
        disp ('error: Cannot get problem result');
        return;
    end

    % We will get ten points of the magnetic field from the midfield plane
    B = 1:1:51;
    % A is the Area of the curve, we will try to minimize the Area
    % maintaining a negative gradient
    A = 0;
    % C is the term that checks if the gradient is positive or negative.
    % If positive C = 1, if negative C = 0.
    C = 0;

    for i = 1:1:51
        B(i) = res.GetLocalValues(app.QF.PointXY (0,(i-1)*app.re/50)).Grad.X;
    end
```

```
    for i = 1:1:10
        A = A + (B(i) - B(51))*10;
        if (B(i) - B(i+1)) < 0
            C = 1;
        end
    end
    %B0 is the value of the magnetic field at the center
    B0 = B(1);
end
function B_max = maxField(app,prb)
    %% magneticField
    % Postprocessor. Checkt the maximum value of the magnetic field
    % Input:
    %         Global data
    % output:
    %         B_max:
    % QuickField object
    %% magneticField()

    res = prb.Result;
    if ~isinterface (res)
        disp ('error: Cannot get problem result');
        return;
    end

    Bmax = 1:1:41;

    for i = 1:1:41
        Bmax(i) = res.GetLocalValues(app.QF.PointXY(app.z - app.L/2 + (i-1)*app.L/40,
            app.rp + app.c)).Grad.X;
    end

    % Find maximum value
    B_max = max(Bmax);
end
function [Bmin,Bmax] = findBmaxmin(app)
    %% Definition
    % %findBmax()
    %  Looks for the maximum field for a certain temperature
    %
    %  Data for check:
    %      global data
    %  Output:
    %      global data
    %% finBmax
```

```
        [Max, I] = max(app.B(:));
        for i = app.B(1):app.B(I)
            Current_Density = app.Curve(i,app.Tk);
            if Current_Density >= 0
                Bmax = i;
            elseif Current_Density <= 0
                break;
            end
        end

        for i = app.B(1):app.B(I)
            Current_Density = app.Curve(i,app.Tk);
            if Current_Density >= 0
                Bmin = i;
                break;
            end
        end

        Bmax = Bmax/1.6;
        if Bmax >= 9
            Bmax = 9;
        end
    end
    function [W_out,B_out,P_out] = findOptimum(app)
        % Find Optimum
        i = 1;
        [B_min,B_max] = findBmaxmin(app);
        B_min = max([2 B_min]);
        pre = 0.005;    % Precision of the curve

        W_P = B_min:pre:B_max;
        P_P = B_min:pre:B_max;
        B = B_min:pre:B_max;

        for a = B_min:pre:B_max
            app.Bo = a;
            DetermineGeometry(app); % Determines the geometry of the ciclotron
            W_P(i) = app.W;
            B(i) = app.Bo;
            P_P(i) = app.P;
            i = i + 1;
        end
        [W_out,B_out,P_out] = Cost_Function(app,B,W_P,P_P);
    end
    function [W_out,B_out,P_out] = Cost_Function(app,B,W_P,P_P)
```

```
    % Cost function C = WF*W/Wmax + PF*P/Pmax
    % Maximum Weight
    Wmax = max(W_P(:));
    % Maximum Weight
    Pmax = max(P_P(:));

    % Minimum Weight
    Wmin = min(W_P(:));
    % Minimum Price
    Pmin = min(P_P(:));

    C = app.WF*(W_P-Wmin)/(Wmax - Wmin) + app.PF*(P_P-Pmin)/(Pmax-Pmin);

    % Plot C vs Field
    % Fit a curve for more detailed precision
    B = B';
    C = C';
    ft = fittype('smoothingspline');
    f = fit(B, C, ft);
    f1 = feval(f,B);
    plot(app.UIAxes_Optimum,B,f1);

    % Find Minimum
    [Cmin, I] = min(C(:));

    % Recalculate values at minimum
    app.Bo = B(I);
    DetermineGeometry(app);



    W_out = W_P(I);
    B_out = B(I);
    P_out = P_P(I);
end
function Read_Steel(app)
    % Read superconductor properties
    fileID = fopen(strcat('acero','.txt'),'r');
    app.dat_steel = textscan(fileID, '%f %s');
    fclose(fileID);
    steel_names = app.dat_steel{2};

    % Write items for dropdown
    app.SteelDropDown.Items = steel_names;
    app.SteelDropDown_Comparison.Items = steel_names;
    app.SteelDropDown_Optimum.Items = steel_names;
```

```
    app.SteelDropDown_OptimumScore.Items = steel_names;
    app.SteelDropDown_Score.Items = steel_names;
end
function Properties_steel(app,steel)
    steel_num = app.dat_steel{1};
    steel_names = app.dat_steel{2};

    for i = 1:size(steel_num)
        if strcmp(steel, steel_names{i})
            a = i;
            break;
        end
    end

    app.Steel = a;
end
function Read_superconductor(app,text)
    % Read superconductor properties
    fileID = fopen(strcat(text,'.txt'),'r');
    app.dat = textscan(fileID, '%f %s %f %f %f %f');
    fclose(fileID);
    supercond_names = app.dat{2};

    % Write items for dropdown
    app.SuperconductorDropDown_3.Items = supercond_names;
    app.SuperconductorDropDown.Items = supercond_names;
    app.SuperconductorDropDown_Optimum.Items = supercond_names;
    app.SuperconductorDropDownComparison.Items = supercond_names;
    app.SuperconductorDropDownComparison_2.Items = supercond_names;
    app.SuperconductorDropDownOptimumScore.Items = supercond_names;
    app.SuperconductorDropDown_Score_1.Items = supercond_names;
    app.SuperconductorDropDown_Score_2.Items = supercond_names;
    app.SuperconductorDropDown_Score_3.Items = supercond_names;
    app.SuperconductorDropDown_Score_4.Items = supercond_names;
end
function Write_supercond(app)
    Read_superconductor(app,'Superconductores');
    %
    supercond_num = app.dat{1};
    supercond_names = app.dat{2};
    supercond_fill = app.dat{3};
    supercond_price = app.dat{4};

    % Ask for name
    prompt = {'What is the name of the superconductor'};
```

```
    dlgtitle = 'Superconductor name';
    definput = {'NbTi_Supercon'};
    opts.Interpreter = 'None';
    name_answer = inputdlg(prompt,dlgtitle,[1 60],definput,opts);

    % Ask for fill factor
    prompt = {'What is the typical fill factor of the superconductor'};
    dlgtitle = 'Superconductor Fill Factor';
    definput = {'4'};
    opts.Interpreter = 'None';
    fill_answer = inputdlg(prompt,dlgtitle,[1 60],definput,opts);

    % Ask for price
    prompt = {'What is the price of the superconductor (ÿ/m3)'};
    dlgtitle = 'Superconductor price';
    definput = {'1183924'};
    opts.Interpreter = 'None';
    price_answer = inputdlg(prompt,dlgtitle,[1 60],definput,opts);

    % Add new data
    new_supercond_num = [supercond_num;supercond_num(end)+1];
    new_supercond_names = [supercond_names;name_answer];
    new_supercond_fill = [supercond_fill;str2double(fill_answer{1})];
    new_supercond_price = [supercond_price;str2double(price_answer{1})];

    new_data = table(new_supercond_num, new_supercond_names, new_supercond_fill,
        new_supercond_price);
    writetable(new_data,'Superconductores.txt','WriteVariableNames',false,"Delimiter",'
        ');
    Read_superconductor(app,'Superconductores')
end
function Properties_Supercond(app,supercond)
    supercond_num = app.dat{1};
    supercond_names = app.dat{2};
    supercond_fill = app.dat{3};
    supercond_price = app.dat{4};

    for i = 1:size(supercond_num)
        if strcmp(supercond, supercond_names{i})
            a = i;
            break;
        end
    end

    app.Supercond = a;
```

```
        app.Supercond_name = supercond_names{a};
        app.CopperSectionEditField.Value = supercond_fill(a);
        app.Cu_SC = app.CopperSectionEditField.Value;
        app.CopperSectionEditField_Optimum.Value = supercond_fill(a);
        app.Fill_Factor.Value = supercond_fill(a);
        app.p_unit = supercond_price(a);
    end
    function score_price = Price_score(app,P)
        supercond_price = P;
        score_price = zeros(size(supercond_price));

        %Eliminate maximum price infinite
        Max_Price = P;
        max_price = max(Max_Price);

        while max_price == inf
            [max_price, ind1] = max(Max_Price);
            if max_price == inf
                Max_Price(ind1) = -Inf;
            end
        end

        for i = 1:size(supercond_price)
            if supercond_price(i) == inf
                score_price(i) = 0;
            else
                score_price(i) =
                    10*(1-(supercond_price(i)-min(supercond_price))/max_price);
            end

        end

    end
    function score_weight = Weight_score(app,W)
        score_weight = zeros(size(W));

        %Eliminate maximum weight infinite
        Max_Weight = W;
        max_weight = max(Max_Weight);

        while max_weight == inf
            [max_weight, ind1] = max(Max_Weight);
            if max_weight == inf
                Max_Weight(ind1) = -Inf;
            end
```

```
    end

    for i = 1:size(W)
        if W(i) == inf
            score_weight(i) = 0;
        else
           score_weight(i) = 10*(1 - (W(i)-min(W))/max_weight);
        end
    end

end
function score_manufacturability = Manufacturability_score(app,superconductors)
    supercond_num = app.dat{1};
    supercond_names = app.dat{2};
    manufacturability = app.dat{6};
    score_manufacturability = zeros(size(superconductors));

    a = 1;
    for i = 1:size(supercond_num)
        if strcmp(superconductors(a), supercond_names{i})

            score_manufacturability(a) = manufacturability(i);
            a = a + 1;
        end
        if a > size(superconductors)
            break;
        end
    end

end
function score_length = Length_score(app,superconductors, L)
    supercond_num = app.dat{1};
    supercond_names = app.dat{2};
    length = app.dat{5};
    score_length = zeros(size(superconductors));
    supercond_length = zeros(size(superconductors));


    if size(L) == 1
        min_length = L*ones(size(length));
    else
        min_length = L;
    end

    a = 1;
```

```
    for i = 1:size(supercond_num)
        if strcmp(superconductors(a), supercond_names{i})
            supercond_length(a) = length(i);
            a = a + 1;
        end
        if a > size(superconductors)
            break;
        end
    end

    for i = 1:size(superconductors)
        score_length(i) = 10*exp(-0.5*floor(min_length(i)/supercond_length(i)));
    end

end
function score_performance = Performance_score(app,B,T)
    supercond_num = app.dat{1};
    supercond_names = app.dat{2};
    supercond_fill = app.dat{3};
    supercond_price = app.dat{4};

    score_performance = zeros(size(supercond_num));
    Performance = zeros(size(supercond_num));

    if size(L) == size(supercond_length)
        B = B;
    else
        B = B*ones(size(supercond_length));
    end

    for i = 1:size(supercond_num)
        app.Supercond = supercond_num(i);
        app.Supercond_name = supercond_names{i};
        Perf_Curve = fitCurve(app);
        if isnan(Perf_Curve(B(i),T))
            Performance(i) = 0;
        else
            Performance(i) = Perf_Curve(B(i),T)/(1+supercond_fill(i));
        end

    end
    for i = 1:size(supercond_num)
        score_performance(i) = (Performance(i)-min(Performance))/max(Performance);
    end
```

```
end
function score_perf_price = Perf_Price_score(app,B,T)
    supercond_num = app.dat{1};
    supercond_names = app.dat{2};
    supercond_fill = app.dat{3};
    supercond_price = app.dat{4};

    score_perf_price = zeros(size(supercond_num));
    Performance = zeros(size(supercond_num));
    Max_Performance = zeros(size(supercond_num));

    if size(B) == size(supercond_num)
        B = B;
    else
        B = B*ones(size(supercond_num));
    end

    for i = 1:size(supercond_num)
        app.Supercond = supercond_num(i);
        app.Supercond_name = supercond_names{i};
        Perf_Curve = fitCurve(app);
        if isnan(Perf_Curve(B(i),T))
            Performance(i) = inf;
        else
            Performance(i) =
                supercond_price(i)/Perf_Curve(B(i),T)*(1+supercond_fill(i));
        end

    end
    % Check maximum different than inf
    Max_Performance = Performance;
    max_perf = max(Max_Performance);

    while max_perf == inf
        [max_perf, ind1] = max(Max_Performance);
        if max_perf == inf
            Max_Performance(ind1) = -Inf;
        end
    end

    % Give scores
    for i = 1:size(supercond_num)
        if Performance(i) == Inf
            score_perf_price(i) = 0;
        else
```

```
            score_perf_price(i) = 10*(1 - (Performance(i)-min(Performance))/max_perf);
        end
    end

end
function Score = Material_Cost_function(app, supercond, Perf_Price_factor,
    Length_factor, Man_factor,B,T,L)
    %score_performance =
        Performance_score(app,app.BTComparison.Value,app.TKComparison.Value);
    %score_price = Price_score(app);

    app.score_perf_price = Perf_Price_score(app,B,T);
    app.score_length = Length_score(app, supercond,L);
    app.score_manufacturability = Manufacturability_score(app, supercond);

    Score = zeros(size(app.score_length));

    for i = 1:size(app.score_length)
        Score(i) = Perf_Price_factor*app.score_perf_price(i) +
            Length_factor*app.score_length(i) +
            Man_factor*app.score_manufacturability(i);
    end

end
function Score = Cyclotron_Cost_function(app, supercond_names,
    Weight_factor,Price_factor, Length_factor, Man_factor,W,P,L)

    app.score_weight = Weight_score(app,W);
    app.score_price = Price_score(app,P);
    app.score_length = Length_score(app,supercond_names,L);
    app.score_manufacturability = Manufacturability_score(app,supercond_names);

    Score = zeros(size(supercond_names));

    for i = 1:size(supercond_names)
        Score(i) = Weight_factor*app.score_weight(i) +
            Price_factor*app.score_price(i) + Length_factor*app.score_length(i) +
            Man_factor*app.score_manufacturability(i);
    end

end

function Rankings(app)

    X = app.score_total;
```

```
    supercond_names = app.dat{2};

    [max1, ind1] = max(X);
    X(ind1)      = -Inf;
    app.Name1.Value = supercond_names{ind1};
    app.Score1.Value = max1;
    app.Name1_Comparison.Value = supercond_names{ind1};
    app.Score1_Comparison.Value = max1;
    app.Name1_OptimumScore.Value = supercond_names{ind1};
    app.Score1_OptimumScore.Value = max1;

    [max2, ind2] = max(X);
    X(ind2)      = -Inf;
    app.Name2.Value = supercond_names{ind2};
    app.Score2.Value = max2;
    app.Name2_Comparison.Value = supercond_names{ind2};
    app.Score2_Comparison.Value = max2;
    app.Name2_OptimumScore.Value = supercond_names{ind2};
    app.Score2_OptimumScore.Value = max2;

    [max3, ind3] = max(X);
    X(ind3)      = -Inf;
    app.Name3.Value = supercond_names{ind3};
    app.Score3.Value = max3;
    app.Name3_Comparison.Value = supercond_names{ind3};
    app.Score3_Comparison.Value = max3;
    app.Name3_OptimumScore.Value = supercond_names{ind3};
    app.Score3_OptimumScore.Value = max3;
end
function [Score, Position] = Superconductor_Score(app,supercond,type)
    supercond_num = app.dat{1};
    supercond_names = app.dat{2};

    for i = 1:size(supercond_num)
        if strcmp(supercond, supercond_names{i})
            a = i;
            break;
        end
    end

    if strcmp(type,'Total Score')
        Score = app.score_total(a);
        Position = find_position(app,app.score_total,a);
        app.ScoreNumber.Value = app.score_total(a);
        app.ScoreNumber_2.Value = find_position(app,app.score_total,a);
```

```
    elseif strcmp(type,'Weight')
        Score = app.score_weight(a);
        Position = find_position(app,app.score_weight,a);
    elseif strcmp(type,'Price')
        Score = app.score_price(a);
        Position = find_position(app,app.score_price,a);
    elseif strcmp(type,'Perf/Price')
        Score = app.score_perf_price(a);
        Position = find_position(app,app.score_perf_price,a);
        app.ScoreNumber.Value = app.score_perf_price(a);
        app.ScoreNumber_2.Value = find_position(app,app.score_perf_price,a);


    elseif strcmp(type,'Length')
        Score = app.score_length(a);
        Position = find_position(app, app.score_length,a);
        app.ScoreNumber.Value = app.score_length(a);
        app.ScoreNumber_2.Value = find_position(app, app.score_length,a);


    elseif strcmp(type,'Manufacturability')
        Score = app.score_manufacturability(a);
        Position = find_position(app, app.score_manufacturability,a);
        app.ScoreNumber.Value = app.score_manufacturability(a);
        app.ScoreNumber_2.Value = find_position(app, app.score_manufacturability,a);


    end
end
function rank = find_position(~,array,a)
    array_order = sort(array,'descend');
    rank = find(array_order == array(a));
    rank = rank(1);
end
function [Score, Position] = Score_Position(app, type, a)

    if strcmp(type,'Total Score')
        Score = app.score_total(a);
        Position = find_position(app,app.score_total,a);
    elseif strcmp(type,'Weight')
        Score = app.score_weight(a);
        Position = find_position(app,app.score_weight,a);
    elseif strcmp(type,'Price')
```

```
            Score = app.score_price(a);
            Position = find_position(app,app.score_price,a);
        elseif strcmp(type,'Perf/Price')
            Score = app.score_perf_price(a);
            Position = find_position(app,app.score_perf_price,a);

        elseif strcmp(type,'Length')
            Score = app.score_length(a);
            Position = find_position(app, app.score_length,a);

        elseif strcmp(type,'Manufacturability')
            Score = app.score_manufacturability(a);
            Position = find_position(app, app.score_manufacturability,a);

        end
    end
end


% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        % Initial Conditions
        app.E = 8.5;              % (MeV)
        app.Bo = 4;               % (T)
        app.Tk = 4.2;             % (K)
        app.g = 74;              % (mm)
        app.c = 32;               % (mm)
        app.Muo = 4*pi*10^-7;     % (T*m/A)
        app.I = 100;              % (A)
        app.Cu_SC = 4.5;
        app.WP = 0.65;
        app.AR = 1;
        app.Rep = 0.7;
        app.Rosc = 1.75;
        app.alpha = 1.2;
        app.ext = 4;
        app.WF = 1;
        app.PF = 0;
        app.Ff = 0.9;
        app.Frf = 0.9;
        app.answer = {};
```

```
        app.Supercond = 2;      % [1] = Cu;    [2] = NbTi_Bruker; [3] =  NbTi_Supercon;
        % [4] =  NbSn_Bruker;  [5] =  NbSn_Bruker;  [6] =  MgB2_ASG; [7] =  MgB2_Hyper;
        % [8] =  HTS_Sumitomo; [9] =  HTS_Superpower;   [10] =  HTS_Fujikura;
        % [11] =  HTS_Sinn
        app.Supercond_name = 'NbTi_Bruker';
        app.Steel = 1;           % [1] = 1010; [2] =  1020; [3] =  ARMCO  [4] = Hyperco50
        app.rho_sc = 9000;       % (kg/m3)

        addpath(fullfile(pwd,'Functions'));
        Read_Steel(app);
        Properties_steel(app,'AISI1010');
        Read_superconductor(app,'Superconductores');
        Properties_Supercond(app,app.Supercond_name);
        [app.Curve,app.T,app.B] = fitCurve(app);
    end

    % Selection change function: TabGroup
    function TabGroupSelectionChanged(app, event)
        % Reset initial values
        startupFcn(app);

        % Reset Energy Value
        app.EMeVEditField.Value = 8.5;
        app.EMeVEditField_Optimum.Value = 8.5;

        % Reset Magnetic field
        app.BTEditField.Value = 4;

        % Reset Airgap value
        app.AirgapmmEditField.Value = 74;
        app.AirgapmmEditField_Optimum.Value = 74;

        % Reset Cryostat value
        app.CryostatmmEditField.Value = 32;
        app.CryostatmmEditField_Optimum.Value = 32;

        % Reset Copper/NonCopper value
        app.CopperSectionEditField.Value = 4.5;
        app.CopperSectionEditField_Optimum.Value = 4.5;

        % Reset AR value
        app.AspectRatioEditField.Value = 1;
        app.AspectRatioEditField_Optimum.Value = 1;

        % Reset the superconductor drop down
```

```
    app.SuperconductorDropDown_Optimum.Value = 'NbTi_Bruker';
    app.SuperconductorDropDown.Value = 'NbTi_Bruker';
    app.SuperconductorDropDown_3.Value = 'NbTi_Bruker';

    % Reset the steel drop down
    app.SteelDropDown.Value = 'AISI1010';
    app.SteelDropDown_Optimum.Value = 'AISI1010';

    % Reset cooling temp
    app.CoolingTempEditField_Optimum.Value = 4.2;
    app.CoolingTempEditField.Value = 4.2;

end

% Value changed function: SuperconductorDropDown_3
function SuperconductorDropDown_3ValueChanged(app, event)
    value = app.SuperconductorDropDown_3.Value;

    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    [app.Curve,app.T,app.B]=fitCurve(app);
    app.J_Material.Value = app.Curve(app.BT_Material.Value,app.TK_Material.Value)/(1
        + app.Fill_Factor.Value);
    app.TK_Material.Limits = [app.T(1) app.T(end)];
end

% Value changed function: BT_Material
function BT_MaterialValueChanged(app, event)
    app.J_Material.Value = app.Curve(app.BT_Material.Value,app.TK_Material.Value)/(1
        + app.Fill_Factor.Value);
end

% Value changed function: TK_Material
function TK_MaterialValueChanged(app, event)
    app.J_Material.Value = app.Curve(app.BT_Material.Value,app.TK_Material.Value)/(1
        + app.Fill_Factor.Value);
end

% Value changed function: Fill_Factor
function Fill_FactorValueChanged(app, event)
    app.J_Material.Value = app.Curve(app.BT_Material.Value,app.TK_Material.Value)/(1
        + app.Fill_Factor.Value);
end

% Button pushed function: AddMaterialButton
```

```
function AddMaterialButtonPushed(app, event)
    Write_supercond(app);
end

% Button pushed function: RunButtonComparison
function RunButtonComparisonPushed(app, event)
    % Read values
    % Weights
    Perf_Price_weight = app.Performance.Value;
    Length_weight = app.Length.Value;
    Man_weight = app.Manufacturability.Value;

    supercond_names = app.dat{2};
    B = app.BTComparison.Value;
    T = app.TKComparison.Value;
    L = app.MinLength.Value;

    % Check sum
    sum = Perf_Price_weight + Length_weight + Man_weight;
    if sum ~= 1
        error('Check that the weights sum is equal to 1')
    end
    % Cost function

    app.score_total = Material_Cost_function(app,supercond_names,
        Perf_Price_weight,Length_weight,Man_weight,B,T,L);

    % Create rankings

    Rankings(app);

    % Give Superconductor performance

    supercond = app.SuperconductorDropDownComparison.Value;
    type = app.ScoreDropDown.Value;
    Superconductor_Score(app,supercond,type);
end

% Value changed function: SuperconductorDropDownComparison
function SuperconductorDropDownComparisonValueChanged(app, event)
    % Superconductor
    value = app.SuperconductorDropDownComparison.Value;

    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
```

```
    fitCurve(app);

    % Give Superconductor performance

    supercond = app.SuperconductorDropDownComparison.Value;
    type = app.ScoreDropDown.Value;
    Superconductor_Score(app,supercond,type);
end

% Value changed function: ScoreDropDown
function ScoreDropDownValueChanged(app, event)
    % Give Superconductor performance

    supercond = app.SuperconductorDropDownComparison.Value;
    type = app.ScoreDropDown.Value;
    Superconductor_Score(app,supercond,type);
end

% Value changed function: OnlySuperconductorsCheckBox
function OnlySuperconductorsCheckBoxValueChanged(app, event)
    value = app.OnlySuperconductorsCheckBox.Value;

    if value == 1
        Read_superconductor(app,'Superconductores_No_Copper');
    else
        Read_superconductor(app,'Superconductores');
    end
end

% Value changed function: OnlyReliableDataCheckBox
function OnlyReliableDataCheckBoxValueChanged(app, event)
    value = app.OnlyReliableDataCheckBox.Value;

    if value == 1
        Read_superconductor(app,'Superconductores_Reliable_Data');
    else
        Read_superconductor(app,'Superconductores');
    end
end

% Button pushed function: RunButton
function RunButtonPushed(app, event)
    % Read values
    % Energy (MeV)
    app.E = app.EMeVEditField.Value;
```

```
    % B at the center (T)
    app.Bo = app.BTEditField.Value;

    % Current through wire
    app.I = app.WireCurentAEditField.Value;

    % Working point
    app.WP = app.WorkingPointEditField.Value;

    % Airgap (mm)
    app.g = app.AirgapmmEditField.Value;

    % Cryostat width (mm)
    app.c = app.CryostatmmEditField.Value;

    % Copper Section
    app.Cu_SC = app.CopperSectionEditField.Value;

    % Aspect Ratio
    app.AR = app.AspectRatioEditField.Value;

    % Function that determines the geometry of the cyclotron
    DetermineGeometry(app);

    % Plot cyclotron in UIAxes
    PlotCyclotron(app);

    % Show Output Data
    app.WeightkgEditField.Value = app.W;
    app.PriceEditField.Value = app.P;
    app.WireDiammmEditField.Value = app.wire_diameter;
    app.WireLenghtmEditField.Value = app.wire_lenght;
    app.WireAreammEditField.Value = app.wire_area;
    app.DiametermEditField_Custom.Value = app.w/1000;
end

% Button pushed function: RunQFButton
function RunQFButtonPushed(app, event)
    % In case Run button has no been pushed run it
    RunButtonPushed(app);
    % Run QF
    prb = RunQF(app);

    % Ask User if he wants to optimize the case with QF
```

```
    response = questdlg('Would you like to optimize the case?');
    switch response
        case 'Yes'
            % Optimize case call function
            OptimizeQF(app,prb);

            % Plot cyclotron with new data and update weight and
            % price
            PlotCyclotron(app);
            app.WeightkgEditField.Value = app.W;
            app.PriceEditField.Value = app.P;
            app.WireDiammmEditField.Value = app.wire_diameter;
            app.WireLenghtmEditField.Value = app.wire_lenght;
        case 'No'
    end

end

% Value changed function: SteelDropDown
function SteelDropDownValueChanged(app, event)
    value = app.SteelDropDown.Value;

    Properties_steel(app,value);
end

% Value changed function: SuperconductorDropDown
function SuperconductorDropDownValueChanged(app, event)
    value = app.SuperconductorDropDown.Value;

    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);

    % Set Superconductor curve
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);
    app.CoolingTempEditField.Limits = [app.T(1) app.T(end)];
end

% Value changed function: CoolingTempEditField
function CoolingTempEditFieldValueChanged(app, event)
    value = app.CoolingTempEditField.Value;
    app.Tk = value;
end

% Value changed function: WireDiammmEditField
```

```
function WireDiammmEditFieldValueChanged(app, event)
    value = app.WireDiammmEditField.Value;

    app.WireCurentAEditField.Value = app.Jc*app.WP*pi*((value/2)^2)/1000000;
end

% Value changed function: WireAreammEditField
function WireAreammEditFieldValueChanged(app, event)
    value = app.WireAreammEditField.Value;

    app.WireCurentAEditField.Value = app.Jc*app.WP*value/1000000;
end

% Button pushed function: RunButton_Optimum
function RunButton_OptimumPushed(app, event)
    % Read values
    % Energy (MeV)
    app.E = app.EMeVEditField_Optimum.Value;

    % Working point
    app.WP = app.WorkingPointEditField_Optimum.Value;

    % Airgap (mm)
    app.g = app.AirgapmmEditField_Optimum.Value;

    % Cryostat width (mm)
    app.c = app.CryostatmmEditField_Optimum.Value;

    % Copper Section
    app.Cu_SC = app.CopperSectionEditField_Optimum.Value;

    % Aspect Ratio
    app.AR = app.AspectRatioEditField_Optimum.Value;

    % Weight Factor
    app.WF = app.WeightFactorEditField.Value;

    % Price Factor
    app.PF = app.PriceFactorEditField.Value;

    % Find Optimum
    [W_out,B_out,P_out] = findOptimum(app);

    app.WeightkgEditField_Optimum.Value = W_out;
    app.BTEditField_Optimum.Value = B_out;
```

```
    app.PriceEditField_Optimum.Value = P_out;
    app.DiametermEditField_Optimum.Value = app.w/1000;
end

% Button pushed function: RunQFButton_Optimum
function RunQFButton_OptimumPushed(app, event)
    RunButton_OptimumPushed(app);

    app.Bo = app.BTEditField_Optimum.Value;
    DetermineGeometry(app);
    prb = RunQF(app);
    % Ask User if he wants to optimize the case with QF
    response = questdlg('Would you like to optimize the case?');
    switch response
        case 'Yes'
            OptimizeQF(app,prb);
            app.WeightkgEditField_Optimum.Value = app.W;
            app.PriceEditField_Optimum.Value = app.P;
        case 'No'
    end

end

% Value changed function: SteelDropDown_Optimum
function SteelDropDown_OptimumValueChanged(app, event)
    value = app.SteelDropDown_Optimum.Value;

    Properties_steel(app,value);
end

% Value changed function: SuperconductorDropDown_Optimum
function SuperconductorDropDown_OptimumValueChanged(app, event)
    value = app.SuperconductorDropDown_Optimum.Value;

    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    % Set Superconductor curve
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);
    app.CoolingTempEditField_Optimum.Limits = [app.T(1) app.T(end)];
end

% Value changed function: CoolingTempEditField_Optimum
function CoolingTempEditField_OptimumValueChanged(app, event)
    value = app.CoolingTempEditField_Optimum.Value;
```

```
    app.Tk = value;
end

% Value changed function: WeightFactorEditField
function WeightFactorEditFieldValueChanged(app, event)
    value = app.WeightFactorEditField.Value;

    app.PriceFactorEditField.Value = 1 - value;
end

% Value changed function: PriceFactorEditField
function PriceFactorEditFieldValueChanged(app, event)
    value = app.PriceFactorEditField.Value;

    app.WeightFactorEditField.Value = 1 - value;
end

% Button pushed function: Zomm_Optimum
function Zoom_OptimumButtonPushed(app, event)
    %% optimum(NI_c,z_c)
    %  Determines Weight an Price through a range of fields
    %  Data:
    %      global data
    %  Output:
    %      answer = Answer to run it in QF.
    %% optimum

    if app.BTEditField_Optimum.Value == 0
        RunButton_OptimumPushed(app)
    end
    % Ask User if he wants to run a simulation in QF
    zoom_answer = questdlg('Would you like to input the limits?');

    switch zoom_answer
        case 'Yes'
            % Ask for input
            prompt = {'Enter lower limit','Enter upper limit','Step size'};
            dlgtitle = 'Limits';
            definput = {'3','4','0.1'};
            opts.Interpreter = 'None';
            Limits_answer = inputdlg(prompt,dlgtitle,[1 60],definput,opts);

            B_min = str2double(Limits_answer(1));
            B_max = str2double(Limits_answer(2));
```

```
            step = str2double(Limits_answer(3));
        case 'No'
            B_min = app.BTEditField_Optimum.Value - 0.5;
            B_max = app.BTEditField_Optimum.Value + 0.5;
            step = 0.1;     % Precision of the curve
    end

    B = B_min:step:B_max;
    W_P = B_min:step:B_max;
    P_P = B_min:step:B_max;

    % Find Optimum
    i = 1;
    for a = B_min:step:B_max
        app.Bo = a;
        DetermineGeometry(app); % Determines the geometry of the ciclotron
        prb = RunQF(app);
        OptimizeQF(app,prb);
        closeQF(app,prb);
        B(i) = app.Bo;
        W_P(i) = app.W;
        P_P(i) = app.P;
        i = i + 1;
    end
    Cost_Function(app,B,W_P,P_P);
    % Save results in .csv file
    Results = [B W_P P_P];
    name = strcat(num2str(app.Supercond),'_',num2str(app.Tk),'_optimized');
    writematrix(Results,string(strcat(name,'.csv')));
end

% Button pushed function: RunButton_Comparison
function RunButton_ComparisonPushed(app, event)
    % Read values
    % Weights
    Weight_weight = app.Performance_Comparison.Value;
    Price_weight = app.Price_Comparison.Value;
    Length_weight = app.Length_Comparison.Value;
    Man_weight = app.Manufacturability_Comparison.Value;

    % Energy (MeV)
    app.E = app.EMeVEditField_Comparison.Value;

    % B at the center (T)
    app.Bo = app.BTEditField_Comparison.Value;
```

```
% Current through wire
app.I = app.WireCurentAEditField_Comparison.Value;

% Working point
app.WP = app.WorkingPointEditField_Comparison.Value;

% Airgap (mm)
app.g = app.AirgapmmEditField_Comparison.Value;

% Cryostat width (mm)
app.c = app.CryostatmmEditField_Comparison.Value;

% Aspect Ratio
app.AR = app.AspectRatioEditField_Comparison.Value;
% Cooling temperature
app.Tk = app.CoolingTempEditField_Comparison.Value;

% Function that determines the geometry of the cyclotron
supercond_num = app.dat{1};
supercond_names = app.dat{2};
L = zeros(size(supercond_names));
W = zeros(size(supercond_names));
P = zeros(size(supercond_names));
T = app.CoolingTempEditField_Comparison.Value;

% For loop to determine Bsc and Lengths
for i = 1:size(supercond_num)
    value = supercond_names(i);
    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);

    if app.Curve(app.Rosc*app.Bo,T) >= 0
        DetermineGeometry(app);
        L(i) = app.wire_lenght;
        W(i) = app.W;
        P(i) = app.P;
    else
        L(i) = inf;
        W(i) = inf;
        P(i) = inf;
    end
end
```

```
        % Check sum
        sum = Weight_weight + Price_weight + Length_weight + Man_weight;
        if sum ~= 1
            error('Check that the weights sum is equal to 1')
        end

        % Cost function

        app.score_total = Cyclotron_Cost_function(app, supercond_names,
            Weight_weight,Price_weight,Length_weight,Man_weight,W,P,L);

        % Create rankings

        Rankings(app);

        SuperconductorDropDownComparison_2ValueChanged(app, event);
    end

    % Button pushed function: RunQFButton_Comparison
    function RunQFButton_ComparisonPushed(app, event)
        RunButton_ComparisonPushed(app, event);
        % Weights
        Weight_weight = app.Performance_Comparison.Value;
        Price_weight = app.Price_Comparison.Value;
        Length_weight = app.Length_Comparison.Value;
        Man_weight = app.Manufacturability_Comparison.Value;

        % Function that determines the geometry of the cyclotron
        supercond_num = app.dat{1};
        supercond_names = app.dat{2};
        L = zeros(size(supercond_names));
        W = zeros(size(supercond_names));
        P = zeros(size(supercond_names));
        T = app.CoolingTempEditField_Comparison.Value;

        % For loop to determine Bsc and Lengths
        for i = 1:size(supercond_names)
            value = supercond_names(i);
            % Change type of superconductor and adjust temperature
            Properties_Supercond(app,value);
            % Adjust cooling temperature limits
            [app.Curve,app.T,app.B] = fitCurve(app);
```

```
            if app.Curve(app.Rosc*app.Bo,T) >= 0
                DetermineGeometry(app);
                % Run QF
                prb = RunQF(app);
                % Optimize case call function
                OptimizeQF(app,prb);
                closeQF(app,prb);

                L(i) = app.wire_lenght;
                W(i) = app.W;
                P(i) = app.P;
            else
                L(i) = inf;
                W(i) = inf;
                P(i) = inf;
            end
        end

        % Cost function

        app.score_total = Cyclotron_Cost_function(app, supercond_names,
            Weight_weight,Price_weight,Length_weight,Man_weight,W,P,L);

        % Create rankings

        Rankings(app);

        SuperconductorDropDownComparison_2ValueChanged(app, event);
    end

    % Value changed function: SuperconductorDropDownComparison_2
    function SuperconductorDropDownComparison_2ValueChanged(app, event)
        % Superconductor
        value = app.SuperconductorDropDownComparison_2.Value;

        % Change type of superconductor and adjust temperature
        Properties_Supercond(app,value);
        fitCurve(app);
        % Adjust cooling temperature limits
        [app.Curve,app.T,app.B] = fitCurve(app);

        % Read values
        % Energy (MeV)
        app.E = app.EMeVEditField_Comparison.Value;
```

```
% B at the center (T)
app.Bo = app.BTEditField_Comparison.Value;

% Current through wire
app.I = app.WireCurentAEditField_Comparison.Value;

% Working point
app.WP = app.WorkingPointEditField_Comparison.Value;

% Airgap (mm)
app.g = app.AirgapmmEditField_Comparison.Value;

% Cryostat width (mm)
app.c = app.CryostatmmEditField_Comparison.Value;

% Aspect Ratio
app.AR = app.AspectRatioEditField_Comparison.Value;

% Cooling temperature
app.Tk = app.CoolingTempEditField_Comparison.Value;



if app.Curve(app.Bsc,app.Tk)>=0
    % Function that determines the geometry of the cyclotron
    DetermineGeometry(app);
    % Show Output Data
    app.WeightkgEditField_Comparison.Value = app.W;
    app.PriceEditField_Comparison.Value = app.P;
    app.WireDiammmEditField_Comparison.Value = app.wire_diameter;
    app.WireLenghtmEditField_Comparison.Value = app.wire_lenght;
    app.WireAreammEditField_Comparison.Value = app.wire_area;
    app.DiametermEditField_Custom_Comparison.Value = app.w/1000;
else
    % Show Output Data
    app.WeightkgEditField_Comparison.Value = 0;
    app.PriceEditField_Comparison.Value = 0;
    app.WireDiammmEditField_Comparison.Value = 0;
    app.WireLenghtmEditField_Comparison.Value = 0;
    app.WireAreammEditField_Comparison.Value = 0;
    app.DiametermEditField_Custom_Comparison.Value = 0;
end

% Give Superconductor performance
ScoreDropDown_ComparisonValueChanged(app, event)
```

```
end

% Value changed function: SteelDropDown_Comparison
function SteelDropDown_ComparisonValueChanged(app, event)
    value = app.SteelDropDown_Comparison.Value;

    Properties_steel(app,value);
end

% Value changed function: ScoreDropDown_Comparison
function ScoreDropDown_ComparisonValueChanged(app, event)
    % Give Superconductor performance
    supercond = app.SuperconductorDropDownComparison_2.Value;
    type = app.ScoreDropDown_Comparison.Value;
    [Score, Position] = Superconductor_Score(app,supercond,type);

    app.ScoreNumber_Comparison.Value = Score;
    app.ScoreNumber2_Comparison.Value = Position;
end

% Value changed function: OnlySuperconductorsCheckBox_2
function OnlySuperconductorsCheckBox_2ValueChanged(app, event)
    value = app.OnlySuperconductorsCheckBox_2.Value;

    if value == 1
        Read_superconductor(app,'Superconductores_No_Copper');
    else
        Read_superconductor(app,'Superconductores');
    end
end

% Value changed function: OnlyReliableDataCheckBox_2
function OnlyReliableDataCheckBox_2ValueChanged(app, event)
    value = app.OnlyReliableDataCheckBox_2.Value;

    if value == 1
        Read_superconductor(app,'Superconductores_Reliable_Data');
    else
        Read_superconductor(app,'Superconductores');
    end
end

% Button pushed function: RunButton_OptimumScore
function RunButton_OptimumScorePushed(app, event)
    % Read values
```

```
% Weights
Weight_weight = app.Performance_OptimumScore.Value;
Price_weight = app.Price_OptimumScore.Value;
Length_weight = app.Length_OptimumScore.Value;
Man_weight = app.Manufacturability_OptimumScore.Value;

% Energy (MeV)
app.E = app.EMeVEditField_OptimumScore.Value;

% Current through wire
app.I = app.WireCurentAEditField_OptimumScore.Value;

% Working point
app.WP = app.WorkingPointEditField_OptimumScore.Value;

% Airgap (mm)
app.g = app.AirgapmmEditField_OptimumScore.Value;

% Cryostat width (mm)
app.c = app.CryostatmmEditField_OptimumScore.Value;

% Aspect Ratio
app.AR = app.AspectRatioEditField_OptimumScore.Value;
% Cooling temperature
app.Tk = app.CoolingTempEditField_OptimumScore.Value;

% Function that determines the geometry of the cyclotron
supercond_num = app.dat{1};
supercond_names = app.dat{2};
L = zeros(size(supercond_names));
W = zeros(size(supercond_names));
P = zeros(size(supercond_names));
T = app.CoolingTempEditField_OptimumScore.Value;

% For loop to determine Bsc and Lengths
for i = 1:size(supercond_num)
    value = supercond_names(i);
    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);

    if app.Curve(2,T) >= 0
        [W_out,B_out,P_out] = findOptimum(app);
        W(i) = W_out;
```

```
            P(i) = P_out;
            L(i) = app.wire_lenght;
        else
            L(i) = inf;
            W(i) = inf;
            P(i) = inf;
        end

    end



    % Check sum
    sum = Weight_weight + Price_weight + Length_weight + Man_weight;
    if sum ~= 1
        error('Check that the weights sum is equal to 1')
    end

    % Cost function

    app.score_total = Cyclotron_Cost_function(app, supercond_names, Weight_weight,
        Price_weight,Length_weight,Man_weight,W,P,L);

    % Create rankings

    Rankings(app);

    SuperconductorDropDownOptimumScoreValueChanged(app, event);
end

% Value changed function: SuperconductorDropDownOptimumScore
function SuperconductorDropDownOptimumScoreValueChanged(app, event)
    value = app.SuperconductorDropDownOptimumScore.Value;
    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    fitCurve(app);
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);

    % Read values
    % Energy (MeV)
    app.E = app.EMeVEditField_OptimumScore.Value;

    % Working point
    app.WP = app.WorkingPointEditField_OptimumScore.Value;
```

```
        % Airgap (mm)
        app.g = app.AirgapmmEditField_OptimumScore.Value;

        % Cryostat width (mm)
        app.c = app.CryostatmmEditField_OptimumScore.Value;

        % Aspect Ratio
        app.AR = app.AspectRatioEditField_OptimumScore.Value;

        % Cooling Temperature
        app.Tk = app.CoolingTempEditField_OptimumScore.Value;

        % Weight Factor
        app.WF = 1;

        % Price Factor
        app.PF = 0;

        if app.Curve(2,app.Tk)>=0
            [W_out,B_out,P_out] = findOptimum(app);

            app.WeightkgEditField_OptimumCyclotron.Value = W_out;
            app.BTEditField_OptimumScore.Value = B_out;
            app.PriceEditField_OptimumCyclotron.Value = P_out;
            app.DiametermEditField_OptimumCyclotron.Value = app.w/1000;
        else
            app.WeightkgEditField_OptimumCyclotron.Value = 0;
            app.BTEditField_OptimumScore.Value = 0;
            app.PriceEditField_OptimumCyclotron.Value = 0;
            app.DiametermEditField_OptimumCyclotron.Value = 0;
        end



        % Give Superconductor performance
        ScoreDropDown_OptimumScoreValueChanged(app, event)
    end

    % Value changed function: ScoreDropDown_OptimumScore
    function ScoreDropDown_OptimumScoreValueChanged(app, event)

        % Give Superconductor performance
        supercond = app.SuperconductorDropDownOptimumScore.Value;
        type = app.ScoreDropDown_OptimumScore.Value;
        [Score, Position] = Superconductor_Score(app,supercond,type);
```

```
    app.ScoreNumber_OptimumScore.Value = Score;
    app.ScoreNumber2_OptimumScore.Value = Position;
end

% Value changed function: SteelDropDown_OptimumScore
function SteelDropDown_OptimumScoreValueChanged(app, event)
    value = app.SteelDropDown_OptimumScore.Value;

    Properties_steel(app,value);
end

% Value changed function: OnlySuperconductorsCheckBox_3
function OnlySuperconductorsCheckBox_3ValueChanged(app, event)
    value = app.OnlySuperconductorsCheckBox_3.Value;

    if value == 1
        Read_superconductor(app,'Superconductores_No_Copper');
    else
        Read_superconductor(app,'Superconductores');
    end
end

% Value changed function: OnlyReliableDataCheckBox_3
function OnlyReliableDataCheckBox_3ValueChanged(app, event)
    value = app.OnlyReliableDataCheckBox_3.Value;

    if value == 1
        Read_superconductor(app,'Superconductores_Reliable_Data');
    else
        Read_superconductor(app,'Superconductores');
    end
end

% Button pushed function: RunButton_Score
function RunButton_ScorePushed(app, event)
    % Read values
    % Weights
    Weight_weight = app.Performance_Score.Value;
    Price_weight = app.Price_Score.Value;
    Length_weight = app.Length_Score.Value;
    Man_weight = app.Manufacturability_Score.Value;

    % Energy (MeV)
    app.E = app.EMeVEditField_Score.Value;
```

```
% Current through wire
app.I = app.WireCurentAEditField_Score.Value;

% Working point
app.WP = app.WorkingPointEditField_Score.Value;

% Cryostat width (mm)
app.c = app.CryostatmmEditField_Score.Value;

% Aspect Ratio
app.AR = app.AspectRatioEditField_Score.Value;


% Function that determines the geometry of the cyclotron

L = zeros(4,1);
W = zeros(4,1);
P = zeros(4,1);
T = app.CoolingTempEditField_Score_1.Value;

Superconductors = {app.SuperconductorDropDown_Score_1.Value;
    app.SuperconductorDropDown_Score_2.Value;
    app.SuperconductorDropDown_Score_3.Value;
    app.SuperconductorDropDown_Score_4.Value};

% For loop to determine Bsc and Lengths
for i = 1:4

    if i==1
        value = app.SuperconductorDropDown_Score_1.Value;
        % Read rest of values
        app.Bo = app.BTEditField_Score_1.Value;
        app.Tk = app.CoolingTempEditField_Score_1.Value;
        app.g = app.AirgapmmEditField_Score_1.Value;
    elseif i==2
        value = app.SuperconductorDropDown_Score_2.Value;
        % Read rest of values
        app.Bo = app.BTEditField_Score_2.Value;
        app.Tk = app.CoolingTempEditField_Score_2.Value;
        app.g = app.AirgapmmEditField_Score_2.Value;
    elseif i ==3
        value = app.SuperconductorDropDown_Score_3.Value;
        % Read rest of values
        app.Bo = app.BTEditField_Score_3.Value;
        app.Tk = app.CoolingTempEditField_Score_3.Value;
```

```
            app.g = app.AirgapmmEditField_Score_3.Value;
        else
            value = app.SuperconductorDropDown_Score_4.Value;
            % Read rest of values
            app.Bo = app.BTEditField_Score_4.Value;
            app.Tk = app.CoolingTempEditField_Score_4.Value;
            app.g = app.AirgapmmEditField_Score_4.Value;
        end

        % Change type of superconductor and adjust temperature
        Properties_Supercond(app,value);
        % Adjust cooling temperature limits
        [app.Curve,app.T,app.B] = fitCurve(app);

        if app.Curve(app.Rosc*app.Bo,T) >= 0
            DetermineGeometry(app);
            L(i) = app.wire_lenght;
            W(i) = app.W;
            P(i) = app.P;
        else
            L(i) = inf;
            W(i) = inf;
            P(i) = inf;
        end
    end



    % Check sum
    sum = Weight_weight + Price_weight + Length_weight + Man_weight;
    if sum ~= 1
        error('Check that the weights sum is equal to 1')
    end

    % Cost function

    app.score_total = Cyclotron_Cost_function(app, Superconductors, Weight_weight,
        Price_weight,Length_weight,Man_weight,W,P,L);

    % GIve scores
    ScoreDropDown_Score_1ValueChanged(app, event);

end

% Button pushed function: RunQFButton_Score
function RunQFButton_ScorePushed(app, event)
```

```
RunButton_ScorePushed(app, event)
% Read values
% Weights
Weight_weight = app.Performance_Score.Value;
Price_weight = app.Price_Score.Value;
Length_weight = app.Length_Score.Value;
Man_weight = app.Manufacturability_Score.Value;

% Function that determines the geometry of the cyclotron

L = zeros(4,1);
W = zeros(4,1);
P = zeros(4,1);
T = app.CoolingTempEditField_Score_1.Value;

Superconductors = {app.SuperconductorDropDown_Score_1.Value;
    app.SuperconductorDropDown_Score_2.Value;
    app.SuperconductorDropDown_Score_3.Value;
    app.SuperconductorDropDown_Score_4.Value};

% For loop to determine Bsc and Lengths
for i = 1:4

    if i==1
        value = app.SuperconductorDropDown_Score_1.Value;
        % Read rest of values
        app.Bo = app.BTEditField_Score_1.Value;
        app.Tk = app.CoolingTempEditField_Score_1.Value;
        app.g = app.AirgapmmEditField_Score_1.Value;
    elseif i==2
        value = app.SuperconductorDropDown_Score_2.Value;
        % Read rest of values
        app.Bo = app.BTEditField_Score_2.Value;
        app.Tk = app.CoolingTempEditField_Score_2.Value;
        app.g = app.AirgapmmEditField_Score_2.Value;
    elseif i ==3
        value = app.SuperconductorDropDown_Score_3.Value;
        % Read rest of values
        app.Bo = app.BTEditField_Score_3.Value;
        app.Tk = app.CoolingTempEditField_Score_3.Value;
        app.g = app.AirgapmmEditField_Score_3.Value;
    else
        value = app.SuperconductorDropDown_Score_4.Value;
        % Read rest of values
        app.Bo = app.BTEditField_Score_4.Value;
```

```
            app.Tk = app.CoolingTempEditField_Score_4.Value;
            app.g = app.AirgapmmEditField_Score_4.Value;
        end

        % Change type of superconductor and adjust temperature
        Properties_Supercond(app,value);
        % Adjust cooling temperature limits
        [app.Curve,app.T,app.B] = fitCurve(app);

        if app.Curve(app.Rosc*app.Bo,T) >= 0
            DetermineGeometry(app);
            % Run QF
            prb = RunQF(app);
            % Optimize case call function
            OptimizeQF(app,prb);
            closeQF(app,prb);
            L(i) = app.wire_lenght;
            W(i) = app.W;
            P(i) = app.P;
        else
            L(i) = inf;
            W(i) = inf;
            P(i) = inf;
        end
    end

    % Cost function

    app.score_total = Cyclotron_Cost_function(app, Superconductors, Weight_weight,
        Price_weight,Length_weight,Man_weight,W,P,L);

    % GIve scores
    ScoreDropDown_Score_1ValueChanged(app, event);
end

% Value changed function: SteelDropDown_Score
function SteelDropDown_ScoreValueChanged(app, event)
    value = app.SteelDropDown_Score.Value;
    Properties_steel(app,value);
end

% Value changed function: SuperconductorDropDown_Score_1
function SuperconductorDropDown_Score_1ValueChanged(app, event)
    value = app.SuperconductorDropDown_Score_1.Value;
    % Change type of superconductor and adjust temperature
```

```
    Properties_Supercond(app,value);
    fitCurve(app);
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);
    app.CoolingTempEditField_Score_1.Limits = [app.T(1) app.T(end)];
end

% Value changed function: SuperconductorDropDown_Score_2
function SuperconductorDropDown_Score_2ValueChanged(app, event)
    value = app.SuperconductorDropDown_Score_2.Value;
    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    fitCurve(app);
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);
    app.CoolingTempEditField_Score_2.Limits = [app.T(1) app.T(end)];
end

% Value changed function: SuperconductorDropDown_Score_3
function SuperconductorDropDown_Score_3ValueChanged(app, event)
    value = app.SuperconductorDropDown_Score_3.Value;
    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    fitCurve(app);
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);
    app.CoolingTempEditField_Score_2.Limits = [app.T(1) app.T(end)];
end

% Value changed function: SuperconductorDropDown_Score_4
function SuperconductorDropDown_Score_4ValueChanged(app, event)
    value = app.SuperconductorDropDown_Score_4.Value;
    % Change type of superconductor and adjust temperature
    Properties_Supercond(app,value);
    fitCurve(app);
    % Adjust cooling temperature limits
    [app.Curve,app.T,app.B] = fitCurve(app);
    app.CoolingTempEditField_Score_2.Limits = [app.T(1) app.T(end)];
end

% Value changed function: ScoreDropDown_Score_1
function ScoreDropDown_Score_1ValueChanged(app, event)
    % Give Superconductor performance
    type = app.ScoreDropDown_Score_1.Value;
```

```
        for i = 1:4
            [Score, Position] = Score_Position(app,type,i);

            if i==1
                app.ScoreNumber_Score_1.Value = Score;
                app.ScoreNumber2_Score_1.Value = Position;
            elseif i==2
                app.ScoreNumber_Score_2.Value = Score;
                app.ScoreNumber2_Score_2.Value = Position;
            elseif i==3
                app.ScoreNumber_Score_3.Value = Score;
                app.ScoreNumber2_Score_3.Value = Position;
            else
                app.ScoreNumber_Score_4.Value = Score;
                app.ScoreNumber2_Score_4.Value = Position;
            end
        end

    end

    % Value changed function: OnlySuperconductorsCheckBox_Score
    function OnlySuperconductorsCheckBox_ScoreValueChanged(app, event)
        value = app.OnlySuperconductorsCheckBox_Score.Value;

        if value == 1
            Read_superconductor(app,'Superconductores_No_Copper');
        else
            Read_superconductor(app,'Superconductores');
        end
    end

    % Value changed function: OnlyReliableDataCheckBox_Score
    function OnlyReliableDataCheckBox_ScoreValueChanged(app, event)
        value = app.OnlyReliableDataCheckBox_Score.Value;

        if value == 1
            Read_superconductor(app,'Superconductores_Reliable_Data');
        else
            Read_superconductor(app,'Superconductores');
        end
    end
end

% Component initialization
methods (Access = private)
```

```
% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.Color = [1 1 1];
    app.UIFigure.Position = [100 100 685 560];
    app.UIFigure.Name = 'UI Figure';

    % Create TabGroup
    app.TabGroup = uitabgroup(app.UIFigure);
    app.TabGroup.SelectionChangedFcn = createCallbackFcn(app,
        @TabGroupSelectionChanged, true);
    app.TabGroup.Position = [0 2 685 560];

    % Create MaterialTab
    app.MaterialTab = uitab(app.TabGroup);
    app.MaterialTab.Title = 'Material';
    app.MaterialTab.BackgroundColor = [0.9412 0.9412 0.9412];

    % Create MaterialPanel_3
    app.MaterialPanel_3 = uipanel(app.MaterialTab);
    app.MaterialPanel_3.ForegroundColor = [1 1 1];
    app.MaterialPanel_3.Title = 'Material';
    app.MaterialPanel_3.BackgroundColor = [0 0.4471 0.7412];
    app.MaterialPanel_3.Position = [18 457 229 63];

    % Create SuperconductorDropDown_3Label
    app.SuperconductorDropDown_3Label = uilabel(app.MaterialPanel_3);
    app.SuperconductorDropDown_3Label.HorizontalAlignment = 'right';
    app.SuperconductorDropDown_3Label.FontColor = [1 1 1];
    app.SuperconductorDropDown_3Label.Position = [12 13 90 22];
    app.SuperconductorDropDown_3Label.Text = 'Superconductor';

    % Create SuperconductorDropDown_3
    app.SuperconductorDropDown_3 = uidropdown(app.MaterialPanel_3);
    app.SuperconductorDropDown_3.Items = {'NbTi_Bruker', 'NbTi_Supercon',
        'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
        'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
    app.SuperconductorDropDown_3.ValueChangedFcn = createCallbackFcn(app,
        @SuperconductorDropDown_3ValueChanged, true);
    app.SuperconductorDropDown_3.Position = [117 13 100 22];
    app.SuperconductorDropDown_3.Value = 'NbTi_Bruker';
```

```
% Create UIAxes_3
app.UIAxes_3 = uiaxes(app.MaterialTab);
title(app.UIAxes_3, '')
xlabel(app.UIAxes_3, 'X (mm)')
ylabel(app.UIAxes_3, 'Y (mm)')
app.UIAxes_3.AmbientLightColor = [0.9412 0.9412 0.9412];
app.UIAxes_3.PlotBoxAspectRatio = [1.6063829787234 1 1];
app.UIAxes_3.ColorOrder = [0 0.4471 0.7412;0.851 0.3255 0.098;0.9294 0.6941
    0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451 0.9333;0.6392
    0.0784 0.1804];
app.UIAxes_3.GridColor = [0.149 0.149 0.149];
app.UIAxes_3.MinorGridColor = [0.149 0.149 0.149];
app.UIAxes_3.XColor = [0.149 0.149 0.149];
app.UIAxes_3.Color = [0.9412 0.9412 0.9412];
app.UIAxes_3.XGrid = 'on';
app.UIAxes_3.YGrid = 'on';
app.UIAxes_3.View = [120 50];
app.UIAxes_3.BackgroundColor = [0.9412 0.9412 0.9412];
app.UIAxes_3.Position = [15 10 669 440];

% Create AddMaterialButton
app.AddMaterialButton = uibutton(app.MaterialTab, 'push');
app.AddMaterialButton.ButtonPushedFcn = createCallbackFcn(app,
    @AddMaterialButtonPushed, true);
app.AddMaterialButton.BackgroundColor = [0 0.4471 0.7412];
app.AddMaterialButton.FontColor = [1 1 1];
app.AddMaterialButton.Position = [562 498 100 22];
app.AddMaterialButton.Text = 'Add Material';

% Create DataPanel_Material
app.DataPanel_Material = uipanel(app.MaterialTab);
app.DataPanel_Material.ForegroundColor = [1 1 1];
app.DataPanel_Material.Title = 'Data';
app.DataPanel_Material.BackgroundColor = [0 0.4471 0.7412];
app.DataPanel_Material.Position = [18 304 196 146];

% Create BTEditField_2Label_3
app.BTEditField_2Label_3 = uilabel(app.DataPanel_Material);
app.BTEditField_2Label_3.HorizontalAlignment = 'right';
app.BTEditField_2Label_3.FontColor = [1 1 1];
app.BTEditField_2Label_3.Position = [29 95 32 22];
app.BTEditField_2Label_3.Text = 'B (T)';

% Create BT_Material
app.BT_Material = uieditfield(app.DataPanel_Material, 'numeric');
```

```
app.BT_Material.Limits = [0 Inf];
app.BT_Material.ValueChangedFcn = createCallbackFcn(app,
    @BT_MaterialValueChanged, true);
app.BT_Material.Position = [77 95 100 22];
app.BT_Material.Value = 4;

% Create TKEditFieldLabel_2
app.TKEditFieldLabel_2 = uilabel(app.DataPanel_Material);
app.TKEditFieldLabel_2.HorizontalAlignment = 'right';
app.TKEditFieldLabel_2.FontColor = [1 1 1];
app.TKEditFieldLabel_2.Position = [30 68 32 22];
app.TKEditFieldLabel_2.Text = 'T (K)';

% Create TK_Material
app.TK_Material = uieditfield(app.DataPanel_Material, 'numeric');
app.TK_Material.Limits = [0 1000];
app.TK_Material.ValueChangedFcn = createCallbackFcn(app,
    @TK_MaterialValueChanged, true);
app.TK_Material.Position = [77 66 100 22];
app.TK_Material.Value = 4.2;

% Create FillFactorLabel
app.FillFactorLabel = uilabel(app.DataPanel_Material);
app.FillFactorLabel.HorizontalAlignment = 'right';
app.FillFactorLabel.FontColor = [1 1 1];
app.FillFactorLabel.Position = [4 38 58 22];
app.FillFactorLabel.Text = 'Fill Factor';

% Create Fill_Factor
app.Fill_Factor = uieditfield(app.DataPanel_Material, 'numeric');
app.Fill_Factor.Limits = [0 Inf];
app.Fill_Factor.ValueChangedFcn = createCallbackFcn(app,
    @Fill_FactorValueChanged, true);
app.Fill_Factor.Position = [77 37 100 22];

% Create JAm2Label_2
app.JAm2Label_2 = uilabel(app.DataPanel_Material);
app.JAm2Label_2.HorizontalAlignment = 'right';
app.JAm2Label_2.FontColor = [1 1 1];
app.JAm2Label_2.Position = [11 8 51 22];
app.JAm2Label_2.Text = {'J (A/m2)'; ''};

% Create J_Material
app.J_Material = uieditfield(app.DataPanel_Material, 'numeric');
app.J_Material.Limits = [0 Inf];
```

```
app.J_Material.Position = [77 8 100 22];

% Create MaterialComparisonTab
app.MaterialComparisonTab = uitab(app.TabGroup);
app.MaterialComparisonTab.Title = 'Material Comparison';

% Create DataPanel
app.DataPanel = uipanel(app.MaterialComparisonTab);
app.DataPanel.ForegroundColor = [1 1 1];
app.DataPanel.Title = 'Data';
app.DataPanel.BackgroundColor = [0 0.4471 0.7412];
app.DataPanel.Position = [6 390 243 126];

% Create BTEditField_2Label_2
app.BTEditField_2Label_2 = uilabel(app.DataPanel);
app.BTEditField_2Label_2.HorizontalAlignment = 'right';
app.BTEditField_2Label_2.FontColor = [1 1 1];
app.BTEditField_2Label_2.Position = [70 75 32 22];
app.BTEditField_2Label_2.Text = 'B (T)';

% Create BTComparison
app.BTComparison = uieditfield(app.DataPanel, 'numeric');
app.BTComparison.Limits = [0 Inf];
app.BTComparison.Position = [118 75 100 22];
app.BTComparison.Value = 4;

% Create TKEditFieldLabel
app.TKEditFieldLabel = uilabel(app.DataPanel);
app.TKEditFieldLabel.HorizontalAlignment = 'right';
app.TKEditFieldLabel.FontColor = [1 1 1];
app.TKEditFieldLabel.Position = [71 44 32 22];
app.TKEditFieldLabel.Text = 'T (K)';

% Create TKComparison
app.TKComparison = uieditfield(app.DataPanel, 'numeric');
app.TKComparison.Limits = [0 1000];
app.TKComparison.Position = [118 43 100 22];
app.TKComparison.Value = 4.2;

% Create MinLengthmEditFieldLabel
app.MinLengthmEditFieldLabel = uilabel(app.DataPanel);
app.MinLengthmEditFieldLabel.HorizontalAlignment = 'right';
app.MinLengthmEditFieldLabel.FontColor = [1 1 1];
app.MinLengthmEditFieldLabel.Position = [13 12 90 22];
app.MinLengthmEditFieldLabel.Text = 'Min. Length (m)';
```

```
% Create MinLength
app.MinLength = uieditfield(app.DataPanel, 'numeric');
app.MinLength.Limits = [0 Inf];
app.MinLength.Position = [118 12 100 22];
app.MinLength.Value = 3000;

% Create WeightsPanel
app.WeightsPanel = uipanel(app.MaterialComparisonTab);
app.WeightsPanel.ForegroundColor = [1 1 1];
app.WeightsPanel.BorderType = 'none';
app.WeightsPanel.Title = 'Weights';
app.WeightsPanel.BackgroundColor = [0 0.4471 0.7412];
app.WeightsPanel.Position = [282 419 379 97];

% Create PerfPriceLabel
app.PerfPriceLabel = uilabel(app.WeightsPanel);
app.PerfPriceLabel.HorizontalAlignment = 'right';
app.PerfPriceLabel.FontColor = [1 1 1];
app.PerfPriceLabel.Position = [38 47 58 22];
app.PerfPriceLabel.Text = 'Perf/Price';

% Create Performance
app.Performance = uieditfield(app.WeightsPanel, 'numeric');
app.Performance.Limits = [0 1];
app.Performance.Position = [111 47 100 22];
app.Performance.Value = 0.6;

% Create LenghtLabel
app.LenghtLabel = uilabel(app.WeightsPanel);
app.LenghtLabel.HorizontalAlignment = 'right';
app.LenghtLabel.FontColor = [1 1 1];
app.LenghtLabel.Position = [211 48 42 22];
app.LenghtLabel.Text = 'Lenght';

% Create Length
app.Length = uieditfield(app.WeightsPanel, 'numeric');
app.Length.Limits = [0 1];
app.Length.Position = [267 47 100 22];
app.Length.Value = 0.2;

% Create ManufacturabilityEditFieldLabel_2
app.ManufacturabilityEditFieldLabel_2 = uilabel(app.WeightsPanel);
app.ManufacturabilityEditFieldLabel_2.HorizontalAlignment = 'right';
app.ManufacturabilityEditFieldLabel_2.FontColor = [1 1 1];
```

```
app.ManufacturabilityEditFieldLabel_2.Position = [2 15 96 22];
app.ManufacturabilityEditFieldLabel_2.Text = 'Manufacturability';

% Create Manufacturability
app.Manufacturability = uieditfield(app.WeightsPanel, 'numeric');
app.Manufacturability.Limits = [0 1];
app.Manufacturability.Position = [112 15 100 22];
app.Manufacturability.Value = 0.2;

% Create MaterialPanelComparison
app.MaterialPanelComparison = uipanel(app.MaterialComparisonTab);
app.MaterialPanelComparison.ForegroundColor = [1 1 1];
app.MaterialPanelComparison.Title = 'Material';
app.MaterialPanelComparison.BackgroundColor = [0 0.4471 0.7412];
app.MaterialPanelComparison.Position = [282 328 129 63];

% Create SuperconductorDropDownComparison
app.SuperconductorDropDownComparison = uidropdown(app.MaterialPanelComparison);
app.SuperconductorDropDownComparison.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDownComparison.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDownComparisonValueChanged, true);
app.SuperconductorDropDownComparison.Position = [15 10 100 22];
app.SuperconductorDropDownComparison.Value = 'NbTi_Bruker';

% Create UIAxesComparison
app.UIAxesComparison = uiaxes(app.MaterialComparisonTab);
title(app.UIAxesComparison, '')
xlabel(app.UIAxesComparison, 'X (mm)')
ylabel(app.UIAxesComparison, 'Y (mm)')
app.UIAxesComparison.AmbientLightColor = [0.9412 0.9412 0.9412];
app.UIAxesComparison.PlotBoxAspectRatio = [1.6063829787234 1 1];
app.UIAxesComparison.ColorOrder = [0 0.4471 0.7412;0.851 0.3255 0.098;0.9294
    0.6941 0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451
    0.9333;0.6392 0.0784 0.1804];
app.UIAxesComparison.GridColor = [0.149 0.149 0.149];
app.UIAxesComparison.MinorGridColor = [0.149 0.149 0.149];
app.UIAxesComparison.XColor = [0.149 0.149 0.149];
app.UIAxesComparison.Color = [0.9412 0.9412 0.9412];
app.UIAxesComparison.XGrid = 'on';
app.UIAxesComparison.YGrid = 'on';
app.UIAxesComparison.View = [120 50];
app.UIAxesComparison.BackgroundColor = [0.9412 0.9412 0.9412];
app.UIAxesComparison.Position = [6 1 462 317];
```

```
% Create ScoresPanel
app.ScoresPanel = uipanel(app.MaterialComparisonTab);
app.ScoresPanel.ForegroundColor = [1 1 1];
app.ScoresPanel.BorderType = 'none';
app.ScoresPanel.Title = 'Scores';
app.ScoresPanel.BackgroundColor = [0 0.4471 0.7412];
app.ScoresPanel.Position = [425 329 236 62];

% Create ScoreNumber
app.ScoreNumber = uieditfield(app.ScoresPanel, 'numeric');
app.ScoreNumber.Limits = [0 10];
app.ScoreNumber.Position = [125 12 42 22];
app.ScoreNumber.Value = 5;

% Create ScoreDropDown
app.ScoreDropDown = uidropdown(app.ScoresPanel);
app.ScoreDropDown.Items = {'Total Score', 'Perf/Price', 'Manufacturability',
    'Length'};
app.ScoreDropDown.ValueChangedFcn = createCallbackFcn(app,
    @ScoreDropDownValueChanged, true);
app.ScoreDropDown.Position = [8 12 100 22];
app.ScoreDropDown.Value = 'Total Score';

% Create ScoreNumber_2
app.ScoreNumber_2 = uieditfield(app.ScoresPanel, 'numeric');
app.ScoreNumber_2.Limits = [0 10];
app.ScoreNumber_2.Position = [178 12 42 22];
app.ScoreNumber_2.Value = 5;

% Create RankingsPanel
app.RankingsPanel = uipanel(app.MaterialComparisonTab);
app.RankingsPanel.ForegroundColor = [1 1 1];
app.RankingsPanel.Title = 'Rankings';
app.RankingsPanel.BackgroundColor = [0 0.4471 0.7412];
app.RankingsPanel.Position = [425 174 236 144];

% Create Label
app.Label = uilabel(app.RankingsPanel);
app.Label.HorizontalAlignment = 'right';
app.Label.FontWeight = 'bold';
app.Label.FontAngle = 'italic';
app.Label.FontColor = [1 1 1];
app.Label.Position = [8 87 25 22];
app.Label.Text = '1';
```

```
% Create Name1
app.Name1 = uieditfield(app.RankingsPanel, 'text');
app.Name1.Position = [48 87 100 22];

% Create Label_2
app.Label_2 = uilabel(app.RankingsPanel);
app.Label_2.HorizontalAlignment = 'right';
app.Label_2.FontWeight = 'bold';
app.Label_2.FontAngle = 'italic';
app.Label_2.FontColor = [1 1 1];
app.Label_2.Position = [8 51 25 22];
app.Label_2.Text = '2';

% Create Name2
app.Name2 = uieditfield(app.RankingsPanel, 'text');
app.Name2.Position = [48 51 100 22];

% Create Label_3
app.Label_3 = uilabel(app.RankingsPanel);
app.Label_3.HorizontalAlignment = 'right';
app.Label_3.FontWeight = 'bold';
app.Label_3.FontAngle = 'italic';
app.Label_3.FontColor = [1 1 1];
app.Label_3.Position = [8 15 25 22];
app.Label_3.Text = '3';

% Create Name3
app.Name3 = uieditfield(app.RankingsPanel, 'text');
app.Name3.Position = [48 15 100 22];

% Create Score1
app.Score1 = uieditfield(app.RankingsPanel, 'numeric');
app.Score1.ValueDisplayFormat = '%11.2g';
app.Score1.Position = [175 87 42 22];

% Create Score2
app.Score2 = uieditfield(app.RankingsPanel, 'numeric');
app.Score2.ValueDisplayFormat = '%11.2g';
app.Score2.Position = [175 51 43 22];

% Create Score3
app.Score3 = uieditfield(app.RankingsPanel, 'numeric');
app.Score3.ValueDisplayFormat = '%11.2g';
app.Score3.Position = [175 15 44 22];
```

```
% Create RunButtonComparison
app.RunButtonComparison = uibutton(app.MaterialComparisonTab, 'push');
app.RunButtonComparison.ButtonPushedFcn = createCallbackFcn(app,
    @RunButtonComparisonPushed, true);
app.RunButtonComparison.BackgroundColor = [0 0.4471 0.7412];
app.RunButtonComparison.FontColor = [1 1 1];
app.RunButtonComparison.Position = [149 344 100 22];
app.RunButtonComparison.Text = 'Run';

% Create OnlySuperconductorsCheckBox
app.OnlySuperconductorsCheckBox = uicheckbox(app.MaterialComparisonTab);
app.OnlySuperconductorsCheckBox.ValueChangedFcn = createCallbackFcn(app,
    @OnlySuperconductorsCheckBoxValueChanged, true);
app.OnlySuperconductorsCheckBox.Text = 'Only Superconductors';
app.OnlySuperconductorsCheckBox.Position = [6 359 141 22];

% Create OnlyReliableDataCheckBox
app.OnlyReliableDataCheckBox = uicheckbox(app.MaterialComparisonTab);
app.OnlyReliableDataCheckBox.ValueChangedFcn = createCallbackFcn(app,
    @OnlyReliableDataCheckBoxValueChanged, true);
app.OnlyReliableDataCheckBox.Text = 'Only Reliable Data';
app.OnlyReliableDataCheckBox.Position = [6 329 122 17];

% Create CyclotronTab
app.CyclotronTab = uitab(app.TabGroup);
app.CyclotronTab.Title = 'Cyclotron';
app.CyclotronTab.BackgroundColor = [0.9412 0.9412 0.9412];

% Create InitialDataPanel
app.InitialDataPanel = uipanel(app.CyclotronTab);
app.InitialDataPanel.ForegroundColor = [1 1 1];
app.InitialDataPanel.Title = 'Initial Data';
app.InitialDataPanel.BackgroundColor = [0 0.4471 0.7412];
app.InitialDataPanel.Position = [12 353 229 169];

% Create EMeVLabel
app.EMeVLabel = uilabel(app.InitialDataPanel);
app.EMeVLabel.HorizontalAlignment = 'right';
app.EMeVLabel.FontColor = [1 1 1];
app.EMeVLabel.Position = [53 115 50 22];
app.EMeVLabel.Text = 'E (MeV)';

% Create EMeVEditField
app.EMeVEditField = uieditfield(app.InitialDataPanel, 'numeric');
```

```
app.EMeVEditField.Limits = [0 Inf];
app.EMeVEditField.Position = [118 115 100 22];
app.EMeVEditField.Value = 8.5;

% Create BTLabel
app.BTLabel = uilabel(app.InitialDataPanel);
app.BTLabel.HorizontalAlignment = 'right';
app.BTLabel.FontColor = [1 1 1];
app.BTLabel.Position = [71 82 32 22];
app.BTLabel.Text = 'B (T)';

% Create BTEditField
app.BTEditField = uieditfield(app.InitialDataPanel, 'numeric');
app.BTEditField.Limits = [0 Inf];
app.BTEditField.Position = [118 82 100 22];
app.BTEditField.Value = 4;

% Create WorkingPointEditFieldLabel
app.WorkingPointEditFieldLabel = uilabel(app.InitialDataPanel);
app.WorkingPointEditFieldLabel.HorizontalAlignment = 'right';
app.WorkingPointEditFieldLabel.FontColor = [1 1 1];
app.WorkingPointEditFieldLabel.Position = [23 46 80 22];
app.WorkingPointEditFieldLabel.Text = 'Working Point';

% Create WorkingPointEditField
app.WorkingPointEditField = uieditfield(app.InitialDataPanel, 'numeric');
app.WorkingPointEditField.Limits = [0 1];
app.WorkingPointEditField.Position = [118 46 100 22];
app.WorkingPointEditField.Value = 0.65;

% Create WireCurentAEditFieldLabel
app.WireCurentAEditFieldLabel = uilabel(app.InitialDataPanel);
app.WireCurentAEditFieldLabel.HorizontalAlignment = 'right';
app.WireCurentAEditFieldLabel.FontColor = [1 1 1];
app.WireCurentAEditFieldLabel.Position = [14 10 89 22];
app.WireCurentAEditFieldLabel.Text = 'Wire Curent (A)';

% Create WireCurentAEditField
app.WireCurentAEditField = uieditfield(app.InitialDataPanel, 'numeric');
app.WireCurentAEditField.Limits = [0 Inf];
app.WireCurentAEditField.Position = [118 10 100 22];
app.WireCurentAEditField.Value = 100;

% Create GeometricDimensionsPanel
app.GeometricDimensionsPanel = uipanel(app.CyclotronTab);
```

```
app.GeometricDimensionsPanel.ForegroundColor = [1 1 1];
app.GeometricDimensionsPanel.BorderType = 'none';
app.GeometricDimensionsPanel.Title = 'Geometric Dimensions';
app.GeometricDimensionsPanel.BackgroundColor = [0 0.4471 0.7412];
app.GeometricDimensionsPanel.Position = [12 248 229 98];

% Create AirgapmmEditFieldLabel
app.AirgapmmEditFieldLabel = uilabel(app.GeometricDimensionsPanel);
app.AirgapmmEditFieldLabel.HorizontalAlignment = 'right';
app.AirgapmmEditFieldLabel.FontColor = [1 1 1];
app.AirgapmmEditFieldLabel.Position = [32 48 71 22];
app.AirgapmmEditFieldLabel.Text = 'Airgap (mm)';

% Create AirgapmmEditField
app.AirgapmmEditField = uieditfield(app.GeometricDimensionsPanel, 'numeric');
app.AirgapmmEditField.Limits = [0 Inf];
app.AirgapmmEditField.Position = [118 48 100 22];
app.AirgapmmEditField.Value = 74;

% Create CryostatmmEditFieldLabel
app.CryostatmmEditFieldLabel = uilabel(app.GeometricDimensionsPanel);
app.CryostatmmEditFieldLabel.HorizontalAlignment = 'right';
app.CryostatmmEditFieldLabel.FontColor = [1 1 1];
app.CryostatmmEditFieldLabel.Position = [23 13 81 22];
app.CryostatmmEditFieldLabel.Text = 'Cryostat (mm)';

% Create CryostatmmEditField
app.CryostatmmEditField = uieditfield(app.GeometricDimensionsPanel, 'numeric');
app.CryostatmmEditField.Limits = [0 Inf];
app.CryostatmmEditField.Position = [118 13 100 22];
app.CryostatmmEditField.Value = 32;

% Create RunButton
app.RunButton = uibutton(app.CyclotronTab, 'push');
app.RunButton.ButtonPushedFcn = createCallbackFcn(app, @RunButtonPushed, true);
app.RunButton.BackgroundColor = [0 0.4471 0.7412];
app.RunButton.FontColor = [1 1 1];
app.RunButton.Position = [300 489 100 22];
app.RunButton.Text = 'Run';

% Create RunQFButton
app.RunQFButton = uibutton(app.CyclotronTab, 'push');
app.RunQFButton.ButtonPushedFcn = createCallbackFcn(app, @RunQFButtonPushed,
    true);
app.RunQFButton.BackgroundColor = [0 0.4471 0.7412];
```

```
app.RunQFButton.FontColor = [1 1 1];
app.RunQFButton.Position = [542 489 100 22];
app.RunQFButton.Text = 'Run QF';

% Create UIAxes
app.UIAxes = uiaxes(app.CyclotronTab);
title(app.UIAxes, 'Cyclotron Dimensions')
xlabel(app.UIAxes, 'X (mm)')
ylabel(app.UIAxes, 'Y (mm)')
app.UIAxes.AmbientLightColor = [0.9412 0.9412 0.9412];
app.UIAxes.PlotBoxAspectRatio = [1.61870503597122 1 1];
app.UIAxes.GridColor = [0.149 0.149 0.149];
app.UIAxes.MinorGridColor = [0.149 0.149 0.149];
app.UIAxes.XColor = [0.149 0.149 0.149];
app.UIAxes.Color = [0.9412 0.9412 0.9412];
app.UIAxes.XGrid = 'on';
app.UIAxes.YGrid = 'on';
app.UIAxes.BackgroundColor = [0.9412 0.9412 0.9412];
app.UIAxes.Position = [259 188 410 281];

% Create MaterialPanel
app.MaterialPanel = uipanel(app.CyclotronTab);
app.MaterialPanel.ForegroundColor = [1 1 1];
app.MaterialPanel.Title = 'Material';
app.MaterialPanel.BackgroundColor = [0 0.4471 0.7412];
app.MaterialPanel.Position = [12 31 229 207];

% Create SteelDropDownLabel
app.SteelDropDownLabel = uilabel(app.MaterialPanel);
app.SteelDropDownLabel.HorizontalAlignment = 'right';
app.SteelDropDownLabel.FontColor = [1 1 1];
app.SteelDropDownLabel.Position = [70 153 33 22];
app.SteelDropDownLabel.Text = 'Steel';

% Create SteelDropDown
app.SteelDropDown = uidropdown(app.MaterialPanel);
app.SteelDropDown.Items = {'AISI 1010', 'AISI 1020', 'ARMCO'};
app.SteelDropDown.ValueChangedFcn = createCallbackFcn(app,
    @SteelDropDownValueChanged, true);
app.SteelDropDown.Position = [118 153 100 22];
app.SteelDropDown.Value = 'AISI 1010';

% Create CopperSectionEditField_2Label
app.CopperSectionEditField_2Label = uilabel(app.MaterialPanel);
app.CopperSectionEditField_2Label.HorizontalAlignment = 'right';
```

```
app.CopperSectionEditField_2Label.FontColor = [1 1 1];
app.CopperSectionEditField_2Label.Position = [16 48 88 22];
app.CopperSectionEditField_2Label.Text = 'Copper Section';

% Create CopperSectionEditField
app.CopperSectionEditField = uieditfield(app.MaterialPanel, 'numeric');
app.CopperSectionEditField.Limits = [0 Inf];
app.CopperSectionEditField.Position = [118 48 100 22];
app.CopperSectionEditField.Value = 4.5;

% Create AspectRatioEditFieldLabel
app.AspectRatioEditFieldLabel = uilabel(app.MaterialPanel);
app.AspectRatioEditFieldLabel.HorizontalAlignment = 'right';
app.AspectRatioEditFieldLabel.FontColor = [1 1 1];
app.AspectRatioEditFieldLabel.Position = [30 14 74 22];
app.AspectRatioEditFieldLabel.Text = 'Aspect Ratio';

% Create AspectRatioEditField
app.AspectRatioEditField = uieditfield(app.MaterialPanel, 'numeric');
app.AspectRatioEditField.Limits = [0 Inf];
app.AspectRatioEditField.Position = [118 14 100 22];
app.AspectRatioEditField.Value = 1;

% Create CoolingTempEditFieldLabel
app.CoolingTempEditFieldLabel = uilabel(app.MaterialPanel);
app.CoolingTempEditFieldLabel.HorizontalAlignment = 'right';
app.CoolingTempEditFieldLabel.FontColor = [1 1 1];
app.CoolingTempEditFieldLabel.Position = [22 83 82 22];
app.CoolingTempEditFieldLabel.Text = 'Cooling Temp.';

% Create CoolingTempEditField
app.CoolingTempEditField = uieditfield(app.MaterialPanel, 'numeric');
app.CoolingTempEditField.Limits = [2 9];
app.CoolingTempEditField.ValueChangedFcn = createCallbackFcn(app,
    @CoolingTempEditFieldValueChanged, true);
app.CoolingTempEditField.Position = [118 83 100 22];
app.CoolingTempEditField.Value = 4.2;

% Create SuperconductorDropDown_4Label
app.SuperconductorDropDown_4Label = uilabel(app.MaterialPanel);
app.SuperconductorDropDown_4Label.HorizontalAlignment = 'right';
app.SuperconductorDropDown_4Label.FontColor = [1 1 1];
app.SuperconductorDropDown_4Label.Position = [12 118 90 22];
app.SuperconductorDropDown_4Label.Text = 'Superconductor';
```

```
% Create SuperconductorDropDown
app.SuperconductorDropDown = uidropdown(app.MaterialPanel);
app.SuperconductorDropDown.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDown.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDownValueChanged, true);
app.SuperconductorDropDown.Position = [117 118 100 22];
app.SuperconductorDropDown.Value = 'NbTi_Bruker';

% Create OutputdataPanel
app.OutputdataPanel = uipanel(app.CyclotronTab);
app.OutputdataPanel.ForegroundColor = [1 1 1];
app.OutputdataPanel.BorderType = 'none';
app.OutputdataPanel.Title = 'Output data';
app.OutputdataPanel.BackgroundColor = [0 0.4471 0.7412];
app.OutputdataPanel.Position = [259 31 410 140];

% Create WireDiammmEditFieldLabel
app.WireDiammmEditFieldLabel = uilabel(app.OutputdataPanel);
app.WireDiammmEditFieldLabel.HorizontalAlignment = 'right';
app.WireDiammmEditFieldLabel.FontColor = [1 1 1];
app.WireDiammmEditFieldLabel.Position = [192 84 99 22];
app.WireDiammmEditFieldLabel.Text = 'Wire Diam. (mm) ';

% Create WireDiammmEditField
app.WireDiammmEditField = uieditfield(app.OutputdataPanel, 'numeric');
app.WireDiammmEditField.ValueDisplayFormat = '%.3f';
app.WireDiammmEditField.ValueChangedFcn = createCallbackFcn(app,
    @WireDiammmEditFieldValueChanged, true);
app.WireDiammmEditField.Position = [304 84 100 22];

% Create WireLenghtmEditFieldLabel
app.WireLenghtmEditFieldLabel = uilabel(app.OutputdataPanel);
app.WireLenghtmEditFieldLabel.HorizontalAlignment = 'right';
app.WireLenghtmEditFieldLabel.FontColor = [1 1 1];
app.WireLenghtmEditFieldLabel.Position = [185 49 103 22];
app.WireLenghtmEditFieldLabel.Text = 'Wire Lenght (m)';

% Create WireLenghtmEditField
app.WireLenghtmEditField = uieditfield(app.OutputdataPanel, 'numeric');
app.WireLenghtmEditField.ValueDisplayFormat = '%.1f';
app.WireLenghtmEditField.Position = [303 50 100 22];

% Create WeightkgEditFieldLabel
```

```
app.WeightkgEditFieldLabel = uilabel(app.OutputdataPanel);
app.WeightkgEditFieldLabel.HorizontalAlignment = 'right';
app.WeightkgEditFieldLabel.FontColor = [1 1 1];
app.WeightkgEditFieldLabel.Position = [1 84 67 22];
app.WeightkgEditFieldLabel.Text = 'Weight (kg)';

% Create WeightkgEditField
app.WeightkgEditField = uieditfield(app.OutputdataPanel, 'numeric');
app.WeightkgEditField.ValueDisplayFormat = '%.1f';
app.WeightkgEditField.Position = [83 84 100 22];

% Create PriceLabel
app.PriceLabel = uilabel(app.OutputdataPanel);
app.PriceLabel.HorizontalAlignment = 'right';
app.PriceLabel.FontColor = [1 1 1];
app.PriceLabel.Position = [17 49 51 22];
app.PriceLabel.Text = 'Price (ÿ)';

% Create PriceEditField
app.PriceEditField = uieditfield(app.OutputdataPanel, 'numeric');
app.PriceEditField.ValueDisplayFormat = '%.1f';
app.PriceEditField.Position = [83 49 100 22];

% Create WireAreamm2Label
app.WireAreamm2Label = uilabel(app.OutputdataPanel);
app.WireAreamm2Label.HorizontalAlignment = 'right';
app.WireAreamm2Label.FontColor = [1 1 1];
app.WireAreamm2Label.Position = [193 15 97 22];
app.WireAreamm2Label.Text = 'Wire Area (mm2)';

% Create WireAreammEditField
app.WireAreammEditField = uieditfield(app.OutputdataPanel, 'numeric');
app.WireAreammEditField.ValueDisplayFormat = '%.1f';
app.WireAreammEditField.ValueChangedFcn = createCallbackFcn(app,
    @WireAreammEditFieldValueChanged, true);
app.WireAreammEditField.Position = [305 15 100 22];

% Create DiametermLabel
app.DiametermLabel = uilabel(app.OutputdataPanel);
app.DiametermLabel.HorizontalAlignment = 'right';
app.DiametermLabel.FontColor = [1 1 1];
app.DiametermLabel.Position = [0 15 75 22];
app.DiametermLabel.Text = 'Diameter (m)';

% Create DiametermEditField_Custom
```

```
app.DiametermEditField_Custom = uieditfield(app.OutputdataPanel, 'numeric');
app.DiametermEditField_Custom.ValueDisplayFormat = '%.2f';
app.DiametermEditField_Custom.Position = [83 15 100 22];

% Create CyclotronScoreTab
app.CyclotronScoreTab = uitab(app.TabGroup);
app.CyclotronScoreTab.Title = 'Cyclotron Score';
app.CyclotronScoreTab.BackgroundColor = [0.9412 0.9412 0.9412];

% Create InitialDataPanel_Comparison
app.InitialDataPanel_Comparison = uipanel(app.CyclotronScoreTab);
app.InitialDataPanel_Comparison.ForegroundColor = [1 1 1];
app.InitialDataPanel_Comparison.Title = 'Initial Data';
app.InitialDataPanel_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.InitialDataPanel_Comparison.Position = [12 290 229 240];

% Create EMeVEditField_2Label_2
app.EMeVEditField_2Label_2 = uilabel(app.InitialDataPanel_Comparison);
app.EMeVEditField_2Label_2.HorizontalAlignment = 'right';
app.EMeVEditField_2Label_2.FontColor = [1 1 1];
app.EMeVEditField_2Label_2.Position = [52 185 50 22];
app.EMeVEditField_2Label_2.Text = 'E (MeV)';

% Create EMeVEditField_Comparison
app.EMeVEditField_Comparison = uieditfield(app.InitialDataPanel_Comparison,
    'numeric');
app.EMeVEditField_Comparison.Limits = [0 Inf];
app.EMeVEditField_Comparison.Position = [117 185 100 22];
app.EMeVEditField_Comparison.Value = 8.5;

% Create BTEditField_3Label
app.BTEditField_3Label = uilabel(app.InitialDataPanel_Comparison);
app.BTEditField_3Label.HorizontalAlignment = 'right';
app.BTEditField_3Label.FontColor = [1 1 1];
app.BTEditField_3Label.Position = [70 150 32 22];
app.BTEditField_3Label.Text = 'B (T)';

% Create BTEditField_Comparison
app.BTEditField_Comparison = uieditfield(app.InitialDataPanel_Comparison,
    'numeric');
app.BTEditField_Comparison.Limits = [0 Inf];
app.BTEditField_Comparison.Position = [117 150 100 22];
app.BTEditField_Comparison.Value = 4;

% Create WorkingPointEditField_2Label_2
```

```
app.WorkingPointEditField_2Label_2 = uilabel(app.InitialDataPanel_Comparison);
app.WorkingPointEditField_2Label_2.HorizontalAlignment = 'right';
app.WorkingPointEditField_2Label_2.FontColor = [1 1 1];
app.WorkingPointEditField_2Label_2.Position = [22 81 80 22];
app.WorkingPointEditField_2Label_2.Text = 'Working Point';

% Create WorkingPointEditField_Comparison
app.WorkingPointEditField_Comparison =
    uieditfield(app.InitialDataPanel_Comparison, 'numeric');
app.WorkingPointEditField_Comparison.Limits = [0 1];
app.WorkingPointEditField_Comparison.Position = [117 81 100 22];
app.WorkingPointEditField_Comparison.Value = 0.65;

% Create WireCurentAEditField_2Label
app.WireCurentAEditField_2Label = uilabel(app.InitialDataPanel_Comparison);
app.WireCurentAEditField_2Label.HorizontalAlignment = 'right';
app.WireCurentAEditField_2Label.FontColor = [1 1 1];
app.WireCurentAEditField_2Label.Position = [13 47 89 22];
app.WireCurentAEditField_2Label.Text = 'Wire Curent (A)';

% Create WireCurentAEditField_Comparison
app.WireCurentAEditField_Comparison =
    uieditfield(app.InitialDataPanel_Comparison, 'numeric');
app.WireCurentAEditField_Comparison.Limits = [0 Inf];
app.WireCurentAEditField_Comparison.Position = [117 47 100 22];
app.WireCurentAEditField_Comparison.Value = 100;

% Create CoolingTempEditField_2Label
app.CoolingTempEditField_2Label = uilabel(app.InitialDataPanel_Comparison);
app.CoolingTempEditField_2Label.HorizontalAlignment = 'right';
app.CoolingTempEditField_2Label.FontColor = [1 1 1];
app.CoolingTempEditField_2Label.Position = [21 115 82 22];
app.CoolingTempEditField_2Label.Text = 'Cooling Temp.';

% Create CoolingTempEditField_Comparison
app.CoolingTempEditField_Comparison =
    uieditfield(app.InitialDataPanel_Comparison, 'numeric');
app.CoolingTempEditField_Comparison.Limits = [0 Inf];
app.CoolingTempEditField_Comparison.Position = [117 115 100 22];
app.CoolingTempEditField_Comparison.Value = 4.2;

% Create SteelDropDown_2Label_2
app.SteelDropDown_2Label_2 = uilabel(app.InitialDataPanel_Comparison);
app.SteelDropDown_2Label_2.HorizontalAlignment = 'right';
app.SteelDropDown_2Label_2.FontColor = [1 1 1];
```

```
app.SteelDropDown_2Label_2.Position = [69 13 33 22];
app.SteelDropDown_2Label_2.Text = 'Steel';

% Create SteelDropDown_Comparison
app.SteelDropDown_Comparison = uidropdown(app.InitialDataPanel_Comparison);
app.SteelDropDown_Comparison.Items = {'AISI 1010', 'AISI 1020', 'ARMCO'};
app.SteelDropDown_Comparison.ValueChangedFcn = createCallbackFcn(app,
    @SteelDropDown_ComparisonValueChanged, true);
app.SteelDropDown_Comparison.Position = [117 13 100 22];
app.SteelDropDown_Comparison.Value = 'AISI 1010';

% Create GeometricDimensionsPanel_Comparison
app.GeometricDimensionsPanel_Comparison = uipanel(app.CyclotronScoreTab);
app.GeometricDimensionsPanel_Comparison.ForegroundColor = [1 1 1];
app.GeometricDimensionsPanel_Comparison.BorderType = 'none';
app.GeometricDimensionsPanel_Comparison.Title = 'Geometric Dimensions';
app.GeometricDimensionsPanel_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.GeometricDimensionsPanel_Comparison.Position = [13 156 229 129];

% Create AirgapmmEditField_2Label_2
app.AirgapmmEditField_2Label_2 = uilabel(app.GeometricDimensionsPanel_Comparison);
app.AirgapmmEditField_2Label_2.HorizontalAlignment = 'right';
app.AirgapmmEditField_2Label_2.FontColor = [1 1 1];
app.AirgapmmEditField_2Label_2.Position = [32 79 71 22];
app.AirgapmmEditField_2Label_2.Text = 'Airgap (mm)';

% Create AirgapmmEditField_Comparison
app.AirgapmmEditField_Comparison =
    uieditfield(app.GeometricDimensionsPanel_Comparison, 'numeric');
app.AirgapmmEditField_Comparison.Limits = [0 Inf];
app.AirgapmmEditField_Comparison.Position = [118 79 100 22];
app.AirgapmmEditField_Comparison.Value = 74;

% Create CryostatmmEditField_2Label_2
app.CryostatmmEditField_2Label_2 =
    uilabel(app.GeometricDimensionsPanel_Comparison);
app.CryostatmmEditField_2Label_2.HorizontalAlignment = 'right';
app.CryostatmmEditField_2Label_2.FontColor = [1 1 1];
app.CryostatmmEditField_2Label_2.Position = [23 44 81 22];
app.CryostatmmEditField_2Label_2.Text = 'Cryostat (mm)';

% Create CryostatmmEditField_Comparison
app.CryostatmmEditField_Comparison =
    uieditfield(app.GeometricDimensionsPanel_Comparison, 'numeric');
app.CryostatmmEditField_Comparison.Limits = [0 Inf];
```

```
app.CryostatmmEditField_Comparison.Position = [118 44 100 22];
app.CryostatmmEditField_Comparison.Value = 32;

% Create AspectRatioEditField_2Label_2
app.AspectRatioEditField_2Label_2 =
    uilabel(app.GeometricDimensionsPanel_Comparison);
app.AspectRatioEditField_2Label_2.HorizontalAlignment = 'right';
app.AspectRatioEditField_2Label_2.FontColor = [1 1 1];
app.AspectRatioEditField_2Label_2.Position = [31 10 74 22];
app.AspectRatioEditField_2Label_2.Text = 'Aspect Ratio';

% Create AspectRatioEditField_Comparison
app.AspectRatioEditField_Comparison =
    uieditfield(app.GeometricDimensionsPanel_Comparison, 'numeric');
app.AspectRatioEditField_Comparison.Limits = [0 Inf];
app.AspectRatioEditField_Comparison.Position = [119 10 100 22];
app.AspectRatioEditField_Comparison.Value = 1;

% Create RunButton_Comparison
app.RunButton_Comparison = uibutton(app.CyclotronScoreTab, 'push');
app.RunButton_Comparison.ButtonPushedFcn = createCallbackFcn(app,
    @RunButton_ComparisonPushed, true);
app.RunButton_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.RunButton_Comparison.FontColor = [1 1 1];
app.RunButton_Comparison.Position = [265 488 100 22];
app.RunButton_Comparison.Text = 'Run';

% Create RunQFButton_Comparison
app.RunQFButton_Comparison = uibutton(app.CyclotronScoreTab, 'push');
app.RunQFButton_Comparison.ButtonPushedFcn = createCallbackFcn(app,
    @RunQFButton_ComparisonPushed, true);
app.RunQFButton_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.RunQFButton_Comparison.FontColor = [1 1 1];
app.RunQFButton_Comparison.Position = [439 488 100 22];
app.RunQFButton_Comparison.Text = 'Run QF';

% Create RankingsPanel_Comparison
app.RankingsPanel_Comparison = uipanel(app.CyclotronScoreTab);
app.RankingsPanel_Comparison.ForegroundColor = [1 1 1];
app.RankingsPanel_Comparison.Title = 'Rankings';
app.RankingsPanel_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.RankingsPanel_Comparison.Position = [265 77 225 144];

% Create Label_4
app.Label_4 = uilabel(app.RankingsPanel_Comparison);
```

```
app.Label_4.HorizontalAlignment = 'right';
app.Label_4.FontWeight = 'bold';
app.Label_4.FontAngle = 'italic';
app.Label_4.FontColor = [1 1 1];
app.Label_4.Position = [8 87 25 22];
app.Label_4.Text = '1';

% Create Name1_Comparison
app.Name1_Comparison = uieditfield(app.RankingsPanel_Comparison, 'text');
app.Name1_Comparison.Position = [48 87 100 22];

% Create Label_5
app.Label_5 = uilabel(app.RankingsPanel_Comparison);
app.Label_5.HorizontalAlignment = 'right';
app.Label_5.FontWeight = 'bold';
app.Label_5.FontAngle = 'italic';
app.Label_5.FontColor = [1 1 1];
app.Label_5.Position = [8 51 25 22];
app.Label_5.Text = '2';

% Create Name2_Comparison
app.Name2_Comparison = uieditfield(app.RankingsPanel_Comparison, 'text');
app.Name2_Comparison.Position = [48 51 100 22];

% Create Label_6
app.Label_6 = uilabel(app.RankingsPanel_Comparison);
app.Label_6.HorizontalAlignment = 'right';
app.Label_6.FontWeight = 'bold';
app.Label_6.FontAngle = 'italic';
app.Label_6.FontColor = [1 1 1];
app.Label_6.Position = [8 15 25 22];
app.Label_6.Text = '3';

% Create Name3_Comparison
app.Name3_Comparison = uieditfield(app.RankingsPanel_Comparison, 'text');
app.Name3_Comparison.Position = [48 15 100 22];

% Create Score1_Comparison
app.Score1_Comparison = uieditfield(app.RankingsPanel_Comparison, 'numeric');
app.Score1_Comparison.ValueDisplayFormat = '%11.2g';
app.Score1_Comparison.Position = [175 87 39 22];

% Create Score2_Comparison
app.Score2_Comparison = uieditfield(app.RankingsPanel_Comparison, 'numeric');
app.Score2_Comparison.ValueDisplayFormat = '%11.2g';
```

```
app.Score2_Comparison.Position = [175 51 39 22];

% Create Score3_Comparison
app.Score3_Comparison = uieditfield(app.RankingsPanel_Comparison, 'numeric');
app.Score3_Comparison.ValueDisplayFormat = '%11.2g';
app.Score3_Comparison.Position = [175 15 39 22];

% Create MaterialPanelComparison_2
app.MaterialPanelComparison_2 = uipanel(app.CyclotronScoreTab);
app.MaterialPanelComparison_2.ForegroundColor = [1 1 1];
app.MaterialPanelComparison_2.Title = 'Material';
app.MaterialPanelComparison_2.BackgroundColor = [0 0.4471 0.7412];
app.MaterialPanelComparison_2.Position = [265 405 129 63];

% Create SuperconductorDropDownComparison_2
app.SuperconductorDropDownComparison_2 =
    uidropdown(app.MaterialPanelComparison_2);
app.SuperconductorDropDownComparison_2.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDownComparison_2.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDownComparison_2ValueChanged, true);
app.SuperconductorDropDownComparison_2.Position = [15 10 100 22];
app.SuperconductorDropDownComparison_2.Value = 'NbTi_Bruker';

% Create ScoresPanel_Comparison
app.ScoresPanel_Comparison = uipanel(app.CyclotronScoreTab);
app.ScoresPanel_Comparison.ForegroundColor = [1 1 1];
app.ScoresPanel_Comparison.BorderType = 'none';
app.ScoresPanel_Comparison.Title = 'Scores';
app.ScoresPanel_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.ScoresPanel_Comparison.Position = [439 405 236 62];

% Create ScoreNumber_Comparison
app.ScoreNumber_Comparison = uieditfield(app.ScoresPanel_Comparison, 'numeric');
app.ScoreNumber_Comparison.Limits = [0 10];
app.ScoreNumber_Comparison.Position = [125 12 42 22];
app.ScoreNumber_Comparison.Value = 5;

% Create ScoreDropDown_Comparison
app.ScoreDropDown_Comparison = uidropdown(app.ScoresPanel_Comparison);
app.ScoreDropDown_Comparison.Items = {'Total Score', 'Weight', 'Price',
    'Manufacturability', 'Length'};
app.ScoreDropDown_Comparison.ValueChangedFcn = createCallbackFcn(app,
    @ScoreDropDown_ComparisonValueChanged, true);
```

```
app.ScoreDropDown_Comparison.Position = [8 12 100 22];
app.ScoreDropDown_Comparison.Value = 'Total Score';

% Create ScoreNumber2_Comparison
app.ScoreNumber2_Comparison = uieditfield(app.ScoresPanel_Comparison, 'numeric');
app.ScoreNumber2_Comparison.Limits = [0 10];
app.ScoreNumber2_Comparison.Position = [178 12 42 22];
app.ScoreNumber2_Comparison.Value = 5;

% Create OutputdataPanel_Comparison
app.OutputdataPanel_Comparison = uipanel(app.CyclotronScoreTab);
app.OutputdataPanel_Comparison.ForegroundColor = [1 1 1];
app.OutputdataPanel_Comparison.BorderType = 'none';
app.OutputdataPanel_Comparison.Title = 'Output data';
app.OutputdataPanel_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.OutputdataPanel_Comparison.Position = [265 245 410 140];

% Create WireDiammmEditField_2Label
app.WireDiammmEditField_2Label = uilabel(app.OutputdataPanel_Comparison);
app.WireDiammmEditField_2Label.HorizontalAlignment = 'right';
app.WireDiammmEditField_2Label.FontColor = [1 1 1];
app.WireDiammmEditField_2Label.Position = [192 84 99 22];
app.WireDiammmEditField_2Label.Text = 'Wire Diam. (mm) ';

% Create WireDiammmEditField_Comparison
app.WireDiammmEditField_Comparison = uieditfield(app.OutputdataPanel_Comparison,
    'numeric');
app.WireDiammmEditField_Comparison.ValueDisplayFormat = '%.3f';
app.WireDiammmEditField_Comparison.Position = [304 84 100 22];

% Create WireLenghtmEditField_2Label
app.WireLenghtmEditField_2Label = uilabel(app.OutputdataPanel_Comparison);
app.WireLenghtmEditField_2Label.HorizontalAlignment = 'right';
app.WireLenghtmEditField_2Label.FontColor = [1 1 1];
app.WireLenghtmEditField_2Label.Position = [185 49 103 22];
app.WireLenghtmEditField_2Label.Text = 'Wire Lenght (m)';

% Create WireLenghtmEditField_Comparison
app.WireLenghtmEditField_Comparison = uieditfield(app.OutputdataPanel_Comparison,
    'numeric');
app.WireLenghtmEditField_Comparison.ValueDisplayFormat = '%.1f';
app.WireLenghtmEditField_Comparison.Position = [303 50 100 22];

% Create WeightkgEditField_2Label_2
app.WeightkgEditField_2Label_2 = uilabel(app.OutputdataPanel_Comparison);
```

```
app.WeightkgEditField_2Label_2.HorizontalAlignment = 'right';
app.WeightkgEditField_2Label_2.FontColor = [1 1 1];
app.WeightkgEditField_2Label_2.Position = [1 84 67 22];
app.WeightkgEditField_2Label_2.Text = 'Weight (kg)';

% Create WeightkgEditField_Comparison
app.WeightkgEditField_Comparison = uieditfield(app.OutputdataPanel_Comparison,
    'numeric');
app.WeightkgEditField_Comparison.ValueDisplayFormat = '%.1f';
app.WeightkgEditField_Comparison.Position = [83 84 100 22];

% Create PriceEditField_2Label
app.PriceEditField_2Label = uilabel(app.OutputdataPanel_Comparison);
app.PriceEditField_2Label.HorizontalAlignment = 'right';
app.PriceEditField_2Label.FontColor = [1 1 1];
app.PriceEditField_2Label.Position = [17 49 51 22];
app.PriceEditField_2Label.Text = 'Price (ÿ)';

% Create PriceEditField_Comparison
app.PriceEditField_Comparison = uieditfield(app.OutputdataPanel_Comparison,
    'numeric');
app.PriceEditField_Comparison.ValueDisplayFormat = '%.1f';
app.PriceEditField_Comparison.Position = [83 49 100 22];

% Create WireAreamm2Label_2
app.WireAreamm2Label_2 = uilabel(app.OutputdataPanel_Comparison);
app.WireAreamm2Label_2.HorizontalAlignment = 'right';
app.WireAreamm2Label_2.FontColor = [1 1 1];
app.WireAreamm2Label_2.Position = [193 15 97 22];
app.WireAreamm2Label_2.Text = 'Wire Area (mm2)';

% Create WireAreammEditField_Comparison
app.WireAreammEditField_Comparison = uieditfield(app.OutputdataPanel_Comparison,
    'numeric');
app.WireAreammEditField_Comparison.ValueDisplayFormat = '%.1f';
app.WireAreammEditField_Comparison.Position = [305 15 100 22];

% Create DiametermLabel_2
app.DiametermLabel_2 = uilabel(app.OutputdataPanel_Comparison);
app.DiametermLabel_2.HorizontalAlignment = 'right';
app.DiametermLabel_2.FontColor = [1 1 1];
app.DiametermLabel_2.Position = [0 15 75 22];
app.DiametermLabel_2.Text = 'Diameter (m)';

% Create DiametermEditField_Custom_Comparison
```

```
app.DiametermEditField_Custom_Comparison =
    uieditfield(app.OutputdataPanel_Comparison, 'numeric');
app.DiametermEditField_Custom_Comparison.ValueDisplayFormat = '%.2f';
app.DiametermEditField_Custom_Comparison.Position = [83 15 100 22];

% Create WeightsPanel_Comparison
app.WeightsPanel_Comparison = uipanel(app.CyclotronScoreTab);
app.WeightsPanel_Comparison.ForegroundColor = [1 1 1];
app.WeightsPanel_Comparison.BorderType = 'none';
app.WeightsPanel_Comparison.Title = 'Weights';
app.WeightsPanel_Comparison.BackgroundColor = [0 0.4471 0.7412];
app.WeightsPanel_Comparison.Position = [12 2 230 148];

% Create WeightLabel
app.WeightLabel = uilabel(app.WeightsPanel_Comparison);
app.WeightLabel.HorizontalAlignment = 'right';
app.WeightLabel.FontColor = [1 1 1];
app.WeightLabel.Position = [62 99 43 22];
app.WeightLabel.Text = 'Weight';

% Create Performance_Comparison
app.Performance_Comparison = uieditfield(app.WeightsPanel_Comparison, 'numeric');
app.Performance_Comparison.Limits = [0 1];
app.Performance_Comparison.Position = [120 99 100 22];
app.Performance_Comparison.Value = 0.3;

% Create LenghtLabel_2
app.LenghtLabel_2 = uilabel(app.WeightsPanel_Comparison);
app.LenghtLabel_2.HorizontalAlignment = 'right';
app.LenghtLabel_2.FontColor = [1 1 1];
app.LenghtLabel_2.Position = [64 11 42 22];
app.LenghtLabel_2.Text = 'Lenght';

% Create Length_Comparison
app.Length_Comparison = uieditfield(app.WeightsPanel_Comparison, 'numeric');
app.Length_Comparison.Limits = [0 1];
app.Length_Comparison.Position = [120 10 100 22];
app.Length_Comparison.Value = 0.2;

% Create ManufacturabilityEditFieldLabel_3
app.ManufacturabilityEditFieldLabel_3 = uilabel(app.WeightsPanel_Comparison);
app.ManufacturabilityEditFieldLabel_3.HorizontalAlignment = 'right';
app.ManufacturabilityEditFieldLabel_3.FontColor = [1 1 1];
app.ManufacturabilityEditFieldLabel_3.Position = [10 40 96 22];
app.ManufacturabilityEditFieldLabel_3.Text = 'Manufacturability';
```

```
% Create Manufacturability_Comparison
app.Manufacturability_Comparison = uieditfield(app.WeightsPanel_Comparison,
    'numeric');
app.Manufacturability_Comparison.Limits = [0 1];
app.Manufacturability_Comparison.Position = [120 40 100 22];
app.Manufacturability_Comparison.Value = 0.2;

% Create PriceLabel_3
app.PriceLabel_3 = uilabel(app.WeightsPanel_Comparison);
app.PriceLabel_3.HorizontalAlignment = 'right';
app.PriceLabel_3.FontColor = [1 1 1];
app.PriceLabel_3.Position = [73 70 33 22];
app.PriceLabel_3.Text = 'Price';

% Create Price_Comparison
app.Price_Comparison = uieditfield(app.WeightsPanel_Comparison, 'numeric');
app.Price_Comparison.Limits = [0 1];
app.Price_Comparison.Position = [120 69 100 22];
app.Price_Comparison.Value = 0.3;

% Create OnlySuperconductorsCheckBox_2
app.OnlySuperconductorsCheckBox_2 = uicheckbox(app.CyclotronScoreTab);
app.OnlySuperconductorsCheckBox_2.ValueChangedFcn = createCallbackFcn(app,
    @OnlySuperconductorsCheckBox_2ValueChanged, true);
app.OnlySuperconductorsCheckBox_2.Text = 'Only Superconductors';
app.OnlySuperconductorsCheckBox_2.Position = [269 49 141 22];

% Create OnlyReliableDataCheckBox_2
app.OnlyReliableDataCheckBox_2 = uicheckbox(app.CyclotronScoreTab);
app.OnlyReliableDataCheckBox_2.ValueChangedFcn = createCallbackFcn(app,
    @OnlyReliableDataCheckBox_2ValueChanged, true);
app.OnlyReliableDataCheckBox_2.Text = 'Only Reliable Data';
app.OnlyReliableDataCheckBox_2.Position = [269 19 122 17];

% Create OptimumCyclotronTab
app.OptimumCyclotronTab = uitab(app.TabGroup);
app.OptimumCyclotronTab.Title = 'Optimum Cyclotron';
app.OptimumCyclotronTab.BackgroundColor = [0.9412 0.9412 0.9412];

% Create InitialDataPanel_Optimum
app.InitialDataPanel_Optimum = uipanel(app.OptimumCyclotronTab);
app.InitialDataPanel_Optimum.ForegroundColor = [1 1 1];
app.InitialDataPanel_Optimum.Title = 'Initial Data';
app.InitialDataPanel_Optimum.BackgroundColor = [0 0.4471 0.7412];
```

```
app.InitialDataPanel_Optimum.Position = [12 416 229 106];

% Create EMeVEditField_2Label
app.EMeVEditField_2Label = uilabel(app.InitialDataPanel_Optimum);
app.EMeVEditField_2Label.HorizontalAlignment = 'right';
app.EMeVEditField_2Label.FontColor = [1 1 1];
app.EMeVEditField_2Label.Position = [57 52 46 22];
app.EMeVEditField_2Label.Text = 'E(MeV)';

% Create EMeVEditField_Optimum
app.EMeVEditField_Optimum = uieditfield(app.InitialDataPanel_Optimum, 'numeric');
app.EMeVEditField_Optimum.Limits = [0 Inf];
app.EMeVEditField_Optimum.Position = [118 52 100 22];
app.EMeVEditField_Optimum.Value = 8.5;

% Create WorkingPointEditField_2Label
app.WorkingPointEditField_2Label = uilabel(app.InitialDataPanel_Optimum);
app.WorkingPointEditField_2Label.HorizontalAlignment = 'right';
app.WorkingPointEditField_2Label.FontColor = [1 1 1];
app.WorkingPointEditField_2Label.Position = [23 12 80 22];
app.WorkingPointEditField_2Label.Text = 'Working Point';

% Create WorkingPointEditField_Optimum
app.WorkingPointEditField_Optimum = uieditfield(app.InitialDataPanel_Optimum,
    'numeric');
app.WorkingPointEditField_Optimum.Limits = [0 1];
app.WorkingPointEditField_Optimum.Position = [118 12 100 22];
app.WorkingPointEditField_Optimum.Value = 0.65;

% Create GeometricDimensionsPanel_Optimum
app.GeometricDimensionsPanel_Optimum = uipanel(app.OptimumCyclotronTab);
app.GeometricDimensionsPanel_Optimum.ForegroundColor = [1 1 1];
app.GeometricDimensionsPanel_Optimum.Title = 'Geometric Dimensions';
app.GeometricDimensionsPanel_Optimum.BackgroundColor = [0 0.4471 0.7412];
app.GeometricDimensionsPanel_Optimum.Position = [12 298 229 98];

% Create AirgapmmEditField_2Label
app.AirgapmmEditField_2Label = uilabel(app.GeometricDimensionsPanel_Optimum);
app.AirgapmmEditField_2Label.HorizontalAlignment = 'right';
app.AirgapmmEditField_2Label.FontColor = [1 1 1];
app.AirgapmmEditField_2Label.Position = [32 47 71 22];
app.AirgapmmEditField_2Label.Text = 'Airgap (mm)';

% Create AirgapmmEditField_Optimum
```

```
app.AirgapmmEditField_Optimum = uieditfield(app.GeometricDimensionsPanel_Optimum,
    'numeric');
app.AirgapmmEditField_Optimum.Limits = [0 Inf];
app.AirgapmmEditField_Optimum.Position = [118 47 100 22];
app.AirgapmmEditField_Optimum.Value = 74;

% Create CryostatmmEditField_2Label
app.CryostatmmEditField_2Label = uilabel(app.GeometricDimensionsPanel_Optimum);
app.CryostatmmEditField_2Label.HorizontalAlignment = 'right';
app.CryostatmmEditField_2Label.FontColor = [1 1 1];
app.CryostatmmEditField_2Label.Position = [23 12 81 22];
app.CryostatmmEditField_2Label.Text = 'Cryostat (mm)';

% Create CryostatmmEditField_Optimum
app.CryostatmmEditField_Optimum =
    uieditfield(app.GeometricDimensionsPanel_Optimum, 'numeric');
app.CryostatmmEditField_Optimum.Limits = [0 Inf];
app.CryostatmmEditField_Optimum.Position = [118 12 100 22];
app.CryostatmmEditField_Optimum.Value = 32;

% Create MaterialPanel_Optimum
app.MaterialPanel_Optimum = uipanel(app.OptimumCyclotronTab);
app.MaterialPanel_Optimum.ForegroundColor = [1 1 1];
app.MaterialPanel_Optimum.Title = 'Material';
app.MaterialPanel_Optimum.BackgroundColor = [0 0.4471 0.7412];
app.MaterialPanel_Optimum.Position = [12 68 229 207];

% Create SteelDropDown_2Label
app.SteelDropDown_2Label = uilabel(app.MaterialPanel_Optimum);
app.SteelDropDown_2Label.HorizontalAlignment = 'right';
app.SteelDropDown_2Label.FontColor = [1 1 1];
app.SteelDropDown_2Label.Position = [70 153 33 22];
app.SteelDropDown_2Label.Text = 'Steel';

% Create SteelDropDown_Optimum
app.SteelDropDown_Optimum = uidropdown(app.MaterialPanel_Optimum);
app.SteelDropDown_Optimum.Items = {'AISI 1010', 'AISI 1020', 'ARMCO'};
app.SteelDropDown_Optimum.ValueChangedFcn = createCallbackFcn(app,
    @SteelDropDown_OptimumValueChanged, true);
app.SteelDropDown_Optimum.Position = [118 153 100 22];
app.SteelDropDown_Optimum.Value = 'AISI 1010';

% Create CopperSectionEditField_2Label_2
app.CopperSectionEditField_2Label_2 = uilabel(app.MaterialPanel_Optimum);
app.CopperSectionEditField_2Label_2.HorizontalAlignment = 'right';
```

```
app.CopperSectionEditField_2Label_2.FontColor = [1 1 1];
app.CopperSectionEditField_2Label_2.Position = [16 48 88 22];
app.CopperSectionEditField_2Label_2.Text = 'Copper Section';

% Create CopperSectionEditField_Optimum
app.CopperSectionEditField_Optimum = uieditfield(app.MaterialPanel_Optimum,
    'numeric');
app.CopperSectionEditField_Optimum.Limits = [0 Inf];
app.CopperSectionEditField_Optimum.Position = [118 48 100 22];
app.CopperSectionEditField_Optimum.Value = 4.5;

% Create AspectRatioEditField_2Label
app.AspectRatioEditField_2Label = uilabel(app.MaterialPanel_Optimum);
app.AspectRatioEditField_2Label.HorizontalAlignment = 'right';
app.AspectRatioEditField_2Label.FontColor = [1 1 1];
app.AspectRatioEditField_2Label.Position = [30 14 74 22];
app.AspectRatioEditField_2Label.Text = 'Aspect Ratio';

% Create AspectRatioEditField_Optimum
app.AspectRatioEditField_Optimum = uieditfield(app.MaterialPanel_Optimum,
    'numeric');
app.AspectRatioEditField_Optimum.Limits = [0 Inf];
app.AspectRatioEditField_Optimum.Position = [118 14 100 22];
app.AspectRatioEditField_Optimum.Value = 1;

% Create CoolingTempEditField_3Label
app.CoolingTempEditField_3Label = uilabel(app.MaterialPanel_Optimum);
app.CoolingTempEditField_3Label.HorizontalAlignment = 'right';
app.CoolingTempEditField_3Label.FontColor = [1 1 1];
app.CoolingTempEditField_3Label.Position = [22 83 82 22];
app.CoolingTempEditField_3Label.Text = 'Cooling Temp.';

% Create CoolingTempEditField_Optimum
app.CoolingTempEditField_Optimum = uieditfield(app.MaterialPanel_Optimum,
    'numeric');
app.CoolingTempEditField_Optimum.Limits = [2 9];
app.CoolingTempEditField_Optimum.ValueChangedFcn = createCallbackFcn(app,
    @CoolingTempEditField_OptimumValueChanged, true);
app.CoolingTempEditField_Optimum.Position = [118 83 100 22];
app.CoolingTempEditField_Optimum.Value = 4.2;

% Create SuperconductorDropDown_4Label_2
app.SuperconductorDropDown_4Label_2 = uilabel(app.MaterialPanel_Optimum);
app.SuperconductorDropDown_4Label_2.HorizontalAlignment = 'right';
app.SuperconductorDropDown_4Label_2.FontColor = [1 1 1];
```

```
app.SuperconductorDropDown_4Label_2.Position = [13 117 90 22];
app.SuperconductorDropDown_4Label_2.Text = 'Superconductor';

% Create SuperconductorDropDown_Optimum
app.SuperconductorDropDown_Optimum = uidropdown(app.MaterialPanel_Optimum);
app.SuperconductorDropDown_Optimum.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDown_Optimum.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDown_OptimumValueChanged, true);
app.SuperconductorDropDown_Optimum.Position = [118 117 100 22];
app.SuperconductorDropDown_Optimum.Value = 'NbTi_Bruker';

% Create FactorsPanel
app.FactorsPanel = uipanel(app.OptimumCyclotronTab);
app.FactorsPanel.ForegroundColor = [1 1 1];
app.FactorsPanel.Title = 'Factors';
app.FactorsPanel.BackgroundColor = [0 0.4471 0.7412];
app.FactorsPanel.Position = [286 416 229 106];

% Create WeightFactorEditFieldLabel
app.WeightFactorEditFieldLabel = uilabel(app.FactorsPanel);
app.WeightFactorEditFieldLabel.HorizontalAlignment = 'right';
app.WeightFactorEditFieldLabel.FontColor = [1 1 1];
app.WeightFactorEditFieldLabel.Position = [23 52 80 22];
app.WeightFactorEditFieldLabel.Text = 'Weight Factor';

% Create WeightFactorEditField
app.WeightFactorEditField = uieditfield(app.FactorsPanel, 'numeric');
app.WeightFactorEditField.Limits = [0 1];
app.WeightFactorEditField.ValueChangedFcn = createCallbackFcn(app,
    @WeightFactorEditFieldValueChanged, true);
app.WeightFactorEditField.Position = [118 52 100 22];
app.WeightFactorEditField.Value = 1;

% Create PriceFactorEditFieldLabel
app.PriceFactorEditFieldLabel = uilabel(app.FactorsPanel);
app.PriceFactorEditFieldLabel.HorizontalAlignment = 'right';
app.PriceFactorEditFieldLabel.FontColor = [1 1 1];
app.PriceFactorEditFieldLabel.Position = [33 12 70 22];
app.PriceFactorEditFieldLabel.Text = 'Price Factor';

% Create PriceFactorEditField
app.PriceFactorEditField = uieditfield(app.FactorsPanel, 'numeric');
app.PriceFactorEditField.Limits = [0 1];
```

```
app.PriceFactorEditField.ValueChangedFcn = createCallbackFcn(app,
    @PriceFactorEditFieldValueChanged, true);
app.PriceFactorEditField.Position = [118 12 100 22];

% Create UIAxes_Optimum
app.UIAxes_Optimum = uiaxes(app.OptimumCyclotronTab);
title(app.UIAxes_Optimum, 'Optimum Point')
xlabel(app.UIAxes_Optimum, 'B (T)')
ylabel(app.UIAxes_Optimum, 'Y')
app.UIAxes_Optimum.AmbientLightColor = [0.9412 0.9412 0.9412];
app.UIAxes_Optimum.PlotBoxAspectRatio = [1.80241935483871 1 1];
app.UIAxes_Optimum.GridColor = [0.149 0.149 0.149];
app.UIAxes_Optimum.MinorGridColor = [0.149 0.149 0.149];
app.UIAxes_Optimum.XColor = [0.149 0.149 0.149];
app.UIAxes_Optimum.YColor = [0.149 0.149 0.149];
app.UIAxes_Optimum.Color = [0.9412 0.9412 0.9412];
app.UIAxes_Optimum.XGrid = 'on';
app.UIAxes_Optimum.YGrid = 'on';
app.UIAxes_Optimum.BackgroundColor = [0.9412 0.9412 0.9412];
app.UIAxes_Optimum.Position = [248 139 407 257];

% Create RunButton_Optimum
app.RunButton_Optimum = uibutton(app.OptimumCyclotronTab, 'push');
app.RunButton_Optimum.ButtonPushedFcn = createCallbackFcn(app,
    @RunButton_OptimumPushed, true);
app.RunButton_Optimum.BackgroundColor = [0 0.4471 0.7412];
app.RunButton_Optimum.FontColor = [1 1 1];
app.RunButton_Optimum.Position = [546 468 100 22];
app.RunButton_Optimum.Text = 'Run';

% Create RunQFButton_Optimum
app.RunQFButton_Optimum = uibutton(app.OptimumCyclotronTab, 'push');
app.RunQFButton_Optimum.ButtonPushedFcn = createCallbackFcn(app,
    @RunQFButton_OptimumPushed, true);
app.RunQFButton_Optimum.BackgroundColor = [0 0.4471 0.7412];
app.RunQFButton_Optimum.FontColor = [1 1 1];
app.RunQFButton_Optimum.Position = [424 23 100 22];
app.RunQFButton_Optimum.Text = 'Run QF';

% Create WeightkgEditField_2Label
app.WeightkgEditField_2Label = uilabel(app.OptimumCyclotronTab);
app.WeightkgEditField_2Label.HorizontalAlignment = 'right';
app.WeightkgEditField_2Label.Position = [272 61 67 22];
app.WeightkgEditField_2Label.Text = 'Weight (kg)';
```

```
% Create WeightkgEditField_Optimum
app.WeightkgEditField_Optimum = uieditfield(app.OptimumCyclotronTab, 'numeric');
app.WeightkgEditField_Optimum.ValueDisplayFormat = '%.1f';
app.WeightkgEditField_Optimum.Position = [354 61 100 22];

% Create PriceLabel_2
app.PriceLabel_2 = uilabel(app.OptimumCyclotronTab);
app.PriceLabel_2.HorizontalAlignment = 'right';
app.PriceLabel_2.Position = [489 103 51 22];
app.PriceLabel_2.Text = 'Price (ÿ)';

% Create PriceEditField_Optimum
app.PriceEditField_Optimum = uieditfield(app.OptimumCyclotronTab, 'numeric');
app.PriceEditField_Optimum.ValueDisplayFormat = '%.1f';
app.PriceEditField_Optimum.Position = [555 103 100 22];

% Create BTEditField_2Label
app.BTEditField_2Label = uilabel(app.OptimumCyclotronTab);
app.BTEditField_2Label.HorizontalAlignment = 'right';
app.BTEditField_2Label.Position = [307 103 32 22];
app.BTEditField_2Label.Text = 'B (T)';

% Create BTEditField_Optimum
app.BTEditField_Optimum = uieditfield(app.OptimumCyclotronTab, 'numeric');
app.BTEditField_Optimum.Position = [354 103 100 22];

% Create DiametermEditFieldLabel
app.DiametermEditFieldLabel = uilabel(app.OptimumCyclotronTab);
app.DiametermEditFieldLabel.HorizontalAlignment = 'right';
app.DiametermEditFieldLabel.Position = [465 61 75 22];
app.DiametermEditFieldLabel.Text = 'Diameter (m)';

% Create DiametermEditField_Optimum
app.DiametermEditField_Optimum = uieditfield(app.OptimumCyclotronTab, 'numeric');
app.DiametermEditField_Optimum.ValueDisplayFormat = '%.2f';
app.DiametermEditField_Optimum.Position = [555 61 100 22];

% Create Zomm_Optimum
app.Zomm_Optimum = uibutton(app.OptimumCyclotronTab, 'push');
app.Zomm_Optimum.ButtonPushedFcn = createCallbackFcn(app,
    @Zoom_OptimumButtonPushed, true);
app.Zomm_Optimum.BackgroundColor = [0 0.4471 0.7412];
app.Zomm_Optimum.FontColor = [1 1 1];
app.Zomm_Optimum.Position = [546 428 100 22];
app.Zomm_Optimum.Text = 'Zoom';
```

```
% Create OptimumScoreTab
app.OptimumScoreTab = uitab(app.TabGroup);
app.OptimumScoreTab.Title = 'Optimum Score';
app.OptimumScoreTab.BackgroundColor = [0.9412 0.9412 0.9412];

% Create InitialDataPanel_OptimumScore
app.InitialDataPanel_OptimumScore = uipanel(app.OptimumScoreTab);
app.InitialDataPanel_OptimumScore.ForegroundColor = [1 1 1];
app.InitialDataPanel_OptimumScore.Title = 'Initial Data';
app.InitialDataPanel_OptimumScore.BackgroundColor = [0 0.4471 0.7412];
app.InitialDataPanel_OptimumScore.Position = [12 321 229 201];

% Create EMeVEditField_2Label_3
app.EMeVEditField_2Label_3 = uilabel(app.InitialDataPanel_OptimumScore);
app.EMeVEditField_2Label_3.HorizontalAlignment = 'right';
app.EMeVEditField_2Label_3.FontColor = [1 1 1];
app.EMeVEditField_2Label_3.Position = [52 146 50 22];
app.EMeVEditField_2Label_3.Text = 'E (MeV)';

% Create EMeVEditField_OptimumScore
app.EMeVEditField_OptimumScore = uieditfield(app.InitialDataPanel_OptimumScore,
    'numeric');
app.EMeVEditField_OptimumScore.Limits = [0 Inf];
app.EMeVEditField_OptimumScore.Position = [117 146 100 22];
app.EMeVEditField_OptimumScore.Value = 8.5;

% Create WorkingPointEditField_2Label_3
app.WorkingPointEditField_2Label_3 = uilabel(app.InitialDataPanel_OptimumScore);
app.WorkingPointEditField_2Label_3.HorizontalAlignment = 'right';
app.WorkingPointEditField_2Label_3.FontColor = [1 1 1];
app.WorkingPointEditField_2Label_3.Position = [23 77 80 22];
app.WorkingPointEditField_2Label_3.Text = 'Working Point';

% Create WorkingPointEditField_OptimumScore
app.WorkingPointEditField_OptimumScore =
    uieditfield(app.InitialDataPanel_OptimumScore, 'numeric');
app.WorkingPointEditField_OptimumScore.Limits = [0 1];
app.WorkingPointEditField_OptimumScore.Position = [118 77 100 22];
app.WorkingPointEditField_OptimumScore.Value = 0.65;

% Create WireCurentAEditField_2Label_2
app.WireCurentAEditField_2Label_2 = uilabel(app.InitialDataPanel_OptimumScore);
app.WireCurentAEditField_2Label_2.HorizontalAlignment = 'right';
app.WireCurentAEditField_2Label_2.FontColor = [1 1 1];
```

```
app.WireCurentAEditField_2Label_2.Position = [13 43 89 22];
app.WireCurentAEditField_2Label_2.Text = 'Wire Curent (A)';

% Create WireCurentAEditField_OptimumScore
app.WireCurentAEditField_OptimumScore =
    uieditfield(app.InitialDataPanel_OptimumScore, 'numeric');
app.WireCurentAEditField_OptimumScore.Limits = [0 Inf];
app.WireCurentAEditField_OptimumScore.Position = [117 43 100 22];
app.WireCurentAEditField_OptimumScore.Value = 100;

% Create CoolingTempEditField_2Label_2
app.CoolingTempEditField_2Label_2 = uilabel(app.InitialDataPanel_OptimumScore);
app.CoolingTempEditField_2Label_2.HorizontalAlignment = 'right';
app.CoolingTempEditField_2Label_2.FontColor = [1 1 1];
app.CoolingTempEditField_2Label_2.Position = [21 111 82 22];
app.CoolingTempEditField_2Label_2.Text = 'Cooling Temp.';

% Create CoolingTempEditField_OptimumScore
app.CoolingTempEditField_OptimumScore =
    uieditfield(app.InitialDataPanel_OptimumScore, 'numeric');
app.CoolingTempEditField_OptimumScore.Limits = [0 Inf];
app.CoolingTempEditField_OptimumScore.Position = [117 111 100 22];
app.CoolingTempEditField_OptimumScore.Value = 4.2;

% Create SteelDropDown_2Label_3
app.SteelDropDown_2Label_3 = uilabel(app.InitialDataPanel_OptimumScore);
app.SteelDropDown_2Label_3.HorizontalAlignment = 'right';
app.SteelDropDown_2Label_3.FontColor = [1 1 1];
app.SteelDropDown_2Label_3.Position = [70 9 33 22];
app.SteelDropDown_2Label_3.Text = 'Steel';

% Create SteelDropDown_OptimumScore
app.SteelDropDown_OptimumScore = uidropdown(app.InitialDataPanel_OptimumScore);
app.SteelDropDown_OptimumScore.Items = {'AISI 1010', 'AISI 1020', 'ARMCO'};
app.SteelDropDown_OptimumScore.ValueChangedFcn = createCallbackFcn(app,
    @SteelDropDown_OptimumScoreValueChanged, true);
app.SteelDropDown_OptimumScore.Position = [118 9 100 22];
app.SteelDropDown_OptimumScore.Value = 'AISI 1010';

% Create GeometricDimensionsPanel_OptimumScore
app.GeometricDimensionsPanel_OptimumScore = uipanel(app.OptimumScoreTab);
app.GeometricDimensionsPanel_OptimumScore.ForegroundColor = [1 1 1];
app.GeometricDimensionsPanel_OptimumScore.BorderType = 'none';
app.GeometricDimensionsPanel_OptimumScore.Title = 'Geometric Dimensions';
app.GeometricDimensionsPanel_OptimumScore.BackgroundColor = [0 0.4471 0.7412];
```

```
app.GeometricDimensionsPanel_OptimumScore.Position = [13 178 229 129];

% Create AirgapmmEditField_2Label_3
app.AirgapmmEditField_2Label_3 =
    uilabel(app.GeometricDimensionsPanel_OptimumScore);
app.AirgapmmEditField_2Label_3.HorizontalAlignment = 'right';
app.AirgapmmEditField_2Label_3.FontColor = [1 1 1];
app.AirgapmmEditField_2Label_3.Position = [32 79 71 22];
app.AirgapmmEditField_2Label_3.Text = 'Airgap (mm)';

% Create AirgapmmEditField_OptimumScore
app.AirgapmmEditField_OptimumScore =
    uieditfield(app.GeometricDimensionsPanel_OptimumScore, 'numeric');
app.AirgapmmEditField_OptimumScore.Limits = [0 Inf];
app.AirgapmmEditField_OptimumScore.Position = [118 79 100 22];
app.AirgapmmEditField_OptimumScore.Value = 74;

% Create CryostatmmEditField_2Label_3
app.CryostatmmEditField_2Label_3 =
    uilabel(app.GeometricDimensionsPanel_OptimumScore);
app.CryostatmmEditField_2Label_3.HorizontalAlignment = 'right';
app.CryostatmmEditField_2Label_3.FontColor = [1 1 1];
app.CryostatmmEditField_2Label_3.Position = [23 44 81 22];
app.CryostatmmEditField_2Label_3.Text = 'Cryostat (mm)';

% Create CryostatmmEditField_OptimumScore
app.CryostatmmEditField_OptimumScore =
    uieditfield(app.GeometricDimensionsPanel_OptimumScore, 'numeric');
app.CryostatmmEditField_OptimumScore.Limits = [0 Inf];
app.CryostatmmEditField_OptimumScore.Position = [118 44 100 22];
app.CryostatmmEditField_OptimumScore.Value = 32;

% Create AspectRatioEditField_2Label_3
app.AspectRatioEditField_2Label_3 =
    uilabel(app.GeometricDimensionsPanel_OptimumScore);
app.AspectRatioEditField_2Label_3.HorizontalAlignment = 'right';
app.AspectRatioEditField_2Label_3.FontColor = [1 1 1];
app.AspectRatioEditField_2Label_3.Position = [31 10 74 22];
app.AspectRatioEditField_2Label_3.Text = 'Aspect Ratio';

% Create AspectRatioEditField_OptimumScore
app.AspectRatioEditField_OptimumScore =
    uieditfield(app.GeometricDimensionsPanel_OptimumScore, 'numeric');
app.AspectRatioEditField_OptimumScore.Limits = [0 Inf];
app.AspectRatioEditField_OptimumScore.Position = [119 10 100 22];
```

```
app.AspectRatioEditField_OptimumScore.Value = 1;

% Create RunButton_OptimumScore
app.RunButton_OptimumScore = uibutton(app.OptimumScoreTab, 'push');
app.RunButton_OptimumScore.ButtonPushedFcn = createCallbackFcn(app,
    @RunButton_OptimumScorePushed, true);
app.RunButton_OptimumScore.BackgroundColor = [0 0.4471 0.7412];
app.RunButton_OptimumScore.FontColor = [1 1 1];
app.RunButton_OptimumScore.Position = [265 488 100 22];
app.RunButton_OptimumScore.Text = 'Run';

% Create RankingsPanel_OptimumScore
app.RankingsPanel_OptimumScore = uipanel(app.OptimumScoreTab);
app.RankingsPanel_OptimumScore.ForegroundColor = [1 1 1];
app.RankingsPanel_OptimumScore.Title = 'Rankings';
app.RankingsPanel_OptimumScore.BackgroundColor = [0 0.4471 0.7412];
app.RankingsPanel_OptimumScore.Position = [265 120 234 157];

% Create Label_7
app.Label_7 = uilabel(app.RankingsPanel_OptimumScore);
app.Label_7.HorizontalAlignment = 'right';
app.Label_7.FontWeight = 'bold';
app.Label_7.FontAngle = 'italic';
app.Label_7.FontColor = [1 1 1];
app.Label_7.Position = [8 97 25 22];
app.Label_7.Text = '1';

% Create Name1_OptimumScore
app.Name1_OptimumScore = uieditfield(app.RankingsPanel_OptimumScore, 'text');
app.Name1_OptimumScore.Position = [48 90 100 22];

% Create Label_8
app.Label_8 = uilabel(app.RankingsPanel_OptimumScore);
app.Label_8.HorizontalAlignment = 'right';
app.Label_8.FontWeight = 'bold';
app.Label_8.FontAngle = 'italic';
app.Label_8.FontColor = [1 1 1];
app.Label_8.Position = [8 61 25 22];
app.Label_8.Text = '2';

% Create Name2_OptimumScore
app.Name2_OptimumScore = uieditfield(app.RankingsPanel_OptimumScore, 'text');
app.Name2_OptimumScore.Position = [48 54 100 22];

% Create Label_9
```

```
app.Label_9 = uilabel(app.RankingsPanel_OptimumScore);
app.Label_9.HorizontalAlignment = 'right';
app.Label_9.FontWeight = 'bold';
app.Label_9.FontAngle = 'italic';
app.Label_9.FontColor = [1 1 1];
app.Label_9.Position = [8 25 25 22];
app.Label_9.Text = '3';

% Create Name3_OptimumScore
app.Name3_OptimumScore = uieditfield(app.RankingsPanel_OptimumScore, 'text');
app.Name3_OptimumScore.Position = [48 18 100 22];

% Create Score1_OptimumScore
app.Score1_OptimumScore = uieditfield(app.RankingsPanel_OptimumScore, 'numeric');
app.Score1_OptimumScore.ValueDisplayFormat = '%11.2g';
app.Score1_OptimumScore.Position = [167 90 42 22];

% Create Score2_OptimumScore
app.Score2_OptimumScore = uieditfield(app.RankingsPanel_OptimumScore, 'numeric');
app.Score2_OptimumScore.ValueDisplayFormat = '%11.2g';
app.Score2_OptimumScore.Position = [167 54 42 22];

% Create Score3_OptimumScore
app.Score3_OptimumScore = uieditfield(app.RankingsPanel_OptimumScore, 'numeric');
app.Score3_OptimumScore.ValueDisplayFormat = '%11.2g';
app.Score3_OptimumScore.Position = [167 18 42 22];

% Create MaterialPanelOptimumScore
app.MaterialPanelOptimumScore = uipanel(app.OptimumScoreTab);
app.MaterialPanelOptimumScore.ForegroundColor = [1 1 1];
app.MaterialPanelOptimumScore.Title = 'Material';
app.MaterialPanelOptimumScore.BackgroundColor = [0 0.4471 0.7412];
app.MaterialPanelOptimumScore.Position = [265 405 129 63];

% Create SuperconductorDropDownOptimumScore
app.SuperconductorDropDownOptimumScore =
    uidropdown(app.MaterialPanelOptimumScore);
app.SuperconductorDropDownOptimumScore.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDownOptimumScore.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDownOptimumScoreValueChanged, true);
app.SuperconductorDropDownOptimumScore.Position = [15 10 100 22];
app.SuperconductorDropDownOptimumScore.Value = 'NbTi_Bruker';
```

```
% Create ScoresPanel_OptimumScore
app.ScoresPanel_OptimumScore = uipanel(app.OptimumScoreTab);
app.ScoresPanel_OptimumScore.ForegroundColor = [1 1 1];
app.ScoresPanel_OptimumScore.BorderType = 'none';
app.ScoresPanel_OptimumScore.Title = 'Scores';
app.ScoresPanel_OptimumScore.BackgroundColor = [0 0.4471 0.7412];
app.ScoresPanel_OptimumScore.Position = [439 405 236 62];

% Create ScoreNumber_OptimumScore
app.ScoreNumber_OptimumScore = uieditfield(app.ScoresPanel_OptimumScore,
    'numeric');
app.ScoreNumber_OptimumScore.Limits = [0 10];
app.ScoreNumber_OptimumScore.ValueDisplayFormat = '%.2f';
app.ScoreNumber_OptimumScore.Position = [125 12 42 22];
app.ScoreNumber_OptimumScore.Value = 5;

% Create ScoreDropDown_OptimumScore
app.ScoreDropDown_OptimumScore = uidropdown(app.ScoresPanel_OptimumScore);
app.ScoreDropDown_OptimumScore.Items = {'Total Score', 'Weight', 'Price',
    'Manufacturability', 'Length'};
app.ScoreDropDown_OptimumScore.ValueChangedFcn = createCallbackFcn(app,
    @ScoreDropDown_OptimumScoreValueChanged, true);
app.ScoreDropDown_OptimumScore.Position = [8 12 100 22];
app.ScoreDropDown_OptimumScore.Value = 'Total Score';

% Create ScoreNumber2_OptimumScore
app.ScoreNumber2_OptimumScore = uieditfield(app.ScoresPanel_OptimumScore,
    'numeric');
app.ScoreNumber2_OptimumScore.Limits = [0 10];
app.ScoreNumber2_OptimumScore.Position = [178 12 42 22];
app.ScoreNumber2_OptimumScore.Value = 5;

% Create WeightsPanel_OptimumScore
app.WeightsPanel_OptimumScore = uipanel(app.OptimumScoreTab);
app.WeightsPanel_OptimumScore.ForegroundColor = [1 1 1];
app.WeightsPanel_OptimumScore.BorderType = 'none';
app.WeightsPanel_OptimumScore.Title = 'Weights';
app.WeightsPanel_OptimumScore.BackgroundColor = [0 0.4471 0.7412];
app.WeightsPanel_OptimumScore.Position = [12 14 230 142];

% Create WeightLabel_2
app.WeightLabel_2 = uilabel(app.WeightsPanel_OptimumScore);
app.WeightLabel_2.HorizontalAlignment = 'right';
app.WeightLabel_2.FontColor = [1 1 1];
app.WeightLabel_2.Position = [62 94 43 22];
```

```
app.WeightLabel_2.Text = 'Weight';

% Create Performance_OptimumScore
app.Performance_OptimumScore = uieditfield(app.WeightsPanel_OptimumScore,
    'numeric');
app.Performance_OptimumScore.Limits = [0 1];
app.Performance_OptimumScore.Position = [120 94 100 22];
app.Performance_OptimumScore.Value = 0.3;

% Create LenghtLabel_3
app.LenghtLabel_3 = uilabel(app.WeightsPanel_OptimumScore);
app.LenghtLabel_3.HorizontalAlignment = 'right';
app.LenghtLabel_3.FontColor = [1 1 1];
app.LenghtLabel_3.Position = [64 10 42 22];
app.LenghtLabel_3.Text = 'Lenght';

% Create Length_OptimumScore
app.Length_OptimumScore = uieditfield(app.WeightsPanel_OptimumScore, 'numeric');
app.Length_OptimumScore.Limits = [0 1];
app.Length_OptimumScore.Position = [120 9 100 22];
app.Length_OptimumScore.Value = 0.2;

% Create ManufacturabilityEditFieldLabel_4
app.ManufacturabilityEditFieldLabel_4 = uilabel(app.WeightsPanel_OptimumScore);
app.ManufacturabilityEditFieldLabel_4.HorizontalAlignment = 'right';
app.ManufacturabilityEditFieldLabel_4.FontColor = [1 1 1];
app.ManufacturabilityEditFieldLabel_4.Position = [10 38 96 22];
app.ManufacturabilityEditFieldLabel_4.Text = 'Manufacturability';

% Create Manufacturability_OptimumScore
app.Manufacturability_OptimumScore = uieditfield(app.WeightsPanel_OptimumScore,
    'numeric');
app.Manufacturability_OptimumScore.Limits = [0 1];
app.Manufacturability_OptimumScore.Position = [120 38 100 22];
app.Manufacturability_OptimumScore.Value = 0.2;

% Create PriceLabel_4
app.PriceLabel_4 = uilabel(app.WeightsPanel_OptimumScore);
app.PriceLabel_4.HorizontalAlignment = 'right';
app.PriceLabel_4.FontColor = [1 1 1];
app.PriceLabel_4.Position = [72 66 33 22];
app.PriceLabel_4.Text = 'Price';

% Create Price_OptimumScore
app.Price_OptimumScore = uieditfield(app.WeightsPanel_OptimumScore, 'numeric');
```

```
app.Price_OptimumScore.Limits = [0 1];
app.Price_OptimumScore.Position = [120 66 100 22];
app.Price_OptimumScore.Value = 0.3;

% Create OnlySuperconductorsCheckBox_3
app.OnlySuperconductorsCheckBox_3 = uicheckbox(app.OptimumScoreTab);
app.OnlySuperconductorsCheckBox_3.ValueChangedFcn = createCallbackFcn(app,
    @OnlySuperconductorsCheckBox_3ValueChanged, true);
app.OnlySuperconductorsCheckBox_3.Text = 'Only Superconductors';
app.OnlySuperconductorsCheckBox_3.Position = [265 76 141 22];

% Create OnlyReliableDataCheckBox_3
app.OnlyReliableDataCheckBox_3 = uicheckbox(app.OptimumScoreTab);
app.OnlyReliableDataCheckBox_3.ValueChangedFcn = createCallbackFcn(app,
    @OnlyReliableDataCheckBox_3ValueChanged, true);
app.OnlyReliableDataCheckBox_3.Text = 'Only Reliable Data';
app.OnlyReliableDataCheckBox_3.Position = [265 45 122 17];

% Create OutputdataPanel_OptimumCyclotron
app.OutputdataPanel_OptimumCyclotron = uipanel(app.OptimumScoreTab);
app.OutputdataPanel_OptimumCyclotron.ForegroundColor = [1 1 1];
app.OutputdataPanel_OptimumCyclotron.BorderType = 'none';
app.OutputdataPanel_OptimumCyclotron.Title = 'Output data';
app.OutputdataPanel_OptimumCyclotron.BackgroundColor = [0 0.4471 0.7412];
app.OutputdataPanel_OptimumCyclotron.Position = [265 294 410 102];

% Create BTLabel_2
app.BTLabel_2 = uilabel(app.OutputdataPanel_OptimumCyclotron);
app.BTLabel_2.HorizontalAlignment = 'right';
app.BTLabel_2.FontColor = [1 1 1];
app.BTLabel_2.Position = [38 50 32 22];
app.BTLabel_2.Text = 'B (T)';

% Create BTEditField_OptimumScore
app.BTEditField_OptimumScore = uieditfield(app.OutputdataPanel_OptimumCyclotron,
    'numeric');
app.BTEditField_OptimumScore.ValueDisplayFormat = '%.1f';
app.BTEditField_OptimumScore.Position = [83 50 100 22];

% Create WeightkgEditField_2Label_3
app.WeightkgEditField_2Label_3 = uilabel(app.OutputdataPanel_OptimumCyclotron);
app.WeightkgEditField_2Label_3.HorizontalAlignment = 'right';
app.WeightkgEditField_2Label_3.FontColor = [1 1 1];
app.WeightkgEditField_2Label_3.Position = [1 16 67 22];
app.WeightkgEditField_2Label_3.Text = 'Weight (kg)';
```

```
% Create WeightkgEditField_OptimumCyclotron
app.WeightkgEditField_OptimumCyclotron =
    uieditfield(app.OutputdataPanel_OptimumCyclotron, 'numeric');
app.WeightkgEditField_OptimumCyclotron.ValueDisplayFormat = '%.1f';
app.WeightkgEditField_OptimumCyclotron.Position = [83 16 100 22];

% Create PriceEditField_2Label_2
app.PriceEditField_2Label_2 = uilabel(app.OutputdataPanel_OptimumCyclotron);
app.PriceEditField_2Label_2.HorizontalAlignment = 'right';
app.PriceEditField_2Label_2.FontColor = [1 1 1];
app.PriceEditField_2Label_2.Position = [220 50 51 22];
app.PriceEditField_2Label_2.Text = 'Price (ÿ)';

% Create PriceEditField_OptimumCyclotron
app.PriceEditField_OptimumCyclotron =
    uieditfield(app.OutputdataPanel_OptimumCyclotron, 'numeric');
app.PriceEditField_OptimumCyclotron.ValueDisplayFormat = '%.1f';
app.PriceEditField_OptimumCyclotron.Position = [286 50 100 22];

% Create DiametermLabel_3
app.DiametermLabel_3 = uilabel(app.OutputdataPanel_OptimumCyclotron);
app.DiametermLabel_3.HorizontalAlignment = 'right';
app.DiametermLabel_3.FontColor = [1 1 1];
app.DiametermLabel_3.Position = [203 16 75 22];
app.DiametermLabel_3.Text = 'Diameter (m)';

% Create DiametermEditField_OptimumCyclotron
app.DiametermEditField_OptimumCyclotron =
    uieditfield(app.OutputdataPanel_OptimumCyclotron, 'numeric');
app.DiametermEditField_OptimumCyclotron.ValueDisplayFormat = '%.2f';
app.DiametermEditField_OptimumCyclotron.Position = [286 16 100 22];

% Create ScoreTab
app.ScoreTab = uitab(app.TabGroup);
app.ScoreTab.Title = 'Score';
app.ScoreTab.BackgroundColor = [0.9412 0.9412 0.9412];

% Create InitialDataPanel_Score
app.InitialDataPanel_Score = uipanel(app.ScoreTab);
app.InitialDataPanel_Score.ForegroundColor = [1 1 1];
app.InitialDataPanel_Score.Title = 'Initial Data';
app.InitialDataPanel_Score.BackgroundColor = [0 0.4471 0.7412];
app.InitialDataPanel_Score.Position = [6 345 229 177];
```

```
% Create EMeVEditField_2Label_4
app.EMeVEditField_2Label_4 = uilabel(app.InitialDataPanel_Score);
app.EMeVEditField_2Label_4.HorizontalAlignment = 'right';
app.EMeVEditField_2Label_4.FontColor = [1 1 1];
app.EMeVEditField_2Label_4.Position = [53 122 50 22];
app.EMeVEditField_2Label_4.Text = 'E (MeV)';

% Create EMeVEditField_Score
app.EMeVEditField_Score = uieditfield(app.InitialDataPanel_Score, 'numeric');
app.EMeVEditField_Score.Limits = [0 Inf];
app.EMeVEditField_Score.Position = [118 122 100 22];
app.EMeVEditField_Score.Value = 8.5;

% Create WorkingPointEditField_2Label_4
app.WorkingPointEditField_2Label_4 = uilabel(app.InitialDataPanel_Score);
app.WorkingPointEditField_2Label_4.HorizontalAlignment = 'right';
app.WorkingPointEditField_2Label_4.FontColor = [1 1 1];
app.WorkingPointEditField_2Label_4.Position = [23 85 80 22];
app.WorkingPointEditField_2Label_4.Text = 'Working Point';

% Create WorkingPointEditField_Score
app.WorkingPointEditField_Score = uieditfield(app.InitialDataPanel_Score,
    'numeric');
app.WorkingPointEditField_Score.Limits = [0 1];
app.WorkingPointEditField_Score.Position = [118 85 100 22];
app.WorkingPointEditField_Score.Value = 0.65;

% Create WireCurentAEditField_2Label_3
app.WireCurentAEditField_2Label_3 = uilabel(app.InitialDataPanel_Score);
app.WireCurentAEditField_2Label_3.HorizontalAlignment = 'right';
app.WireCurentAEditField_2Label_3.FontColor = [1 1 1];
app.WireCurentAEditField_2Label_3.Position = [14 48 89 22];
app.WireCurentAEditField_2Label_3.Text = 'Wire Curent (A)';

% Create WireCurentAEditField_Score
app.WireCurentAEditField_Score = uieditfield(app.InitialDataPanel_Score,
    'numeric');
app.WireCurentAEditField_Score.Limits = [0 Inf];
app.WireCurentAEditField_Score.Position = [118 48 100 22];
app.WireCurentAEditField_Score.Value = 100;

% Create SteelDropDown_2Label_4
app.SteelDropDown_2Label_4 = uilabel(app.InitialDataPanel_Score);
app.SteelDropDown_2Label_4.HorizontalAlignment = 'right';
app.SteelDropDown_2Label_4.FontColor = [1 1 1];
```

```
app.SteelDropDown_2Label_4.Position = [70 12 33 22];
app.SteelDropDown_2Label_4.Text = 'Steel';

% Create SteelDropDown_Score
app.SteelDropDown_Score = uidropdown(app.InitialDataPanel_Score);
app.SteelDropDown_Score.Items = {'AISI 1010', 'AISI 1020', 'ARMCO'};
app.SteelDropDown_Score.ValueChangedFcn = createCallbackFcn(app,
    @SteelDropDown_ScoreValueChanged, true);
app.SteelDropDown_Score.Position = [118 12 100 22];
app.SteelDropDown_Score.Value = 'AISI 1010';

% Create GeometricDimensionsPanel_Score
app.GeometricDimensionsPanel_Score = uipanel(app.ScoreTab);
app.GeometricDimensionsPanel_Score.ForegroundColor = [1 1 1];
app.GeometricDimensionsPanel_Score.BorderType = 'none';
app.GeometricDimensionsPanel_Score.Title = 'Geometric Dimensions';
app.GeometricDimensionsPanel_Score.BackgroundColor = [0 0.4471 0.7412];
app.GeometricDimensionsPanel_Score.Position = [473 413 208 108];

% Create CryostatmmEditField_2Label_4
app.CryostatmmEditField_2Label_4 = uilabel(app.GeometricDimensionsPanel_Score);
app.CryostatmmEditField_2Label_4.HorizontalAlignment = 'right';
app.CryostatmmEditField_2Label_4.FontColor = [1 1 1];
app.CryostatmmEditField_2Label_4.Position = [3 54 81 22];
app.CryostatmmEditField_2Label_4.Text = 'Cryostat (mm)';

% Create CryostatmmEditField_Score
app.CryostatmmEditField_Score = uieditfield(app.GeometricDimensionsPanel_Score,
    'numeric');
app.CryostatmmEditField_Score.Limits = [0 Inf];
app.CryostatmmEditField_Score.Position = [98 54 100 22];
app.CryostatmmEditField_Score.Value = 32;

% Create AspectRatioEditField_2Label_4
app.AspectRatioEditField_2Label_4 = uilabel(app.GeometricDimensionsPanel_Score);
app.AspectRatioEditField_2Label_4.HorizontalAlignment = 'right';
app.AspectRatioEditField_2Label_4.FontColor = [1 1 1];
app.AspectRatioEditField_2Label_4.Position = [10 17 74 22];
app.AspectRatioEditField_2Label_4.Text = 'Aspect Ratio';

% Create AspectRatioEditField_Score
app.AspectRatioEditField_Score = uieditfield(app.GeometricDimensionsPanel_Score,
    'numeric');
app.AspectRatioEditField_Score.Limits = [0 Inf];
app.AspectRatioEditField_Score.Position = [98 17 100 22];
```

```
app.AspectRatioEditField_Score.Value = 1;

% Create RunButton_Score
app.RunButton_Score = uibutton(app.ScoreTab, 'push');
app.RunButton_Score.ButtonPushedFcn = createCallbackFcn(app,
    @RunButton_ScorePushed, true);
app.RunButton_Score.BackgroundColor = [0 0.4471 0.7412];
app.RunButton_Score.FontColor = [1 1 1];
app.RunButton_Score.Position = [172 284 100 22];
app.RunButton_Score.Text = 'Run';

% Create MaterialPanelScore
app.MaterialPanelScore = uipanel(app.ScoreTab);
app.MaterialPanelScore.ForegroundColor = [1 1 1];
app.MaterialPanelScore.Title = 'Material';
app.MaterialPanelScore.BackgroundColor = [0 0.4471 0.7412];
app.MaterialPanelScore.Position = [59 51 156 198];

% Create SuperconductorDropDown_Score_1
app.SuperconductorDropDown_Score_1 = uidropdown(app.MaterialPanelScore);
app.SuperconductorDropDown_Score_1.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDown_Score_1.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDown_Score_1ValueChanged, true);
app.SuperconductorDropDown_Score_1.Position = [22 129 100 22];
app.SuperconductorDropDown_Score_1.Value = 'NbTi_Bruker';

% Create SuperconductorDropDown_Score_2
app.SuperconductorDropDown_Score_2 = uidropdown(app.MaterialPanelScore);
app.SuperconductorDropDown_Score_2.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDown_Score_2.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDown_Score_2ValueChanged, true);
app.SuperconductorDropDown_Score_2.Position = [22 88 100 22];
app.SuperconductorDropDown_Score_2.Value = 'NbTi_Bruker';

% Create SuperconductorDropDown_Score_3
app.SuperconductorDropDown_Score_3 = uidropdown(app.MaterialPanelScore);
app.SuperconductorDropDown_Score_3.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDown_Score_3.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDown_Score_3ValueChanged, true);
```

```
app.SuperconductorDropDown_Score_3.Position = [22 48 100 22];
app.SuperconductorDropDown_Score_3.Value = 'NbTi_Bruker';

% Create SuperconductorDropDown_Score_4
app.SuperconductorDropDown_Score_4 = uidropdown(app.MaterialPanelScore);
app.SuperconductorDropDown_Score_4.Items = {'NbTi_Bruker', 'NbTi_Supercon',
    'NbSn_Bruker', 'NbSn_Supercon', 'MgB2_ASG', 'MgB2_Hyper', 'HTS_Sumitomo',
    'HTS_Superpower', 'HTS_Fujikura', 'HTS_Sinn', 'Copper'};
app.SuperconductorDropDown_Score_4.ValueChangedFcn = createCallbackFcn(app,
    @SuperconductorDropDown_Score_4ValueChanged, true);
app.SuperconductorDropDown_Score_4.Position = [22 8 100 22];
app.SuperconductorDropDown_Score_4.Value = 'NbTi_Bruker';

% Create ScoresPanel_Score
app.ScoresPanel_Score = uipanel(app.ScoreTab);
app.ScoresPanel_Score.ForegroundColor = [1 1 1];
app.ScoresPanel_Score.BorderType = 'none';
app.ScoresPanel_Score.Title = 'Scores';
app.ScoresPanel_Score.BackgroundColor = [0 0.4471 0.7412];
app.ScoresPanel_Score.Position = [498 51 126 198];

% Create ScoreNumber_Score_1
app.ScoreNumber_Score_1 = uieditfield(app.ScoresPanel_Score, 'numeric');
app.ScoreNumber_Score_1.Limits = [0 10];
app.ScoreNumber_Score_1.ValueDisplayFormat = '%.2f';
app.ScoreNumber_Score_1.Position = [18 130 42 22];
app.ScoreNumber_Score_1.Value = 5;

% Create ScoreDropDown_Score_1
app.ScoreDropDown_Score_1 = uidropdown(app.ScoresPanel_Score);
app.ScoreDropDown_Score_1.Items = {'Total Score', 'Weight', 'Price',
    'Manufacturability', 'Length'};
app.ScoreDropDown_Score_1.ValueChangedFcn = createCallbackFcn(app,
    @ScoreDropDown_Score_1ValueChanged, true);
app.ScoreDropDown_Score_1.Position = [14 155 100 22];
app.ScoreDropDown_Score_1.Value = 'Total Score';

% Create ScoreNumber2_Score_1
app.ScoreNumber2_Score_1 = uieditfield(app.ScoresPanel_Score, 'numeric');
app.ScoreNumber2_Score_1.Limits = [0 10];
app.ScoreNumber2_Score_1.Position = [71 130 42 22];
app.ScoreNumber2_Score_1.Value = 5;

% Create ScoreNumber_Score_2
app.ScoreNumber_Score_2 = uieditfield(app.ScoresPanel_Score, 'numeric');
```

```
app.ScoreNumber_Score_2.Limits = [0 10];
app.ScoreNumber_Score_2.ValueDisplayFormat = '%.2f';
app.ScoreNumber_Score_2.Position = [18 89 42 22];
app.ScoreNumber_Score_2.Value = 5;

% Create ScoreNumber2_Score_2
app.ScoreNumber2_Score_2 = uieditfield(app.ScoresPanel_Score, 'numeric');
app.ScoreNumber2_Score_2.Limits = [0 10];
app.ScoreNumber2_Score_2.Position = [71 89 42 22];
app.ScoreNumber2_Score_2.Value = 5;

% Create ScoreNumber_Score_3
app.ScoreNumber_Score_3 = uieditfield(app.ScoresPanel_Score, 'numeric');
app.ScoreNumber_Score_3.Limits = [0 10];
app.ScoreNumber_Score_3.ValueDisplayFormat = '%.2f';
app.ScoreNumber_Score_3.Position = [18 49 42 22];
app.ScoreNumber_Score_3.Value = 5;

% Create ScoreNumber2_Score_3
app.ScoreNumber2_Score_3 = uieditfield(app.ScoresPanel_Score, 'numeric');
app.ScoreNumber2_Score_3.Limits = [0 10];
app.ScoreNumber2_Score_3.Position = [71 49 42 22];
app.ScoreNumber2_Score_3.Value = 5;

% Create ScoreNumber_Score_4
app.ScoreNumber_Score_4 = uieditfield(app.ScoresPanel_Score, 'numeric');
app.ScoreNumber_Score_4.Limits = [0 10];
app.ScoreNumber_Score_4.ValueDisplayFormat = '%.2f';
app.ScoreNumber_Score_4.Position = [18 9 42 22];
app.ScoreNumber_Score_4.Value = 5;

% Create ScoreNumber2_Score_4
app.ScoreNumber2_Score_4 = uieditfield(app.ScoresPanel_Score, 'numeric');
app.ScoreNumber2_Score_4.Limits = [0 10];
app.ScoreNumber2_Score_4.Position = [71 9 42 22];
app.ScoreNumber2_Score_4.Value = 5;

% Create WeightsPanel_Score
app.WeightsPanel_Score = uipanel(app.ScoreTab);
app.WeightsPanel_Score.ForegroundColor = [1 1 1];
app.WeightsPanel_Score.BorderType = 'none';
app.WeightsPanel_Score.Title = 'Weights';
app.WeightsPanel_Score.BackgroundColor = [0 0.4471 0.7412];
app.WeightsPanel_Score.Position = [244 345 221 176];
```

```
% Create WeightLabel_3
app.WeightLabel_3 = uilabel(app.WeightsPanel_Score);
app.WeightLabel_3.HorizontalAlignment = 'right';
app.WeightLabel_3.FontColor = [1 1 1];
app.WeightLabel_3.Position = [52 123 43 22];
app.WeightLabel_3.Text = 'Weight';

% Create Performance_Score
app.Performance_Score = uieditfield(app.WeightsPanel_Score, 'numeric');
app.Performance_Score.Limits = [0 1];
app.Performance_Score.Position = [110 123 100 22];
app.Performance_Score.Value = 0.3;

% Create LenghtLabel_4
app.LenghtLabel_4 = uilabel(app.WeightsPanel_Score);
app.LenghtLabel_4.HorizontalAlignment = 'right';
app.LenghtLabel_4.FontColor = [1 1 1];
app.LenghtLabel_4.Position = [54 14 42 22];
app.LenghtLabel_4.Text = 'Lenght';

% Create Length_Score
app.Length_Score = uieditfield(app.WeightsPanel_Score, 'numeric');
app.Length_Score.Limits = [0 1];
app.Length_Score.Position = [110 13 100 22];
app.Length_Score.Value = 0.2;

% Create ManufacturabilityEditFieldLabel_5
app.ManufacturabilityEditFieldLabel_5 = uilabel(app.WeightsPanel_Score);
app.ManufacturabilityEditFieldLabel_5.HorizontalAlignment = 'right';
app.ManufacturabilityEditFieldLabel_5.FontColor = [1 1 1];
app.ManufacturabilityEditFieldLabel_5.Position = [1 49 96 22];
app.ManufacturabilityEditFieldLabel_5.Text = 'Manufacturability';

% Create Manufacturability_Score
app.Manufacturability_Score = uieditfield(app.WeightsPanel_Score, 'numeric');
app.Manufacturability_Score.Limits = [0 1];
app.Manufacturability_Score.Position = [111 49 100 22];
app.Manufacturability_Score.Value = 0.2;

% Create PriceLabel_5
app.PriceLabel_5 = uilabel(app.WeightsPanel_Score);
app.PriceLabel_5.HorizontalAlignment = 'right';
app.PriceLabel_5.FontColor = [1 1 1];
app.PriceLabel_5.Position = [62 86 33 22];
app.PriceLabel_5.Text = 'Price';
```

```
% Create Price_Score
app.Price_Score = uieditfield(app.WeightsPanel_Score, 'numeric');
app.Price_Score.Limits = [0 1];
app.Price_Score.Position = [110 86 100 22];
app.Price_Score.Value = 0.3;

% Create OnlySuperconductorsCheckBox_Score
app.OnlySuperconductorsCheckBox_Score = uicheckbox(app.ScoreTab);
app.OnlySuperconductorsCheckBox_Score.ValueChangedFcn = createCallbackFcn(app,
    @OnlySuperconductorsCheckBox_ScoreValueChanged, true);
app.OnlySuperconductorsCheckBox_Score.Text = 'Only Superconductors';
app.OnlySuperconductorsCheckBox_Score.Position = [12 300 141 22];

% Create OnlyReliableDataCheckBox_Score
app.OnlyReliableDataCheckBox_Score = uicheckbox(app.ScoreTab);
app.OnlyReliableDataCheckBox_Score.ValueChangedFcn = createCallbackFcn(app,
    @OnlyReliableDataCheckBox_ScoreValueChanged, true);
app.OnlyReliableDataCheckBox_Score.Text = 'Only Reliable Data';
app.OnlyReliableDataCheckBox_Score.Position = [12 275 122 17];

% Create Parameters_Score
app.Parameters_Score = uipanel(app.ScoreTab);
app.Parameters_Score.ForegroundColor = [1 1 1];
app.Parameters_Score.BorderType = 'none';
app.Parameters_Score.Title = 'Parameters';
app.Parameters_Score.BackgroundColor = [0 0.4471 0.7412];
app.Parameters_Score.Position = [245 51 216 198];

% Create AirgapmmEditField_Score_1
app.AirgapmmEditField_Score_1 = uieditfield(app.Parameters_Score, 'numeric');
app.AirgapmmEditField_Score_1.Limits = [0 Inf];
app.AirgapmmEditField_Score_1.Position = [125 130 74 22];
app.AirgapmmEditField_Score_1.Value = 74;

% Create CoolingTempEditField_Score_1
app.CoolingTempEditField_Score_1 = uieditfield(app.Parameters_Score, 'numeric');
app.CoolingTempEditField_Score_1.Limits = [0 Inf];
app.CoolingTempEditField_Score_1.Position = [9 130 42 22];
app.CoolingTempEditField_Score_1.Value = 4.2;

% Create TKLabel_Score
app.TKLabel_Score = uilabel(app.Parameters_Score);
app.TKLabel_Score.FontColor = [1 1 1];
app.TKLabel_Score.Position = [14 158 32 22];
```

```
app.TKLabel_Score.Text = 'T (K)';

% Create BTLabel__Score
app.BTLabel__Score = uilabel(app.Parameters_Score);
app.BTLabel__Score.FontColor = [1 1 1];
app.BTLabel__Score.Position = [70 157 32 22];
app.BTLabel__Score.Text = 'B (T)';

% Create AirgapmmLabel_Score
app.AirgapmmLabel_Score = uilabel(app.Parameters_Score);
app.AirgapmmLabel_Score.FontColor = [1 1 1];
app.AirgapmmLabel_Score.Position = [124 157 75 22];
app.AirgapmmLabel_Score.Text = 'Airgap (mm.)';

% Create BTEditField_Score_1
app.BTEditField_Score_1 = uieditfield(app.Parameters_Score, 'numeric');
app.BTEditField_Score_1.Limits = [0 Inf];
app.BTEditField_Score_1.Position = [62 130 42 22];
app.BTEditField_Score_1.Value = 4;

% Create AirgapmmEditField_Score_2
app.AirgapmmEditField_Score_2 = uieditfield(app.Parameters_Score, 'numeric');
app.AirgapmmEditField_Score_2.Limits = [0 Inf];
app.AirgapmmEditField_Score_2.Position = [125 89 74 22];
app.AirgapmmEditField_Score_2.Value = 74;

% Create CoolingTempEditField_Score_2
app.CoolingTempEditField_Score_2 = uieditfield(app.Parameters_Score, 'numeric');
app.CoolingTempEditField_Score_2.Limits = [0 Inf];
app.CoolingTempEditField_Score_2.Position = [9 89 42 22];
app.CoolingTempEditField_Score_2.Value = 4.2;

% Create BTEditField_Score_2
app.BTEditField_Score_2 = uieditfield(app.Parameters_Score, 'numeric');
app.BTEditField_Score_2.Limits = [0 Inf];
app.BTEditField_Score_2.Position = [62 89 42 22];
app.BTEditField_Score_2.Value = 4;

% Create AirgapmmEditField_Score_3
app.AirgapmmEditField_Score_3 = uieditfield(app.Parameters_Score, 'numeric');
app.AirgapmmEditField_Score_3.Limits = [0 Inf];
app.AirgapmmEditField_Score_3.Position = [125 49 74 22];
app.AirgapmmEditField_Score_3.Value = 74;

% Create CoolingTempEditField_Score_3
```

```
        app.CoolingTempEditField_Score_3 = uieditfield(app.Parameters_Score, 'numeric');
        app.CoolingTempEditField_Score_3.Limits = [0 Inf];
        app.CoolingTempEditField_Score_3.Position = [9 49 42 22];
        app.CoolingTempEditField_Score_3.Value = 4.2;

        % Create BTEditField_Score_3
        app.BTEditField_Score_3 = uieditfield(app.Parameters_Score, 'numeric');
        app.BTEditField_Score_3.Limits = [0 Inf];
        app.BTEditField_Score_3.Position = [62 49 42 22];
        app.BTEditField_Score_3.Value = 4;

        % Create AirgapmmEditField_Score_4
        app.AirgapmmEditField_Score_4 = uieditfield(app.Parameters_Score, 'numeric');
        app.AirgapmmEditField_Score_4.Limits = [0 Inf];
        app.AirgapmmEditField_Score_4.Position = [125 9 74 22];
        app.AirgapmmEditField_Score_4.Value = 74;

        % Create CoolingTempEditField_Score_4
        app.CoolingTempEditField_Score_4 = uieditfield(app.Parameters_Score, 'numeric');
        app.CoolingTempEditField_Score_4.Limits = [0 Inf];
        app.CoolingTempEditField_Score_4.Position = [9 9 42 22];
        app.CoolingTempEditField_Score_4.Value = 4.2;

        % Create BTEditField_Score_4
        app.BTEditField_Score_4 = uieditfield(app.Parameters_Score, 'numeric');
        app.BTEditField_Score_4.Limits = [0 Inf];
        app.BTEditField_Score_4.Position = [62 9 42 22];
        app.BTEditField_Score_4.Value = 4;

        % Create RunQFButton_Score
        app.RunQFButton_Score = uibutton(app.ScoreTab, 'push');
        app.RunQFButton_Score.ButtonPushedFcn = createCallbackFcn(app,
            @RunQFButton_ScorePushed, true);
        app.RunQFButton_Score.BackgroundColor = [0 0.4471 0.7412];
        app.RunQFButton_Score.FontColor = [1 1 1];
        app.RunQFButton_Score.Position = [326 284 100 22];
        app.RunQFButton_Score.Text = 'RunQF';

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)
```

```
        % Construct app
        function app = ciclotronapp_v5_exported

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            % Execute the startup function
            runStartupFcn(app, @startupFcn)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

Chapter 2

Ferromagnetic material code

2.1. Main

```
%% 0- Clear all variables
clc
clearvars
%% 1- DATA
% General Data
global Bapp;
global I;
global Map;

% Geometric dimensions
global b_out;
global h_out;
global mat_out;

global b_in;
global h_in;
global r_in;
global mat_in;

global n_MgB2;
global l_MgB2;
global r_MgB2;
global alpha;

global num

% Numero del caso
```

```
num = 1;

% Outer Sheath
b_out = 1.5;
h_out = 1.25;

mat_out = 7; % [1] = Paramagnetic; [2] = 1010; [3] =  1020;
             % [4] =  ARMCO; [5] = Gen_Steel; [6] =  Nickel;
             % [7] = Monel;
%Inner Sheath
b_in = 1;
h_in = 1;

r_in = 0.56;

mat_in = 6;  % [1] = Paramagnetic; [2] = 1010; [3] =  1020;
             % [4] =  ARMCO; [5] = Gen_Steel; [6] =  Nickel;
             % [7] = Monel;

% Magnesium Diboride
n_MgB2 = 6;
l_MgB2 = 0.2;
r_MgB2 = 0.35;
alpha = 50;

% Set simulation limits
Bmin = 0.2;
Bmax = 6;
Bstep = 0.2;

Imin = 20;
Imax = 620;
Istep = 40;
%% 2- Solve
addpath(fullfile(pwd,'Functions'));

doCalculations(Bmin,Bstep,Bmax,Imin,Istep,Imax);
```

2.2. Functions

2.2.1. Control function

```
function doCalculations(Bmin,Bstep,Bmax,Imin,Istep,Imax)
%% doCalculations
```

```
% Close QF windows
%  Input:
%      global QF application
%  Output
%% Check mu engineering
% General Data
global mat_in
global mat_out
global Map

%Ask User to get mu_ave
answer = questdlg('Would you like to calculate de engineering permeability?');

switch answer
    case 'Yes'
        Results = getmu_ave(Bmin,Bstep,Bmax,Imin,Istep,Imax);
    case 'No'
        try
            Results =
                csvread(string(strcat('Results_ferro',num2str(mat_out),'_',num2str(mat_in),'.csv')));
        catch
            msgbox('No results saved, running new process');
            Results = getmu_ave();
        end
end

readMap(Results,1);
%% Create Map
%Ask User to get mu_ave
answer = questdlg('Would you like to calculate de Map function?');

switch answer
    case 'Yes'
        mapBmaxBgen(Results);
    case 'No'
        try
            Map_Results =
                csvread(string(strcat('Map_ferro',num2str(mat_out),'_',num2str(mat_in),'.csv')));
            Map = readMap(Map_Results,2);
        catch
            msgbox('No results saved, running new process');
            mapBmaxBgen(Results);
        end
end
%% Transform I-B data
```

```
Wire_data = csvread('I-B_ejemplo.csv',5,0);

I_wire = Wire_data(:,2);
B_wire = Wire_data(:,1);
B_max = zeros(size(I_wire,1),1);

% Transform data
for i = 1:size(I_wire,1)
    B_max(i) = Map(B_wire(i));
end

figure(3)
plot(B_wire,I_wire);
hold on
plot(B_max,I_wire);
```

2.2.2. Get engineering permeability

```
function Results = getmu_ave(Bmin,Bstep,Bmax,Imin,Istep,Imax)
%%
%%
% General Data
global Bapp;
global I;
global num;
global mat_in;
global mat_out;

prb = createModelQF(1);

ferro_data = csvread(string(strcat('Ferro.csv')),3,0);

[Iplot,Bplot] = meshgrid(Imin:Istep:Imax,Bmin:Bstep:Bmax);
mu_ave = zeros(size(Iplot,1),size(Bplot,2));

for i = 1:size(Iplot,2)
    I = Iplot(1,i);
    for j = 1:size(Bplot,1)
        Bapp = Bplot(j);
        setData(prb,ferro_data);
        solve (prb); % Run solver
        mu_ave(j,i) = getmu (prb); % Show the field picture
        closeQF(prb);
    end
```

```
end
closeQFall();

% Save Results
[fitresult,gof] = createSurfaceFit(Iplot,Bplot,mu_ave,1);

save(string(strcat('Mu_ferro',num2str(mat_out),'_',num2str(mat_in),'.mat')),'fitresult','gof');

% Overwrite results
answer = questdlg('Would you overwrite the results?');
switch answer
    case 'Yes'
        Results = [Iplot Bplot mu_ave];
        writematrix(Results,string(strcat('Results_ferro',num2str(mat_out),'_',num2str(mat_in),'.csv')));
    case 'No'
end
```

2.2.3. Create model

```
function prb = createModelQF(n)
%%  createModelQF()
%  Run QF
%  1. Open QF
%  2. Create/Open Problem
%  3. Build Geometry
%
%  Data:
%      Global data
%  Output:
%      QF problem
%      prb: QF problem location
%% 1- Open Quickfield

global QF;

try
    QF = actxserver ('QuickField.Application');
catch
    error ('QuickField cannot start');
end

QF.MainWindow.Visible = true;
% make the QuickField main window visible
% display the welcome message
```

```
fprintf ('\n QuickField %s started', QF.Version);
%% 2- Create/Open Problem

if n == 1
    prb = createProblem(n);
elseif n == 2
    prb = createProblem(n);
end



%% 3- Build Geometry
if n == 1
    buildGeometry (prb.get ('ReferencedFile', 0)); % Build the geometry model with FE mesh
elseif n == 2
    buildGeometry2 (prb.get ('ReferencedFile', 0)); % Build the geometry model with FE mesh
end
```

2.2.4. Create problem

```
function prb = createProblem (n)
%% createProblem
% Creates QF problem
%
%  Input:
%      global QF application
%  Output:
%      prp = QF problem
%%
 global QF; % the QuickField application
 global answer;

 prompt = {'Under what name would you want to save the file?'};
 dlgtitle = 'File name';
 if n == 1
    definput = {'FerroMatrix'};
elseif n == 2
    definput = {'EquiMatrixEngMu'};
 end
 opts.Interpreter = 'None';
 answer = inputdlg(prompt,dlgtitle,[1 60],definput,opts);

 QF.DefaultFilePath = fullfile(strcat(pwd,'\QF'));

 prb = QF.Problems.Add (); % Create an empty problem
 prb.ProblemType = 'qfMagnetostatics'; % Set the analysis type
```

```
prb.Class = 'qfPlaneParallel'; % Set the geometry class
prb.LengthUnits = 'qfMillimeters'; % Set length units
prb.Coordinates = 'qfCartesian'; % Set coordinate system

% Set geometry and physical data files the problem refers to
prb.set ('ReferencedFile', 0, string(strcat(answer,'.mod')));
prb.set ('ReferencedFile', 1, string(strcat(answer,'.dms')));
prb.SaveAs (string(strcat(answer,'.pbm'))); % Save the new problem
```

2.2.5. Build geometry

```
function buildGeometry (modelFileName)
%% buildGeometry(modelFilename)
% Creates a geometry model
%
% Global data
% QuickField object
%% buildGeometry()
global QF; % the QuickField application

% Set of geometric dimensions
% Geometric dimensions

% Outer Sheath
global b_out;
global h_out;

% Inner Sheath
global b_in;
global h_in;
global r_in

% MgB2
global l_MgB2;
global r_MgB2;
global alpha;
global n_MgB2;

lo = b_out/2;
lo2 = h_out/2;

li = b_in/2;
li2 = h_in/2;

rad = alpha/360*2*pi;
```

```
sep = 2*pi*(360-alpha*n_MgB2)/(n_MgB2*360);

mdl = QF.Models.Add (); % Create an empty geometry model
mdl.SaveAs (modelFileName); % Save it immediately to establish link to the problem


% Create the Outer Sheath
shp = mdl.Shapes.AddEdge (QF.PointXY(lo, lo2), QF.PointXY(lo, -lo2));
mdl.Shapes.AddEdge (QF.PointXY (lo, -lo2), QF.PointXY(-lo, -lo2));
mdl.Shapes.AddEdge (QF.PointXY (-lo, -lo2), QF.PointXY(-lo, lo2));
mdl.Shapes.AddEdge (QF.PointXY (-lo, lo2), QF.PointXY(lo, lo2));
shp.Right.Item (1).Label = 'Outer';

% Create the Inner Sheath
% shp = mdl.Shapes.AddEdge (QF.PointXY(li, li2), QF.PointXY(li, -li2));
% mdl.Shapes.AddEdge (QF.PointXY (li, -li2), QF.PointXY(-li, -li2));
% mdl.Shapes.AddEdge (QF.PointXY (-li, -li2), QF.PointXY(-li, li2));
% mdl.Shapes.AddEdge (QF.PointXY (-li, li2), QF.PointXY(li, li2));
% shp.Right.Item (1).Label = 'MgB2';

shp = mdl.Shapes.AddEdge (QF.PointXY(r_in, 0), QF.PointXY(-r_in, 0),-pi());
mdl.Shapes.AddEdge (QF.PointXY(-r_in, 0), QF.PointXY(r_in, 0),-pi());
shp.Left.Item (1).Label = 'Inner';

% Create the superconductor
% shp = mdl.Shapes.AddEdge (QF.PointXY(e, e), QF.PointXY(e, e + l_MgB2));
% mdl.Shapes.AddEdge (QF.PointXY(e, e), QF.PointXY(e + l_MgB2, e));
% mdl.Shapes.AddEdge (QF.PointXY(e, e + l_MgB2), QF.PointXY(e + l_MgB2, e),-pi()/2);
%
% mdl.Shapes.AddEdge (QF.PointXY(e, -e), QF.PointXY(e, -e - l_MgB2));
% mdl.Shapes.AddEdge (QF.PointXY(e, -e), QF.PointXY(e + l_MgB2, -e));
% mdl.Shapes.AddEdge (QF.PointXY(e, -e - l_MgB2), QF.PointXY(e + l_MgB2, -e),pi()/2);
%
% mdl.Shapes.AddEdge (QF.PointXY(-e, -e), QF.PointXY(-e, -e - l_MgB2));
% mdl.Shapes.AddEdge (QF.PointXY(-e, -e), QF.PointXY(-e - l_MgB2, -e));
% mdl.Shapes.AddEdge (QF.PointXY(-e, -e - l_MgB2), QF.PointXY(-e - l_MgB2, -e),-pi()/2);
%
% mdl.Shapes.AddEdge (QF.PointXY(-e, e), QF.PointXY(-e, e + l_MgB2));
% mdl.Shapes.AddEdge (QF.PointXY(-e, e), QF.PointXY(-e - l_MgB2, e));
% mdl.Shapes.AddEdge (QF.PointXY(-e, e + l_MgB2), QF.PointXY(-e - l_MgB2, e),pi()/2);

for i=0:(n_MgB2-1)
    ang = i*(rad + sep) + sep/2;
    shp = mdl.Shapes.AddEdge (QF.PointXY(r_in*cos(ang), r_in*sin(ang)),
        QF.PointXY(r_in*cos(ang+rad), r_in*sin(ang+rad)),rad);
```

```
    mdl.Shapes.AddEdge (QF.PointXY(r_in*cos(ang+rad), r_in*sin(ang+rad)),
        QF.PointXY(r_MgB2*cos(ang+rad), r_MgB2*sin(ang+rad)));
    mdl.Shapes.AddEdge (QF.PointXY(r_MgB2*cos(ang+rad), r_MgB2*sin(ang+rad)),
        QF.PointXY(r_MgB2*cos(ang), r_MgB2*sin(ang)),-rad);
    mdl.Shapes.AddEdge (QF.PointXY(r_MgB2*cos(ang), r_MgB2*sin(ang)),
        QF.PointXY(r_in*cos(ang), r_in*sin(ang)));
    shp.Left.Item (1).Label = 'MgB2';
end



% Edge Labels
outerEdges = mdl.Shapes.get ('Boundary', 0);

outerEdges.Item (1).Label = 'Side';
outerEdges.Item (2).Label = 'Bottom';
outerEdges.Item (3).Label = 'Side';
outerEdges.Item (4).Label = 'Top';

 % Set Spacing
 mdl.Shapes.get ('LabeledAs', '', '', 'Outer').Spacing = b_out/50;
 mdl.Shapes.get ('LabeledAs', '', '', 'Inner').Spacing = l_MgB2/30;
 mdl.Shapes.get ('LabeledAs', '', '', 'MgB2').Spacing = l_MgB2/30;

% Generate the mesh
try
    mdl.Shapes.BuildMesh ();
catch
    error ('Mesh Problems');
end

 % Save the complete model
mdl.Save ();
```

2.2.6. Set data

```
function setData (prb,ferro_data)
%% setData(prb)
% Set the media properties, filed sources and boundary conditions
%
% Global data
% QuickField object
%% setData(prb)

global QF; % the QuickField application
```

```
% Physical data
% General Data
global I;
global Bapp;

global mat_in;
global mat_out;

% Geometric dimensions

global h_out;

% First set properties for block labels

 blockLabels = prb.get ('Labels', 3);

 for i = 1 : blockLabels.Count

     lab = blockLabels.Item (i);
     cntBlock = lab.Content;

    switch lab.Name
        case {'Inner'}
             spl = cntBlock.CreateBHCurve ();
             [a, b] = size(ferro_data);
             j = 1;
             % Create BH curve for steel
             while j ~= a
                 spl.Add (QF.PointXY(ferro_data(j,3*(mat_in - 1) + 2), ferro_data(j,3*(mat_in
                     - 1) + 3)));
                 j = j + 1;
                 % In case the number of points for each steel are different
                 if ferro_data(j,3*(mat_in - 1) + 2) == 0
                     j = a;
                 end
             end
             cntBlock.Spline = spl;
         case {'MgB2'}
             cntBlock.set('TotalCurrent', true);
             cntBlock.Loading = I;
             % We will suppose the permeability of the superconductor is 1
             cntBlock.set('Kxx', false, 0.999991);
             cntBlock.set('Kyy', false, 0.999991);
         case {'Outer'}
             spl = cntBlock.CreateBHCurve ();
```

```
            [a, b] = size(ferro_data);
            j = 1;
            % Create BH curve for steel
            while j ~= a
                spl.Add (QF.PointXY(ferro_data(j,3*(mat_out - 1) + 2),
                    ferro_data(j,3*(mat_out - 1) + 3)));
                j = j + 1;
                % In case the number of points for each steel are different
                if ferro_data(j,3*(mat_out - 1) + 2) == 0
                    j = a;
                end
            end
            cntBlock.Spline = spl;
    end % of switch
     lab.Content = cntBlock;
end % of the loop

% Magnetic Potential top
 cntEdge = prb.get ('Labels' , 2).Item ('Top').Content;
 cntEdge.Dirichlet = Bapp*h_out/2000;

 prb.get ('Labels' , 2).Item ('Top').Content = cntEdge;

 % Magnetic Potential bottom
 cntEdge = prb.get ('Labels' , 2).Item ('Bottom').Content;
 cntEdge.Dirichlet = -Bapp*h_out/2000;

 prb.get ('Labels' , 2).Item ('Bottom').Content = cntEdge;

 % Symmetry condition
 cntEdge = prb.get ('Labels' , 2).Item ('Side').Content;
 cntEdge.Neumann = 0;

 prb.get ('Labels' , 2).Item ('Side').Content = cntEdge;

% Saving data document
 prb.DataDoc.Save;
```

2.2.7. Solve problem

```
function solve (prb)
%% solve(prb)
% Solves the problem
%
% Global data
```

```
%%
global QF; % the QuickField application

if prb.CanSolve
     prb.SolveProblem;
end
```

2.2.8. Get engineering permeability from QF

```
function mu_ave = getmu (prb)
%% viewResults(prb)
% Solves the problem and starts the postprocessor
%
% Global data
% QuickField object
%% viewResults(prb)

if prb.Solved
     prb.AnalyzeResults;
end
 res = prb.Result;

if ~isinterface (res)
     disp ('error: Cannot get problem result');
    return;
end

 fieldWin = res.GetFieldWindow (1);
 fieldWin.WindowState = 'qfMaximized';
 fieldWin.Height = fieldWin.Width + 10;

 Bave = res.GetIntegralTotal('qfInt_Grad_dv').Value.X;
 Have = res.GetIntegralTotal('qfInt_KGrad_dv').Value.X;

 mu_ave = (Bave/Have)/(4*pi*10^-7);
```

2.2.9. Close QF

```
function closeQF(prb)
%% closeQF
% Close QF windows
%  Input:
%      global QF application
%%
```

```
% Closing data document
prb.DataDoc.Close;

res = prb.Result;

res.Close;
```

2.2.10. Create surface fit

```
function [fitresult, gof] = createSurfaceFit(x, y, z, n)
%CREATESURFACEFIT(X_OUT,Y_OUT,Z_OUT)
%  Create a fit.
%
%  Data for fit:
%      X Input : x_out
%      Y Input : y_out
%      Z Output: z_out
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.



%% Fit: 'B-T-J curve'.
[xData, yData, zData] = prepareSurfaceData(x,y,z);

% Set up fittype and options.

% ft = fittype( 'poly32' );
% opts = fitoptions( 'Method', 'LinearLeastSquares' );
% opts.Robust = 'Off';

ft = 'cubicinterp';

% % Fit model to data.
% [fitresult, gof] = fit( [xData, yData], zData, ft,opts );

% Fit model to data.
[fitresult, gof] = fit( [xData, yData], zData, ft, 'Normalize', 'on' );

% Plot fit with data.
figure( 'Name', 'Surface Response' );
plot( fitresult, [xData, yData], zData );
```

```
% Label axes
xlabel( 'I (A)', 'Interpreter', 'none' );
ylabel( 'Bgen (T)', 'Interpreter', 'none' );
if n == 1
    zlabel( 'Muave', 'Interpreter', 'none' );
elseif n == 2
    zlabel( 'Bmax (T)', 'Interpreter', 'none' );
end
grid on
view( 76.1, 23.2 );
```

2.2.11. Read engineering permeability/ Mapping function

```
function Map = readMap(Map_Results,a)
%%
%%
global mat_in
global mat_out
if a == 1
    n = size(Map_Results,2)/3;

    I = Map_Results(:,[1:n]);
    B = Map_Results(:,[n+1:2*n]);
    Z = Map_Results(:,[2*n+1:3*n]);
elseif a == 2
    n = (size(Map_Results,2)-1)/2;
    I = Map_Results(:,[1:n]);
    B = Map_Results(:,[n+1:2*n]);
    Z = Map_Results(:,end);
end

if a == 1
    [fitresult,gof] = createSurfaceFit(I,B,Z,a);
    Map = fitresult;
    save(string(strcat('Mu_ferro',num2str(mat_out),'_',num2str(mat_in),'.mat')),'fitresult','gof');
elseif a == 2
    [fitresult,gof] = createCurveFit(B(:,1),Z);
    Map = fitresult;
    save(string(strcat('Map_ferro',num2str(mat_out),'_',num2str(mat_in),'.mat')),'fitresult','gof');
end
```

2.2.12. Create mapping function

```
function mapBmaxBgen(Results)
%%
```

```
%%
% General Data
global Map
global mat_in
global mat_out

n = size(Results,2)/3;

Iplot = Results(:,[1:n]);
Bplot = Results(:,[n+1:2*n]);
mu_ave = Results(:,[2*n+1:3*n]);

Bmax = zeros(size(Bplot,1),1);

prb = createModelQF(2);

% for i = 1:size(Iplot,2)
%     I_nom = Iplot(1,i);
%     for j = 1:size(Bplot,1)
%         Bgen = Bplot(j);
%         mu_ing = mu_ave(j,i);
%         setData2(prb,I_nom,Bgen,mu_ing);
%         solve (prb); % Run solver
%         Bmax(j,i) = getBmax(prb);
%         closeQF(prb);
%     end
% end

I_nom = 100;
for j = 1:size(Bplot,1)
    Bgen = Bplot(j);
    mu_ing = mu_ave(j);
    setData2(prb,I_nom,Bgen,mu_ing);
    solve(prb); % Run solver
    Bmax(j) = getBmax(prb);
    closeQF(prb);
end

closeQFall();

[fitresult,gof] = createCurveFit(Bplot(:,1),Bmax);
Map = fitresult;

save(string(strcat('Map_ferro',num2str(mat_out),'_',num2str(mat_in),'.mat')),'fitresult','gof');
```

```
% Reverse Map

[fitresult,gof] = createCurveFit(Bmax,Bplot(:,1));
save(string(strcat('Reverse_Map_ferro',num2str(mat_out),'_',num2str(mat_in),'.mat')),'fitresult','gof');

% Save Results
Results = [Iplot Bplot Bmax];
writematrix(Results,string(strcat('Map_ferro',num2str(mat_out),'_',num2str(mat_in),'.csv')));
```

2.2.13. Build geometry 2

```
function buildGeometry2(modelFileName)
%% buildGeometry(modelFilename)
% Creates a geometry model
%
% Global data
% QuickField object
%% buildGeometry()
global QF; % the QuickField application

% Set of geometric dimensions
% Geometric dimensions
global b_out;
global h_out;

lo = b_out/2;
lo2 = h_out/2;

lair = 50*lo;
lair2 = 50*lo2;

mdl = QF.Models.Add (); % Create an empty geometry model
mdl.SaveAs (modelFileName); % Save it immediately to establish link to the problem


% Create the Conductor
shp = mdl.Shapes.AddEdge (QF.PointXY(lo, lo2), QF.PointXY(lo, -lo2));
mdl.Shapes.AddEdge (QF.PointXY (lo, -lo2), QF.PointXY(-lo, -lo2));
mdl.Shapes.AddEdge (QF.PointXY (-lo, -lo2), QF.PointXY(-lo, lo2));
mdl.Shapes.AddEdge (QF.PointXY (-lo, lo2), QF.PointXY(lo, lo2));
shp.Right.Item (1).Label = 'Conductor';

% Create the Air
shp = mdl.Shapes.AddEdge (QF.PointXY(lair, lair2), QF.PointXY(lair, -lair2));
mdl.Shapes.AddEdge (QF.PointXY (lair, -lair2), QF.PointXY(-lair, -lair2));
```

```
mdl.Shapes.AddEdge (QF.PointXY (-lair, -lair2), QF.PointXY(-lair, lair2));
mdl.Shapes.AddEdge (QF.PointXY (-lair, lair2), QF.PointXY(lair, lair2));
shp.Right.Item (1).Label = 'Air';

% Edge Labels
outerEdges = mdl.Shapes.get ('Boundary', 0);

outerEdges.Item (1).Label = 'Side';
outerEdges.Item (2).Label = 'Bottom';
outerEdges.Item (3).Label = 'Side';
outerEdges.Item (4).Label = 'Top';

 % Set Spacing
 mdl.Shapes.get ('LabeledAs', '', '', 'Air').Spacing = b_out;
 mdl.Shapes.get ('LabeledAs', '', '', 'Conductor').Spacing = b_out/50;



% Generate the mesh
try
    mdl.Shapes.BuildMesh ();
catch
    error ('Mesh Problems');
end

 % Save the complete model
mdl.Save ();
```

2.2.14. Set data 2

```
function setData2 (prb,I_nom,Bgen,mu_ing)
%% setData(prb)
% Set the media properties, filed sources and boundary conditions
%
% Global data
% QuickField object
%% setData(prb)

global QF; % the QuickField application
% Physical data
% General Data

% Geometric dimensions
global h_out;

% First set properties for block labels
```

```
 blockLabels = prb.get ('Labels', 3);

 for i = 1 : blockLabels.Count

     lab = blockLabels.Item (i);
     cntBlock = lab.Content;

    switch lab.Name

        case 'Air'
             cntBlock.set('Kxx', false, 1);
             cntBlock.set('Kyy', false, 1);
         case {'Conductor'}
             cntBlock.set('TotalCurrent', true);
             cntBlock.Loading = I_nom;
             % We will suppose the permeability of the superconductor is 1
             cntBlock.set('Kxx', false, mu_ing);
             cntBlock.set('Kyy', false, mu_ing);

    end % of switch
     lab.Content = cntBlock;
end % of the loop

% Magnetic Potential top
 cntEdge = prb.get ('Labels' , 2).Item ('Top').Content;
 cntEdge.Dirichlet = Bgen*50*h_out/2000;

 prb.get ('Labels' , 2).Item ('Top').Content = cntEdge;

 % Magnetic Potential bottom
 cntEdge = prb.get ('Labels' , 2).Item ('Bottom').Content;
 cntEdge.Dirichlet = -Bgen*50*h_out/2000;

 prb.get ('Labels' , 2).Item ('Bottom').Content = cntEdge;

 % Symmetry condition
 cntEdge = prb.get ('Labels' , 2).Item ('Side').Content;
 cntEdge.Neumann = 0;

 prb.get ('Labels' , 2).Item ('Side').Content = cntEdge;

% Saving data document
 prb.DataDoc.Save;
```

2.2.15. Get Bmax

```
function Bmax = getBmax (prb)
%% viewResults(prb)
% Solves the problem and starts the postprocessor
%
% Global data
% QuickField object
%% viewResults(prb)

if prb.Solved
     prb.AnalyzeResults;
end
 res = prb.Result;

if ~isinterface (res)
     disp ('error: Cannot get problem result');
    return;
end

 fieldWin = res.GetFieldWindow (1);
 fieldWin.WindowState = 'qfMaximized';
 fieldWin.Height = fieldWin.Width + 10;

 cont = fieldWin.Contour;
 cont.AddBlock1('Conductor');
 Bmax = res.GetIntegral('qfInt_Grad_dv',cont).Value.X;
```

2.2.16. Get curve fit

```
function [fitresult, gof] = createCurveFit(x, z)
%CREATEFIT(X_OUT,Y_OUT,Z_OUT)
%  Create a fit.
%
%  Data for fit:
%      X Input : x_out
%      Y Input : y_out
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.

%% Fit: 'B-T-J curve'.
```

```
[xData, zData] = prepareCurveData(x,z);

% % Set up fittype and options.
% ft = fittype( 'poly2' );
% opts = fitoptions( 'Method', 'LinearLeastSquares' );
% opts.Robust = 'Off';
%
% % Fit model to data.
% [fitresult,gof] = fit(xData,zData,ft, opts);

ft = 'cubicinterp';

% Fit model to data.
[fitresult, gof] = fit( xData, zData, ft, 'Normalize', 'on' );

% Plot fit with data.
figure( 'Name', 'Surface Response' );
h = plot( fitresult, xData, zData );
legend( h, 'B vs Bmax', 'Location', 'NorthEast', 'Interpreter', 'none' );
% Label axes
xlabel( 'B (T)', 'Interpreter', 'none' );
ylabel( 'Bmax (T)', 'Interpreter', 'none' );
grid on
```