Alexander Michael Krainer, BSc

Design & simulation of new protection devices in the dispersion suppressor regions for the Future Circular Collider Project

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Technical Physics

submitted to

Graz University of Technology

Supervisor

Priv.-Doz. Dipl.-Ing. Dr.techn. Helmut Vincke

Institute of Theoretical and Computational Physics

European Organization for Nuclear Research(CERN)
Dr. Daniel Schulte

Graz, May 2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Graz, ________________________ ________________________

Date Signature

Abstract

The Future Particle Collider Study (FCC) is developing designs for a higher performance particle collider. The goal of the FCC is to greatly push the energy and intensity frontiers of particle colliders. The proposed hadron-hadron collider (FCC-hh) is aiming at center of mass collision energies of 100 TeV. This translates to proton beam energies of 50 TeV and stored energies of 8.4 GJ per beam. This poses high risks for the superconducting magnets in the dispersion suppressor areas, where high beam losses are expected. To ensure stable operation at top energy, new protection devices are needed to protect the magnets from the impacting beam particles. This thesis investigates and quantifies these losses through particle tracking simulations with the MERLIN tracking code. New protection designs using masks and collimators are created for these regions and their performance is evaluated using energy deposition simulations done with FLUKA. Finally a sophisticated protection design is presented which is capable of reducing the impacting energy on the superconducting magnets below the quench limits.

Contents

Preface

The European Organization for Nuclear Research (CERN) houses the Large Hadron Collider (LHC), the worlds largest and most powerful particle accelerator. It allows testing of the Standard Model of Particle Physics for its validity. Furthermore it helps answering some of the open fundamental questions in physics. One of the most significant discoveries of the LHC was the discovery of the postulated Higgs Boson, a major contribution to the Standard Model. Still, our understanding of physics remains incomplete, with fundamental pieces like dark matter or the incorporation of gravity into the Standard Model missing. To further enhance the capability of discovering new physics, the LHC will undergo a major upgrade to significantly increase the collision rate, called the High Luminosity LHC (HL-LHC) which is planned for 2020/2021. A higher collision rate means further improvement of statistics and uncertainties, but it still only probes the energy range up to 14 TeV. To conquer that the Future Circular Collider Study was launched at CERN in cooperation with the European Union and currently combines the efforts of 111 institutions around the globe. Its goal is to study the feasibility of new circular collider options for the post-LHC era that have a higher potential of discovering new physics. Among the considered options is the construction of a new hadron collider(FCC-hh) (see section 1.1.1) with center of mass energies of up to 100 TeV, 7 times higher as compared to the LHC. In addition to that the aim is to achieve a high collision rate comparable to the HL-LHC (see table 1.1), to further increase the probability to find new physics. These immense energies also propose a great challenge for machine protection and safe operation of the beam, since a small error can lead to a potential destruction of the accelerator. To prevent damage to this future machine, new methods for machine protection and the increased demands on collimation must be investigated.

The focus of this thesis lies on the protection of superconducting magnets in the FCC-hh from beam losses. This is done by designing various new protection devices and evaluating their efficiency through simulations. In Chapter 1, after a short introduction to the FCC-study, the FCC-hh and some of its key parameters as well as a complete explanation of the challenges concerning magnet protection are given. Chapter 2 gives an insight into the basics of accelerator physics as well as some basic concepts concerning circular accelerator design and a short explanation of the simulation tools used within this thesis. Considerations regarding the protection design are given in Chapter 3. In Chapter 4 the particle tracking and energy deposition studies done in this thesis are explained. The evaluation of the energy deposition data is given in Chapter 5. Finally, Chapter 6 gives the findings of this thesis.

1. Introduction

1.1. The Future Circular Collider Study

In 2012 the ATLAS and CMS experiments in the LHC at CERN discovered the postulated Higgs Boson [1, 2] and thus completed the Standard Model of Particle Physics as it is now known. To increase the statistics of this discovery and to find new physics, the LHC is being upgraded. After this upgrade, the collision rate (the luminosity) is expected to be a factor 5 higher than before (see table 1.1). This High Luminosity LHC (HL-LHC) is scheduled to start operating in the early 2020s.

To probe physics in even higher energy regimes, the Future Circular Collider Study investigates the possibility of building new, enhanced circular colliders. The study was launched at CERN in collaboration with the European Union and 111 institutions in 32 countries all over the earth. Currently different possibilities for circular colliders are studied, like an electron-positron collider (FCC-ee), a proton-electron collider (FCC-he), to further study deep inelastic scattering, and a hadron-hadron collider (FCC-hh) to reach the highest center of mass energies [3]. This thesis will focus on the future hadron-hadron collider (FCC-hh), as it is the most challenging option in terms of accelerator protection, as will be explained below.

1.1.1. FCC-hh

The FCC-hh is the proposed proton-proton collider with an additional option to use it for ion-ion collisions like presently the LHC. The proton-proton collider will reach unprecedented center of mass energies of 100 TeV, about 7 times higher than the maximum center of mass energy of the LHC at 14 TeV [4]. This massive collider would be located in a new tunnel in the area around CERN with a circumference of close to 100 km.

An accelerator with beam energies of 50 TeV and a circumference of 100 km requires dipole filed strengths of about 16 T, twice as high as the field strength of the current LHC dipole magnets.[3]

In fig. 1.1 the current layout of the FCC-hh is shown. The straight sections, marked with capital letters, contain the two main interaction regions with the highest collision rates (A, G); beam injection areas and additional interaction regions (L, B); beam extraction (D); radio-frequency (RF) cavities to accelerate the charge particles (H); and the energy collimation δ (F) and betatron-collimation β (J) (see also section 2.2). Some important parameters for the FCC-hh collider are listed in table 1.1 in comparison with LHC and HL-LHC parameters [5] and will be further discussed in the next section.

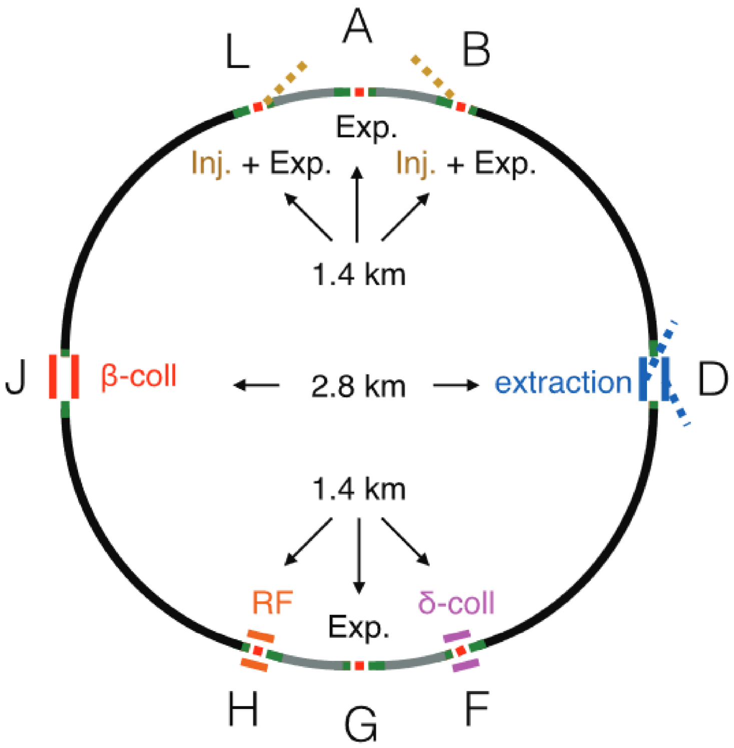


Figure 1.1.: Schematic of the FCC-hh layout [6]. The distances inside the ring show the length of the straight sections.

Table 1.1.: Important FCC-hh parameters compared to LHC and HL-LHC [5].

	LHC	HL-LHC	FCC-hh
center of mass energy [TeV]	14		100
Circumference [km]	26.7		100
Dipole field [T]	8.33		16
Number of bunches	2808		10600
Bunch population [10^{11}]	1.15	2.2	1.0
Stored energy per beam [GJ]	0.362	0.694	8.4
Peak luminosity [$1 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$]	1.0	5.0	5.0

1.1.2. Protection and collimation

As can be deducted from table 1.1, one FCC-beam consists of 10600 bunches with $1 \cdot 10^{11}$ protons with an energy of 50 TeV. This gives a total intensity of $1.06 \cdot 10^{15}$ protons per beam which translates to a stored energy of 8.4 GJ/beam. This is 23 times higher than the stored energy in the LHC (~360 MJ) [4] or 12 times the nominal goal for the high luminosity upgrade for the LHC (~700 MJ) [7].

Such high beam energies pose a great risk for the machine components, since a fraction of the beam already carries enough energy to damage them. Therefore a protection system is necessary to avoid damaging the machine and ensure stable operation. Such protection systems usually consist of masks and collimators (see section 2.2.4), made from materials that can withstand high energy impacts (e.g. graphite or tungsten). Another main purpose of the collimation system is to "shape" the beam by scraping off particles which would hit some other parts of the machine within a few turns, for example particles with energies other than the desired energy, or particles that have been drifting towards the aperture (see also section 2.2).

Challenges in machine protection

Already for the LHC a sophisticated protection and collimation system was necessary to protect the machine components from damage by the beam, since the carried beam energy was already high enough to damage components in the accelerator. [8, 9]

Measurements from the Large Hadron Collider (LHC) show that with the current collimation system particle losses mainly occur at the so called dispersion suppressors (as explained in section 2.1.4)[10]. For the current LHC energies and beam intensities, these losses are still below the quench limit of the superconducting magnets [8]. Already for the HL-LHC upgrade the intensities are high enough that the magnets would quench in the case of small beam errors and therefore safe operation of the machine cannot be ensured. Preliminary studies show that the same particle loss patterns, with the highest losses in the dispersion suppressors, are found in the FCC-hh [11], but because of the much higher beam energies these losses are even more challenging. Apart from that, secondary particle showers become more of a challenge with a factor 7 increase in energy (see section 2.3). Secondary particles are inevitably created when a beam particle hits the collimator material. Considering a 50 TeV incident proton, the resulting showers can contain hadronic particles still in the multi-TeV range. These particles still have enough energy to damage components. It is therefore of high importance to consider them in the design of an optimized collimation system.

2. Theory

2.1. Basics of accelerator physics

To bend and focus a beam of charged particles in an accelerator, magnetic fields are used. When a magnetic(or electric) field is present the force acting on a charged particle in its vicinity is given by the Lorentz force

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}) \tag{2.1}$$

where q is the particle charge, $\vec{E}$ the electric field, $\vec{v}$ the particle velocity and $\vec{B}$ the magnetic field. For circular orbits with no other forces present and no electric fields the Lorentz force must be equal to the inertial force which yields

$$B \cdot R = \frac{p}{q} \tag{2.2}$$

where B is the field strength component perpendicular to the bending plane and the initial particle direction (as only magnetic fields perpendicular to the direction of the particle cause bending). R is the bending radius and p the particle momentum.

In the vicinity of the nominal particle trajectory the magnetic field can be expanded into multipoles to describe arbitrary fields in terms of the distance from the nominal trajectory. This yields [12]:

$$\frac{q}{p}B_\mathrm{y}(x) = \frac{q}{p}B_{\mathrm{y}0} + \frac{q}{p}\frac{\mathrm{d}B_\mathrm{y}}{\mathrm{d}x}x + \frac{1}{2}\frac{q}{p}\frac{\mathrm{d}^2B_\mathrm{y}}{\mathrm{d}x^2}x^2 + ... \tag{2.3}$$

Here the choice of $B_\mathrm{y}(x)$ or $B_\mathrm{x}(y)$ assumes a particle traveling in z-direction. $B_{\mathrm{y}0}$ describes the component of the field which is independent of the displacement x.

2.1.1. Linear beam optics

In linear beam optics only fields that are either constant or increase linearly with transverse displacement of the particle from the ideal trajectory are considered. From eq. (2.3) this leaves only the first and second term.
The first term describes dipole fields which are independent of transverse displacement and can be expressed in terms of the bending radius R

$$\frac{1}{R} = \frac{q}{p} B_{\mathrm{y0}} \tag{2.4}$$

The second term describes quadrupole fields which are linear in transverse displacement. They can be expressed in terms of the quadrupole focusing strength k

$$k = \frac{q}{p} \frac{\mathrm{d}B_{\mathrm{y}}}{\mathrm{d}x} \tag{2.5}$$

If the magnetic fields changes along the trajectory of the particle k and R become functions of the particle position. [12].

When working with circular accelerators it is useful to describe a particle in terms of its displacement from the ideal trajectory, or orbit in a co-moving coordinate system. The origin of such a coordinate system moves along the orbit and follows the longitudinal position (s) of the particle. With the equation of motion in this coordinate system and the expressions of the magnetic field in terms of $R(s)$ and $k(s)$, the equations of motion for a particle traveling through the magnetic structure of an accelerator become [12]:

$$\begin{gathered} x''(s) + \left(\frac{1}{R^2(s)} - k(s) \right) x(s) = \frac{1}{R(s)} \frac{\Delta p}{p} \\ y''(s) + k(s)\, y(s) = 0 \end{gathered} \tag{2.6}$$

The factor $\frac{1}{R^2}$ describes the so called weak focusing of the beam due to the dipole field. The parameter $k(s)$ is the so called strong focusing from the quadrupole field. The focusing term occurs with opposite sign in the transverse equations. This means if a particle is focused in one plane, it

gets equally defocused in the other plane. On the right side of the first equation the dipole bending term $\frac{1}{R}$ and the momentum offset term $\frac{\Delta p}{p}$ can be seen. These only exert a force if a bending field is present and the particle has an offset in momentum.
If only particles with no momentum offset are considered and the bending radius is considered to be big (so that $\frac{1}{R^2} << k$), eq. (2.6) can be written as Hill's differential equation of motion:

$$x''(s) - k(s)\,x(s) = 0 \tag{2.7}$$

In eq. (2.7) the quadrupole strength k(s) is assumed to be a periodic function. This is true for a circular accelerator. The solution can then be written as

$$x(s) = \sqrt{\epsilon}\,\sqrt{\beta(s)}\,\cos(\Psi(s) + \Phi) \tag{2.8}$$

$$\begin{aligned} x'(s) &= -\frac{\sqrt{\epsilon}}{\beta(s)}\,[\alpha(s)\,\cos(\Psi(s)+\Phi) + \sin(\Psi(s)+\Phi)] \\ &\text{with}\,\alpha(s) = -\frac{1}{2}\beta'(s) \end{aligned} \tag{2.9}$$

Here the **beta function** $\beta(s)$ and the single particle **emittance** $\epsilon_{(single)}$ are introduced (see below). The resulting amplitude $\sqrt{\epsilon\,\beta(s)}$ creates an envelope for all trajectories of particles with a smaller emittance. Φ is the starting phase and $\Psi(s)$ is related to $\beta(s)$ by:

$$\Psi(s) = \int_0^s \frac{d\tilde{s}}{\beta(\tilde{s})} + \Phi \tag{2.10}$$

2.1.2. Phase Space

Equations (2.8) and (2.9) can be combined to get an equation for the single particle emittance:

$$\begin{aligned} \gamma(s)\,x(s)^2 + 2\,\alpha(s)\,x(s)\,x'(s) + \beta(s)\,x'(s)^2 &= \epsilon_{(single)} \\ \text{with}\,\gamma(s) &= \frac{1+\alpha(s)^2}{\beta(s)} \end{aligned} \tag{2.11}$$

The three functions ($\alpha(s)$, $\beta(s)$ and $\gamma(s)$) are called TWISS parameters. These parameters are very convenient to describe the beam in phase space. In particle physics the trajectory slope, e.q. x', is usually used instead of the transverse momenta[1]. This is valid as long as the momentum of the particle is constant, which is true for most calculations with a beam of constant energy.

Equation (2.11) describes an ellipse in phase space as seen in fig. 2.1. Here also the meaning of the TWISS parameters and the emittance becomes apparent. Liouville's theorem states that under the influence of conservative forces, the particle density in phase space stays constant[2]. This means that particles of a certain emittance will stay on their respective phase space ellipse. Also the phase space of all particles with lesser emittance is confined within this ellipse. For a particle beam the rms-emittance ϵ_{rms} [14] is used

$$\epsilon_{rms} = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle xx' \rangle^2} \tag{2.12}$$

To describe the whole particle beam at every point in an accelerator, it is enough to know the rms-emittance (which is usually given as a machine parameter), the TWISS parameters at one point in the accelerator and how these parameters transform along the beam line. It should be noted that eq. (2.11) is of course also valid for the other transverse phase space (y, y'). [13]

[1] Because of this different emittance definitions exist; the normalized emittance ϵ_n based on space-momentum phase space; and the geometric emittance ϵ used here. The two are related by: $\epsilon_n = \beta\gamma\epsilon$ with the relativistic β and γ factors [13]. Note: also sometimes a factor π is included in the definition of the emittance.

[2] The condition of conservative forces is generally met in accelerators if the accelerating electric fields are neglected, which is a valid assumption if only the top energy case is considered, as is the case in this thesis.

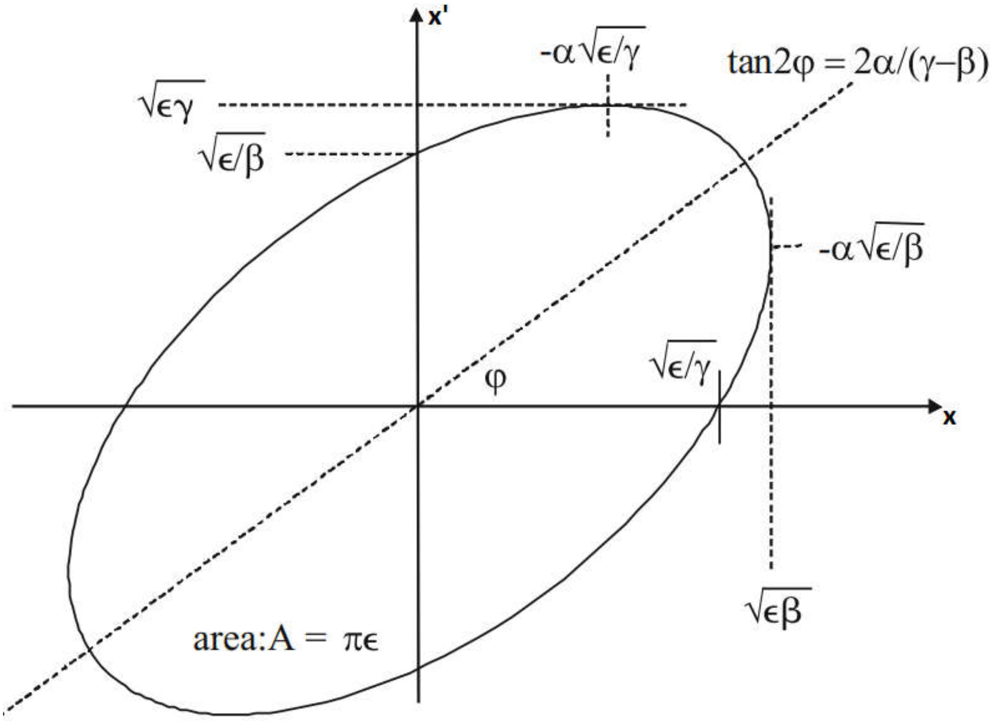


Figure 2.1.: Phase space ellipse from eq. (2.11) [13].

Another important parameter that can be deducted from fig. 2.1 is the actual transverse beam size $\sigma(s)$ at any given point in the accelerator. This is the maximum extension of the phase space ellipse in transverse direction given as

$$\sigma(s) = \sqrt{\epsilon_{rms}\beta(s)} \tag{2.13}$$

It is therefore possible to describe all particle trajectories within the beam with a beam envelope defined by the β-function. A typical example of the evolution of the β-function inside an accelerator is shown in fig. 2.2.

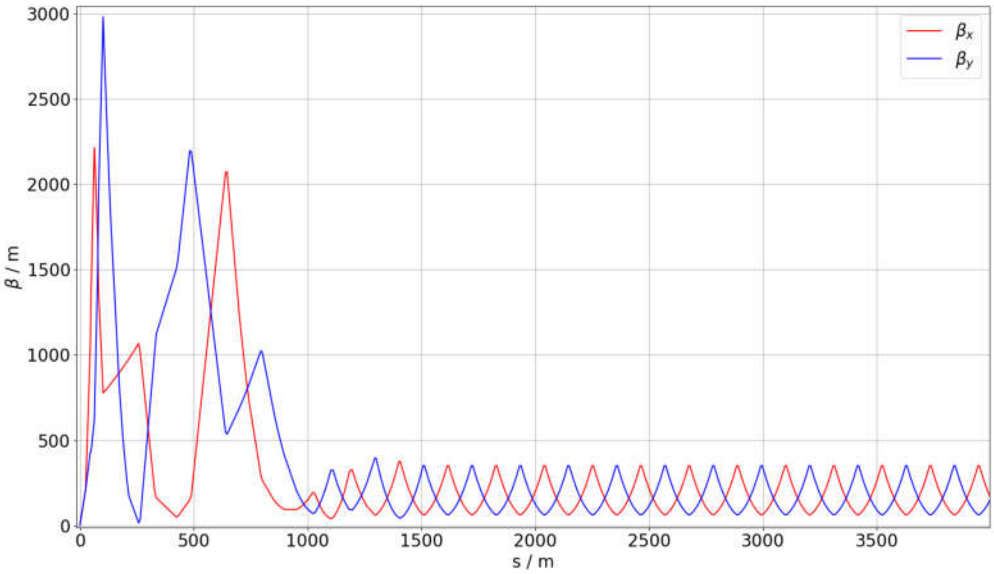


Figure 2.2.: Example for part of the beta function in a circular particle accelerator. From 0 m to 1000 m an interaction region is shown, afterwards the periodic evolution of the β-function inside the arc can be seen.

In real accelerators more the higher order terms of eq. (2.3) are also used to control the beam. This is the field of non-linear beam optics which will not be part of this thesis.

2.1.3. Phase advance and tune

In a circular accelerator the beam passes the same magnetic optics every turn. Here the phase advance of the beam, as calculated in eq. (2.10), becomes important. The phase advance over one whole turn divided by 2π give the number of betatron oscillations, which corresponds to the number of full turns around the phase space ellipse This is called the tune Q of the accelerator.

$$Q_{x,y} = \frac{1}{2\pi} \oint \frac{ds}{\beta_{x,y}(s)} \tag{2.14}$$

If the tune is an integer number the β-functions behaves like a standing wave inside the accelerator. This becomes a big problem if imperfections in the magnetic fields are considered, as this would lead to a resonance catastrophe. It is therefore necessary to choose the tune away from these resonances so that this does not happen.

2.1.4. Dispersion

Until now only particles with the same momentum equal to the ideal design momentum p have been considered. In a real accelerator however a momentum spread is present. To describe the motion of particles with a momentum offset, the $\frac{\Delta p}{p}$ term on the right side of eq. (2.6) has to be considered. As this adds a homogeneity to Hill's equation (eq. (2.7)), the solution is a linear combination of the homogeneous part, which is explained above, and a partial solution for the inhomogeneous part. This partial solution can be written as a displacement function from the ideal orbit and the complete solution is then

$$x(s) = \sqrt{\epsilon}\,\sqrt{\beta_x(s)} + D(s)\,\frac{\Delta p}{p} \tag{2.15}$$

As mentioned above, this displacement is only present perpendicular to the dipole bending fields.

$D(s)$ is called the dispersion function and is the solution to

$$D''(s) + \left(\frac{1}{R^2(s)} - k(s)\right) D(s) = \frac{1}{R(s)} \tag{2.16}$$

In a periodic lattice this solution equates to

$$D(s) = \frac{\sqrt{\beta(s)}}{2\,\sin(\pi\,\nu)} \int_s^{s+L} \frac{\sqrt{\beta(\tilde{s})}}{R(\tilde{s})} \cos[\nu(\varphi(s) - \varphi(\tilde{s}) + \pi)]\,d\tilde{s} \tag{2.17}$$

Which is itself a periodic function depending on all bending magnets in the ring. Also again a resonance phenomenon occurs if the tune becomes an integer. In that case no finite orbits for off momentum particles exist anymore. To ensure stable operation an integer resonance must be avoided, thus $\nu \notin \mathbb{Z}$.

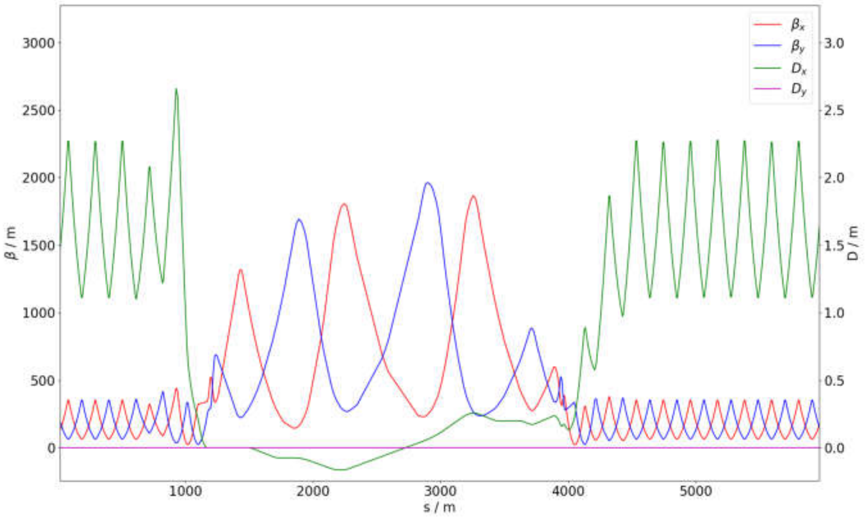


Figure 2.3.: Typical β-function and dispersion-function D through a straight section of a circular collider. Notice the different scales for $\beta(s)$ and D(s).

Dispersion Suppressor

A circular collider consists of bending sections as well as straight sections. In straight sections a periodic lattice cannot be assumed. Furthermore it is desirable to minimize the dispersion in certain straight sections of the collider (e.g. collision points). This is done by special magnet structures, so called dispersion suppressors (DS). By adjusting the magnetic fields and magnet positions the dispersion can be minimized (see [12] and [13] for detailed description). These straight sections are usually symmetric concerning the β- and dispersion-functions to be able to match them with the periodic functions inside the arc. Beam optics between sections are matched when all TWISS parameters at the end of one section have the same value at the beginning of the next section. This matching is of great importance since a mismatched beam does not lie within the accepted orbit range of the machine and will be lost. Figure 2.3 shows a typical dispersion-function from the arc, where it is periodic, through a straight section and back into the arc.

2.2. Collimation

A mentioned before a particle accelerator has a certain design energy with an accompanying design orbit. This orbit usually lies in the center of the accelerator aperture. As explained in section 2.1 the beam particles perform oscillations around this orbit, with a maximum amplitude of $\sigma_{rms}(s)$ (eq. (2.13)) at any given point in the machine. Particles that have a momentum other than the ideal one (i.e. $\frac{\Delta p}{p} > 0$) or that experience some other effect like scattering, emittance growth etc., perform oscillations around a different orbit. If the difference is small, these oscillations still fit within the aperture boundaries of the accelerator. If the difference becomes too large, the particles get lost. To ensure these losses do not happen within sensitive equipment, a minimal aperture is introduced at one point in the accelerator by means of collimators. This aperture is designed to safely remove particles too far off the desired orbit. In modern accelerators this is done by momentum (see section 2.2.2) and betatron collimation (see section 2.2.3).

2.2.1. Beam losses and lossmaps

To quantify the losses and to get a measurement for the efficiency of the collimator system, particle losses are either measured using so called beam loss monitors, or simulated using particle tracking software (see section 2.4). The efficiency is usually given in terms of the local cleaning efficiency (also called local cleaning inefficiency[3]) η as

$$\eta = \frac{N_{loc}}{N_{tot}\,\Delta s} \tag{2.18}$$

where N_{tot} is the total number of lost protons and N_{loc} being the lost protons over a specific distance Δs. From this data a so called lossmap can be generated to track the particle losses in different parts of the machine. Here a distinction is done between cold-losses, in superconducting sections; warm-losses, in normal-conducting sections; and collimator-losses. Figure 2.4 shows such a lossmap for a circular collider.

[3] The term cleaning efficiency is mostly used when the focus lies on collimation studies, and cleaning inefficiency for losses in the aperture, but this depends on the publication and preferrence of the author.

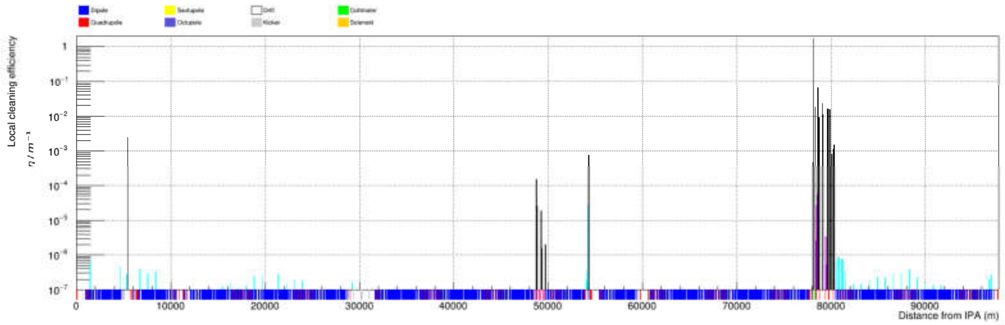


Figure 2.4.: Typical lossmap of a circular collider. Losses on collimators are pictured in black, cold losses in light blue and warm losses in purple.[15]

Beam lifetime

Lossmaps are usually created for specific cases like, steady state operation, injection, or accident scenarios. The important parameters here are the nominal beam energy (E) and the so called beam lifetime (ζ). It is the time interval after which the beam intensity has reached a value of 1/e of its initial intensity and the beam is considered lost. It is calculated using data from the beam loss monitors. For the FCC-hh a operation beam lifetime of 100 h, without collisions, is assumed [16]. For failure studies different scenarios are considered as explained in detail in [17].

- Fast losses ($\zeta << 1$ s): Fast losses and single-turn failures happen because of equipment failure (e. g. magnet quenches). In such scenarios the beam must be dumped as fast as possible and damage to the machine is very likely.

- Steady losses ($\zeta > 1$ s): Such losses can happen e.g. because of beam optic errors or debris in the beam pipe. A lifetime of more than 1 s allows the beam to be dumped normally but puts high requirements on protection devices with possible damage to collimators and absorbers.

In reality the beam will be dumped when ζ reaches 12 min and the beam cannot be recovered by correcting the beam optics within a few turns. A lifetime of $\zeta = 12\,\text{min}$ is therefore common scenario for studies of steady losses and used as a benchmark for protection devices.

2.2.2. Momentum collimation

The momentum collimation is done in a section of the accelerator where the dispersion is very high and therefore off-momentum particles drift rapidly outwards. In addition the β-function is relatively small so (almost) no betatron collimation is performed. [8, 9] For the FCC-hh layout the momentum collimation is foreseen in point F (see fig. 1.1).

2.2.3. Betatron collimation

In the betatron collimation section the β-function is large to increase the beam size. The collimator jaws are set to a defined opening gap usually given in multiples of the beam-size $\sigma(s)$ at their position. The collimation systems usually consist of primary, secondary, tertiary collimators and absorber collimators that follow a hierarchy in their respective gap size from narrow to wide. The primary collimators scatter the particles away from the nominal beam. These particles hit the secondary collimators where they get scattered further and also produce particle showers. The scattered particles and particle showers are then absorbed in the tertiary and absorber collimators (see also section 2.3.1). [8, 9] The betatron collimation section for the FCC-hh is placed in point J (see fig. 1.1).

The naming convention for collimators has been carried over from the LHC design and is shown in table 2.1. Usually the abbreviations are further specified to give the location of the collimator or its specific purpose (e.g. TCLD.8LJ - Target Collimator Long Dispersion Suppressor in cell 8 left of point J)

Table 2.1.: Naming convention for collimators

Type	Long name	Abbreviation
Primary collimator	Target Collimator Primary	TCP
Secondary collimator	Target Collimator Secondary	TCS
Tertiary collimator	Target Collimator Tertiary	TCT
Absorber collimator	Target Collimator Long	TCL

2.2.4. Protection Devices

Masks

Masks are the simplest kind of protection devices, as they usually consist only of slabs of material which fit around the beam pipe. Since they are not restricting the aperture, they cannot be used to stop particles close to the beam. Masks are not movable and are completely passive protection devices. They are used, for example, to prevent shower particles from hitting the front face of magnets.

Collimator

Collimators on the other hand are active protection devices in the sense that they can be moved closer to the beam or further away from it. It is therefore possible to dynamically define the aperture according to the needs of the accelerator. It usually consists of two material blocks, called jaws, which can be moved independently. The opening of the collimator jaw is defined as the distance of the jaw from the nominal beam orbit and is given in multiples of the beam size σ (eq. (2.13)). These collimator openings follow a strict hierarchy throughout the ring (see table 4.1 for example), this is necessary so that no particles are lost at the wrong collimators. In fig. 2.5 a diagram of a LHC collimator is shown.

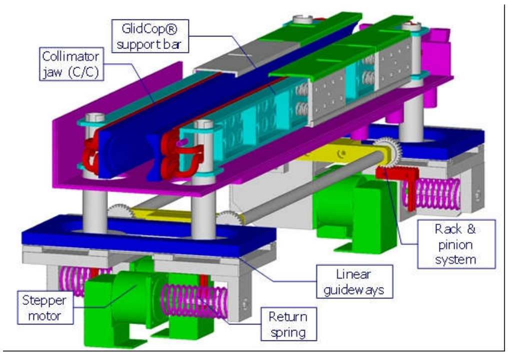


Figure 2.5.: A cutaway of the LHC Secondary Collimator [18]

2.2.5. Materials used for protection devices

The choice of material for the protection elements is of high importance and depends on a number of factors. Elements which experience high intensity, high energy impacts are usually made of a special carbon composite to withstand the high power densities. Carbon fiber composites (CFC) are used because of their relatively low density, high melting point and high specific heat capacity (much higher than for metals, especially at high temperatures). A low density ensures that the energy deposition from secondary showers produced in the material is low, while a high melting point combined with a high specific heat capacity provides a high thermal robustness. For protection devices like masks and absorbers, where the main focus lies on stopping all incident particles and absorbing their energy, high density materials are used. To ensure that these devices do not get destroyed by the absorbed energy, again a high melting point, high specific heat capacity and a good heat conduction is desired. Materials used for this purpose are for example tungsten or copper alloys. In the FCC-hh the material for the TCPs and TCSs is a CFC commercially known as AC-150-K [19, 20]. This carbon compound is a mixture of carbon fibres and graphite flakes and has a low density, high melting (sublimation) point, and good thermodynamic properties (table 2.2). For the TCTs and TCLs the chosen material is a tungsten heavy alloy commercially known as Inermet®-180 (IT-180)[21, 22]. It consists of 95 % tungsten, 3.5 % nickel and 1.5 % copper. It has almost the same density as well as a better workability than pure tungsten, while still maintaining a good heat conduction and a high melting point (table 2.2.

Table 2.2.: Material properties for AC-150-K and IT-180.
ρ...density.[23]
T_{melt}...melting temperature.[23]
c_p...specific heat capacity at room temp.[19, 24]
κ...thermal conductivity at room temp.[19, 24]

	ρ / $\mathrm{g\,cm^{-3}}$	T_{melt} / °C	c_p / $\mathrm{J\,kg^{-1}\,K^{-1}}$	κ / $\mathrm{W\,m^{-1}\,K^{-1}}$
AC-150-K	1.67	3650	710	220
IT-180	18.0	~1400	150	90

It is worth noting that a lot of research is going on to find better suited materials. Especially since the electrical resistance of the exposed surface of the collimator jaws has a high impact on beam stability [21].

2.3. Energy Deposition

2.3.1. High-energy particle interaction with matter

When a high energy proton hits a material there is a probability that the proton undergoes an interaction within the material. This probability is usually expressed in term of the total interaction cross section, which is the sum of the total cross sections of all possible interaction processes. These cross sections depend on the incident particle energy. These scattering processes include proton-electron interactions and nuclear interactions.

proton-electron interactions

The incident proton ionizes the atoms in the material by transferring energy to the bound electrons.

Nuclear interactions

Interactions of an incident proton with a target nucleus can be divided into elastic scattering, where both, the proton and the nucleus survive, and inelastic scattering, where either one of the two, or both, interaction particles get destroyed.

Elastic scattering processes result in momentum exchange and a change in the direction of the incident proton. Since the nucleus is much heavier than the proton, the exchanged energy in such processes is negligible.

Inelastic scattering processes include single-diffractive (SD) and double-diffractive (DD) scattering. In SD-scattering either the incident proton or the target nucleus becomes excited which results in its dissociation. The surviving particle only experiences a small change in its momentum. The particles produced from the dissociation of the scattering particle conserve the momentum. In the case of the dissociation of the nucleus this results in particles with small momenta compared to the total amount of momentum present in the interaction. In the case of the dissociation of the incident proton, the resulting particles have almost all of the momentum present. In DD-scattering both scattering partners dissociate. Again the exchange of momentum between the incident proton and the nucleus is small, so that the resulting particles from the nucleus dissociation have small momenta and the particles created by the dissociation of the proton get almost all of the protons momentum.[25, 26]

Mean free path and nuclear interaction length

From the total interaction cross section σ_T and the density of the material the mean free path between interactions can be calculated [26]:

$$\lambda = \frac{A}{\rho \mathrm{N_a}} \sigma_T \tag{2.19}$$

where A is the atomic mass number, ρ is the density of the material and $\mathrm{N_a}$ is the Avogadro constant. For hadronic particles this is also called the nuclear collision length. If the cross sections for elastic scattering are excluded from the total interaction cross section, the nuclear interaction length is obtained. This describes the mean distance travelled before undergoing an inelastic interaction in the material. Table 2.3 lists nuclear collision lengths and nuclear interaction lengths for some of the materials used in collimators for protons with a momentum of 200 GeV/c [27].

Table 2.3.: Nuclear collision and interaction lengths for some elements used in collimator materials for 200 GeV/c protons [27].

	Nuc. Coll. Length / $\mathrm{g\,cm^{-2}}$	Nuc. Int. Length / $\mathrm{g\,cm^{-2}}$
Carbon	59.2	85.8
Copper	84.2	137.3
Nickel	82.6	134.1
Tungsten	110.4	191.9

For composite materials like AC-150-K and IT-180 the nuclear collision and interaction lengths can be approximated by eq. (2.20), according to [27].

$$\frac{1}{\lambda} = \sum \frac{wt_i}{\lambda_i} \tag{2.20}$$

where wt_i are the mass fractions of the i-th element in the compound. For AC-150-K and IT-180 the approximations are given in table 2.4.

Table 2.4.: Nuclear collision and interaction length for AC-150-K and IT-180.
ρ...density.[23]
λ_{tot}...nuclear collision length.
λ_{inel}...nuclear interaction length.

	ρ / $\mathrm{g\,cm^{-3}}$	λ_{tot} / cm	λ_{inel} / cm
AC-150-K	1.67	35.45	51.38
IT-180	18.0	6.03	10.44

2.3.2. Energy deposition in objects

Beam-particles that do not survive the scattering produce a shower of secondary particles through the above mentioned processes. This shower gets partially absorbed within the material itself, contributing to material damage and energy deposition in form of heat. Depending on the density of the material, the incident particle energy, and the impact parameters (position, angle etc.), part of the shower will extend out of the material.

Energy Deposition in superconducting magnets

One special case of energy deposition in material to be mentioned here is the energy deposition from particles within the superconducting coils of the magnets. Superconducting materials only have zero electrical-resistance below a certain critical temperature (usually a few Kelvin). Above this temperature, these materials become normal conducting. This poses a great risk for an accelerator like the FCC or LHC, since the needed magnetic fields can only be achieved by extremely high currents in the order of several ten-thousand ampere. If such currents are present when the conductors change from the super conducting to the normal conducting state by particle induced energy deposition, significant heat up leading to the destruction of the magnet will occur. This loss of superconductivity is called a "quench". It is obvious that a quench must be prevented under all circumstances. To prevent damage to the accelerator and the magnets, a quench protection system is used to remove the current when the magnet becomes normal-conducting. In this case it takes several hours to restart the accelerator, which must be avoided if steady physics operation is to be achieved.

2.4. Software tools

2.4.1. MAD-X

MAD-X (Methodical Accelerator Design) [28] is a powerful code package for particle accelerator design and beam optics calculations. In the scope of this thesis the TWISS-Module of MAD-X was used to calculate the TWISS parameters (see section 2.1.2) from the FCC-hh lattice.

The FCC-optics repository [29] contains files that describe the different magnet elements and the sequence of elements that describes the whole collider ring as well as a MAD-X input file that describes which TWISS parameters are to be calculated. This input file was the used with

MAD-X to get the desired TWISS files as input for MERLIN.

2.4.2. MERLIN

MERLIN [30] is a C++ accelerator physics library used for particle tracking simulations in accelerators. This is useful for collimation simulations and the generation of particle lossmaps. The accelerator layout must be defined as well as the apertures for each element and the collimator gap settings. Then a beam of particles is generated based on the accelerator beam optics and desired beam distribution and tracked around the lattice. When a particle hits the aperture, the exact position of its loss is recorded. When a collimator is hit, the particle scattering within the collimator is simulated. If the energy loss through scattering becomes higher than a set threshold the particle is also recorded as lost. From this data, a lossmap can then be generated. In this thesis all lossmaps were generated with MERLIN.

For the accelerator layout input and the aperture definition, the TWISS files generated with MAD-X were used (see above). The input file for the collimator settings can be found in appendix A.1. The simulation itself has to be set up as a C++ program. Since MERLIN is very modular and versatile, there is no standard way of setting up a simulation. The simulation code used for this thesis was created with help from James Molson and can be found in appendix A.2.

2.4.3. FLUKA

FLUKA [31, 32] is a Monte-Carlo simulation tool for calculation of particle transport and particle interactions with matter. It can simulate the resulting particle showers with high accuracy in an energy range from 1 keV up to 20 TeV (10 PeV when linking it with the DPMJET Monte-Carlo event generator code [33]) and neutrons down to thermal energies. FLUKA uses a version of Combinatorial Geometry for creation of almost arbitrary complex geometries. It also provides a number of interface routines to link with the underlying Fortran77 code if necessary. FLUKA can simulate for example energy deposition, energy density or fluence in a user defined binning structure.

As mentioned above, to use FLUKA above an energy of 20 TeV it is required to link it with the DPMJET code. For this work DPMJET-3 was used out of simplicity because DPMJET-3 is included in the FLUKA download as well as a predefined script to link it into the FLUKA executable. This is done by calling the script **"ldpmqmd"** from the FLUKA utilities folder **"flutil"**. [34]

FLUKA uses text-based input files to setup a simulation. These input files contain general settings of the simulation, a description of the geometry in a Combinatorial Geometry format, a description of the used materials for each region, a definition of so called detectors to record the desired quantities (see below), the definition of the initial particle beam or distribution and finally the initialization of the random number sequence. FLUKA is written in FORTRAN77, therefore certain format constraints apply to the input files and its commands. Each command consists of one or more lines on the text file (called "cards" for historical reasons). Each card contains the command key word, six floating point values (WHAT(1-6)) and one string (SDUM). An example of such an input file can be found in appendix B.2.

Material description in FLUKA

Every geometry region is supposed to have a corresponding material. This material can either be vacuum, a chemical element, a compound of elements or the fictitious material "blackhole". The "blackhole" material is used to terminate particle trajectories and every geometry is required to be enclosed by a boundary made of it. FLUKA comes with 25 predefined materials (e.g. tungsten, nickel, iron) and 12 predefined compounds (e.g. air, water, polystyrene). If other materials are required they can be added in the input files using the MATERIAL and COMPOUND commands. To define a material, the atomic number (Z) of the element and the density (ρ) in g/cm^3 must be given in the MATERIAL card as well as a name. To define a material made of more than one element a corresponding COMPOUND card must be given, stating the material composition in terms of relative atomic content, mass fraction or volume fraction.

2.4.4. SimpleGeo

Since the Combinatorial Geometry approach of FLUKA can be, albeit its power, very cumbersome when creating advanced geometries, it is useful to utilize other tools for creating the geometry description for FLUKA. One such a tool is SimpleGeo [35], an interactive 3D-modeling software. It allows the creation of model via drag & drop and 3D visualization for direct inspection. Furthermore it incorporates debugging tools to validate the created geometry. It also includes the possibility to graphically display the results obtained from FLUKA simulations directly on top of the geometry.

3. Protection Design

As mentioned before, already the HL-LHC faces challenges concerning losses in the dispersion suppressor (DS) regions of the machine and loss-maps show that the highest losses occur at two points in the dispersion suppressor after the betatron cleaning [36]. Therefore new collimators (TCLDs) of 60 cm length, will be introduced for the HL-LHC. In fig. 3.1 the position of two new collimators in the DS after the betatron cleaning section can be seen.

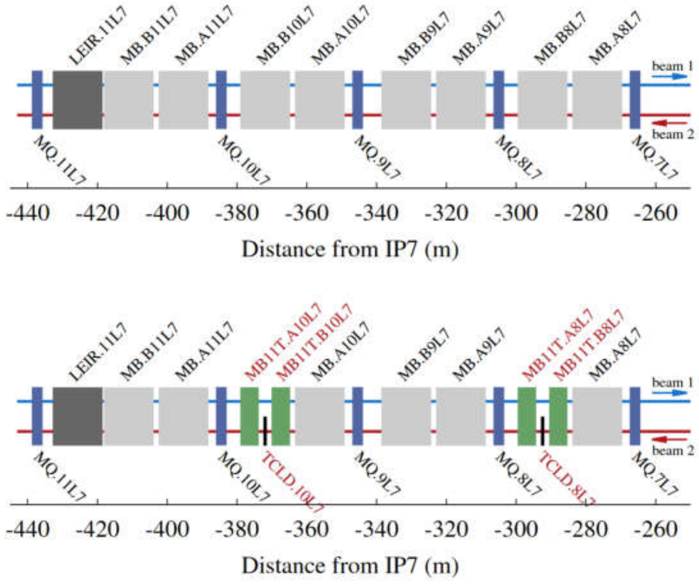


Figure 3.1.: Schematic depiction of the current DS-layout in the LHC (top), and an alternative layout, explained in detail in [36] (bottom). Here two dipoles are replaced by collimators (TCLDs) and shorter magnets with higher filed strengths.[37]

For the FCC-hh study, particle tracking simulations with an early version of the lattice were carried out and found very similar loss patterns as for the LHC. Again two loss clusters form at the end of the betatron cleaning section in the DS-region. Figure 3.2 shows a lossmap made by M. Fiascaris with an early version of the FCC-hh lattice [38], as well as a diagram of the betatron cleaning insertion where the position of the losses is indicated. The origin of these losses in the DS-region is believed to be single-diffractive scattered protons from the primary collimators[10]. As described before these protons experience only a small momentum loss but otherwise their motion is not disturbed. Therefore they can travel very far within the accelerator until the dispersion rises and they drift towards the wall.

3.1. Betatron Collimation System

The betatron collimation system for the FCC-hh is a scaled version of the LHC collimation system [38]. It consists of three primary collimators (TCP), with lengths of 60 cm made out of AC-150-K; 11 secondary collimators (TCS), with a length of 100 cm made out of AC-150-K; and 5 absorbers (TCL), with a length of 100 cm made out of IT-180.

3.2. Dispersion Suppressor layout

Since the preliminary layout of the FCC-hh ring was based on the LHC, the original magnet positions at the end of cell 8 and cell 10 (fig. 3.2) foresaw only a magnet-free drift space of ∼3.6 m in front of the magnets where the losses occur. In the course of this work it became clear that this would not be sufficient to efficiently protect the magnets (see chapter 5), as it would allow only a single collimator of ∼1.5 m length to be placed there, since space for interconnections and connections for the cryostat have to be considered as well. To provide more space for longer collimators and additional protection devices the layout of the ring was altered to increase the drift space to ∼7.6 m (see fig. 3.3).

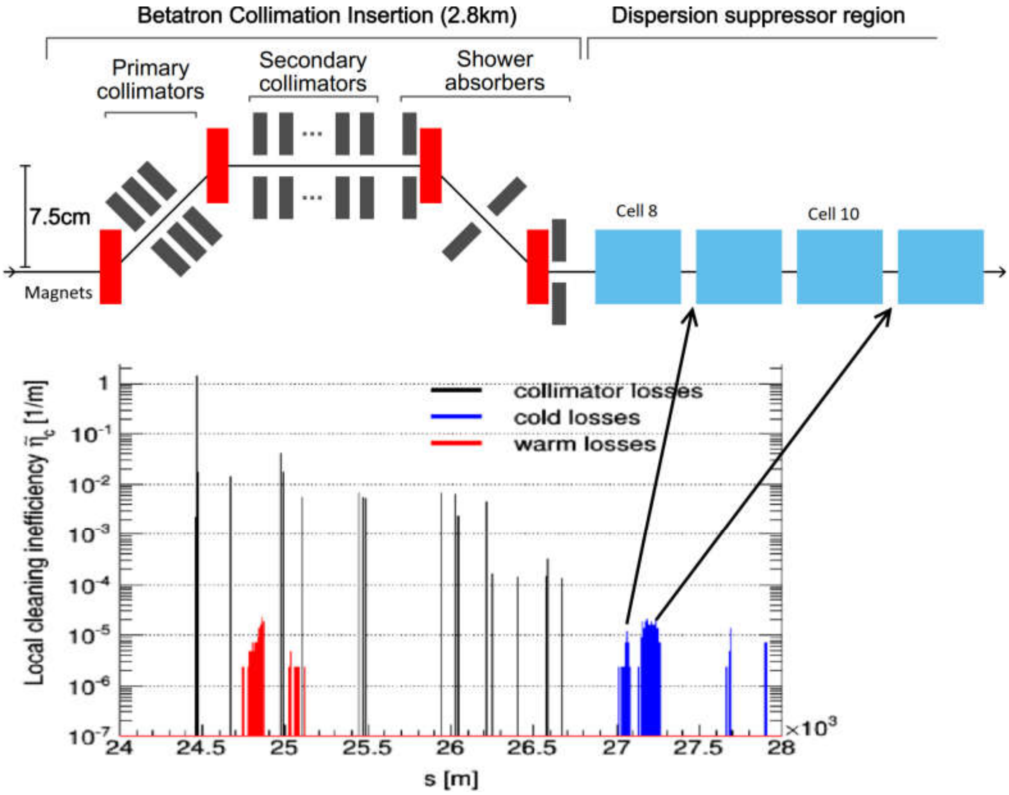


Figure 3.2.: Lossmap made by M. Fiascaris for an early version of the FCC-hh lattice [38] with a diagram of the betatron cleaning insertion indicating the position of the losses (see also section 3.2). The primary collimators are three 60 cm long carbon collimators for horizontal, vertical and skew cleaning. The secondary collimators are a number of carbon collimators at various angles (see [38] for a full description). The shower absorbers in the cleaning insertion consist of tungsten-compound collimators.

3.2.1. Cell structure

Cells 8 to 10 have the same principal magnet structure. As can be seen in fig. 3.3, the first elements after the protection devices are a quadrupole magnet (red), followed by a drift space and dipole magnets (blue).

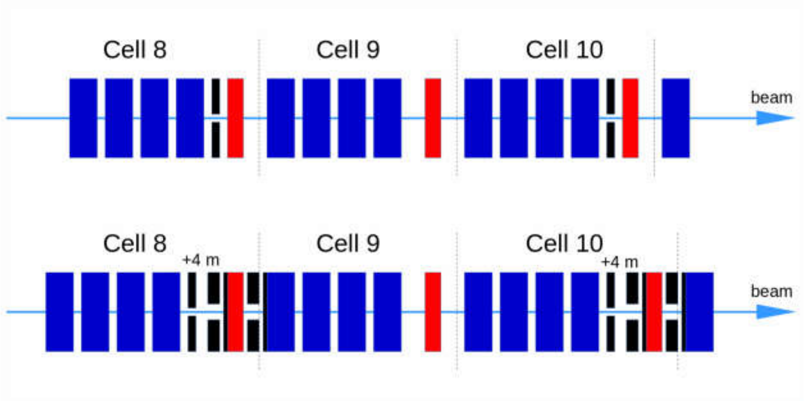


Figure 3.3.: Cell structure in the DS-region. Initial layout (top) and extended layout (bottom). Blue are dipole magnets, red are quadrupole magnets and black are protection devices.

3.3. Beam optics in the Dispersion Suppressor region

As briefly explained in section 2.2, the betatron cleaning section needs very specific beam optics to ensure optimum cleaning efficiency. The β-function in both transverse planes should be large, while the dispersion-function should be comparatively small. This is not the case for the DS collimators, as they essentially perform momentum cleaning. Consequently it is the other way around and a high value for the dispersion-function as well as a small β_x-function is desired. The gap size for the TCLDs for efficient cleaning depends on the distance from the nominal orbit rather than a multiple of the transverse beam size $\sigma(s)$. This is because the dispersion-function shifts the beam orbit for off-momentum particles away from the nominal orbit. It is therefore irrelevant if the transverse beam-size is small, as long as the collimator gap is small enough to intercept the off-momentum orbit. As shown in fig. 3.4 these conditions are well met at the locations of the drift spaces for the TCLDs. This and the following plots of beam optics functions were made using numerical data from MAD-X [28].

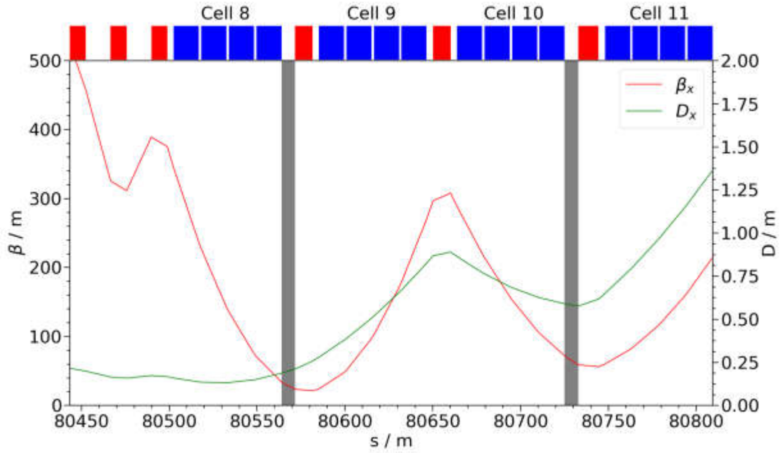


Figure 3.4.: β_x-function and dispersion (D_x) in the dispersion suppressor region after the betatron collimation. The locations of the drift space for the TCLDs is marked in grey.

3.4. Magnet quench limits

As the momentum loss is small, the protons still have close to 50 TeV of energy and therefore a huge damage potential. This is especially problematic since the damage threshold and the magnet quench limits for the superconducting magnets are also potentially much lower compared to the LHC.[8] For superconducting magnets in the LHC the peak power quench limits are estimated to be 25 to 47 mW/cm^3 [39, 40]. Since the 16T magnet structures for the FCC-hh are still under development, no data from quench tests is available, but first estimates done by the magnet-group suggest a quench limit of 5 to 10 mW/cm^3 [41]. This limit already includes a safety margin accounting for variations in real magnet coils like production errors, material impurities and so on. For all further considerations the quench limit was taken to be 5 mW/cm^3.

4. Simulation

4.1. Tracking simulations with MERLIN

4.1.1. Lossmap studies

The layout of the FCC-hh ring changed since the first lossmap simulations shown in fig. 3.2. So new tracking simulations were carried out to get an updated view of the problem. This was done using MERLIN with the current version of the FCC-hh lattice [6]. The collimator gaps were set as listed in table 4.1 without the TCLDs. These gap settings correspond to scaled settings of the HL-LHC. The resulting gaps in millimetre are comparable to the ones in the LHC [11].

As the starting particle distribution a so called direct halo at the horizontal TCP in the betatron cleaning was used. A direct halo is a particle distribution created directly at the beginning of the collimator. As described in [10], there is no advantage in using a particle distribution that fills the whole phase space at the point of injection. The resulting lossmap in the DS region is the same but the particles need to be tracked over many more turns. This is a good approximation for real particle losses in the dispersion suppressor regions, since only particles that undergo single diffractive scattering the horizontal TCP contribute to losses in the DS [10]. Particles are generated only in the areas of the phase space that lies inside the collimator jaws with a maximum impact parameter[1] of 1 µm (see fig. 4.1). All particles are generated with nominal energy ($E = 50\,\mathrm{TeV}$) and with zero vertical offset ($y = 0$) and vertical slope ($y' = 0$) .

The losses are recorded in bins of 10 cm length along the ring. To increase the statistics on smaller losses, $2.44 \cdot 10^9$ particles were tracked

[1] The impact parameter is the horizontal position where the particle hits the collimator, measured from the side of the collimator.

over 250 turns. This resulted in $2.37 \cdot 10^9$ lost particles. The reason that not all tracked particles are lost, is because some of them scatter elastically in a very small angle and it takes many turns for them to actually get lost. Therefore tracking simulations are done for a specified number of turns and the quality of the simulation can be assessed by comparing the number of remaining particles to the number of tracked particles. These should be on the order of ∼5 %.

Table 4.1.: Nominal collimation gap settings for horizontal collimators used for simulations, based on a normalized emittance of ϵ_{rms} =2.2 µm.

Collimator	Nom. gap [σ]
Betatron cleaning	
TCP	7.2
TCSG	9.7
TCLA	12.0
Momentum cleaning	
TCP	21.4
TCSG	25.2
TCLA	27.7
Protection at experiments	
TCTH	13.7
DS protection	
TCLD	35.14

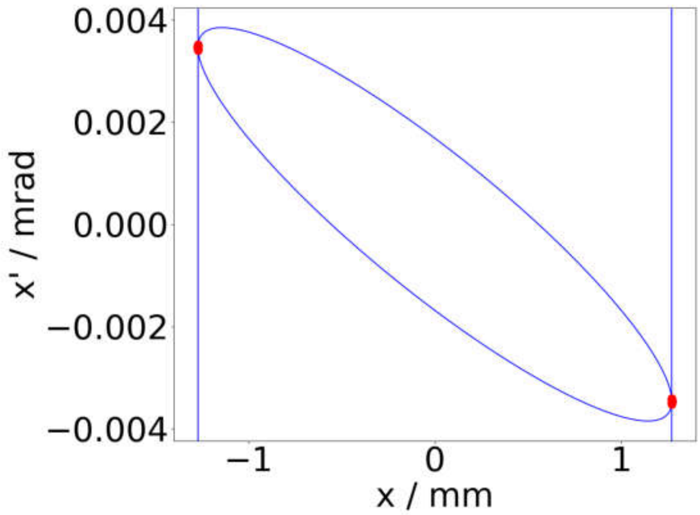


Figure 4.1.: Starting phase space distribution at the TCP for MERLIN simulations. The red points represent the starting particles, the blue ellipse is the 7.2 σ envelope and the vertical lines are the TCP jaw positions. The vertical offset (y) and the vertical slope (y') are 0 for all starting particles.

The resulting lossmap, without TCLDs, is pictured in fig. 4.2. The position along the accelerator is measured from the injection point in IPL (see fig. 1.1). Comparing it to fig. 3.2 reveals some important differences, first of all the cleaning inefficiency is about a factor 10 smaller, but now the losses are more spread and more clusters have formed. Looking at the dispersion function in fig. 4.2, it can be seen that the clusters coincide with a rise in the dispersion. A tolerable amount of cleaning inefficiency in the cold areas is assumed to be $<3 \times 10^{-7}\,\mathrm{m}^{-1}$ for a beam lifetime of 12 minutes for the foreseen FCC beam (see also section 2.2.1) as scaled from the LHC [42], which means the losses in the DS are still about a factor 4 too high.

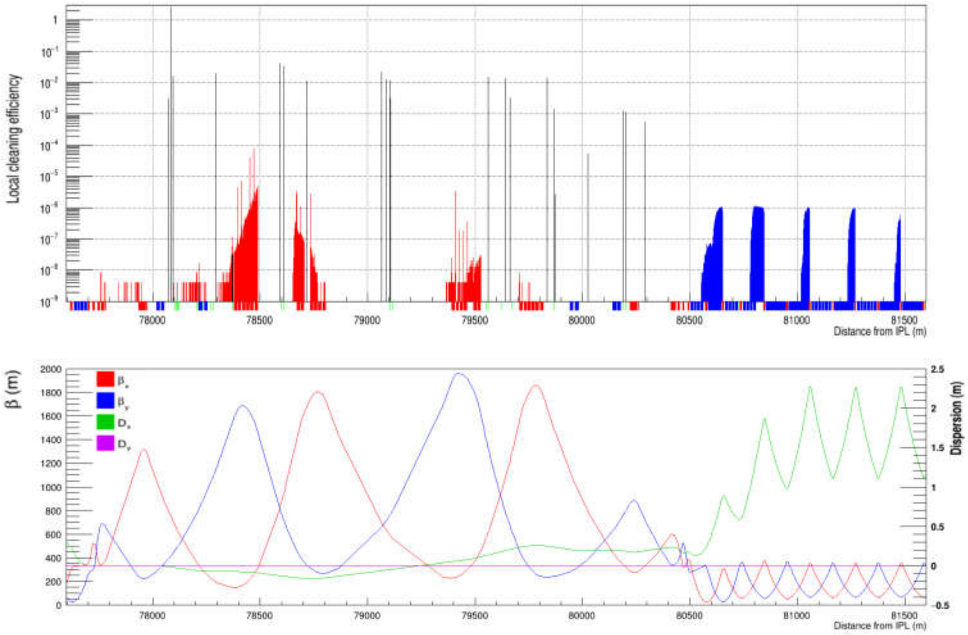


Figure 4.2.: Lossmap of the betatron collimation section without collimators in the DS regions, created with MERLIN. Black are collimator losses, red are warm losses in normal conducting magnets and blue are cold losses in superconducting magnets. The tracking was done with $2.44 \cdot 10^9$ particles over 250 turns which resulted in $2.37 \cdot 10^9$ lost particles.

As a next step a 1 m collimator was introduced in cell 8 (TCLD.8) and cell 10 (TCLD.10) and another tracking simulation was done with the same input settings as described above and $2.49 \cdot 10^9$ particles over 250 turns. The collimator gaps for the TCLDs were set to $35.14\,\sigma$. This results in collimator gaps of 1.3 mm for the TCLD.8 and 1.9 mm for the TCLD.10. These gaps sizes are considered realistic in terms of mechanical implementation, as they are comparable to the ones in the LHC. This resulted in $2.43 \cdot 10^9$ lost particles. The lossmap is shown in fig. 4.3. The TCLDs almost completely mitigate the losses in the cold sections. The remaining losses are of the order of $(2 \pm 1) \times 10^{-8}\,\mathrm{m}^{-1}$ which is assumed sufficient for safe operation of the machine. Another reason for the chosen gap settings of $35.14\,\sigma$ was that both TCLDs reach the same cleaning efficiency. When just the beam particles are considered, as it is done in these tracking studies, it would be possible to collimate all particles with just one TCLD in cell 8. But since every incident particle on the TCLDs generates secondary particle showers, it is important that the losses are shared between the two collimators. In fig. 4.4 a zoom in on the TCLDs is shown. The total cleaning efficiency, normalized over the length of the collimator, is 1.11×10^{-4} for TCLD.8 and 1.12×10^{-4} for TCLD.10. It can be seen that the loss distribution differs and that more losses occur towards the end of TCLD.8 as compared to TCLD.10. It is therefore reasonable to assume that the protection of the magnets from showers is more challenging in cell 8.

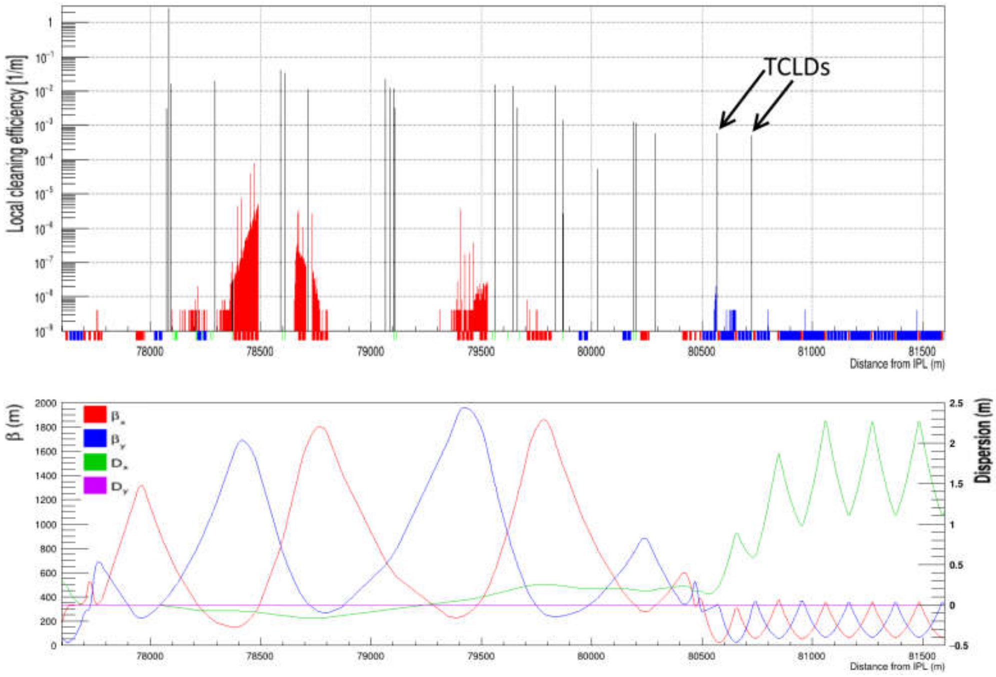


Figure 4.3.: Lossmap of the betatron collimation section with TCLDs in the DS regions, created with MERLIN. Black are collimator losses, red are warm losses and blue are cold losses. The tracking was done with $2.49 \cdot 10^9$ particles over 250 turns which resulted in $2.43 \cdot 10^9$ lost particles.

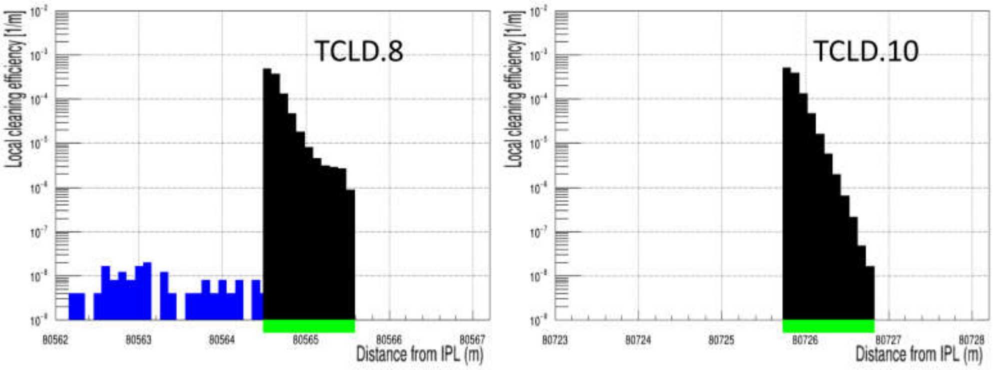


Figure 4.4.: Magnification of the TCLD positions from fig. 4.3.

4.1.2. Particle distribution for shower simulations

To simulate the secondary showers using FLUKA it is necessary to have a good approximation of the particle distribution that hits the TCLDs. For the HL-LHC studies, for example, this was done by creating a distribution of the non-elastic interactions in the TCPs via a particle tracking code. This distribution was then used to generate particles in FLUKA, transport them through a sophisticated model of the complete betatron cleaning section of the LHC to the DS region and record the energy deposition there. This approach was not possible for this work, as no model of the betatron cleaning section of the FCC-hh exists at the moment. A new method for creating the particle distribution was developed using MERLIN. The idea is to do a normal lossmap study with MERLIN, but instead of just recording the losses at the TCLDs, the whole particle information (x, x', y, y', s, p) is recorded for every particle that touches the TCLDs. Since it is not recommended for FLUKA to start particles right at the surface of an object, the recorded particles were then tracked back to the beginning of the drift in cell 8. This distribution was then loaded into FLUKA as primary particles.

4.2. Energy Deposition studies with FLUKA

4.2.1. General simulation settings

The complete input file for FLUKA can be found in appendix B.2. The default parameters for simulations can be set using predefined option sets for specific problems, with the DEFAULTS command. For this work, all simulations were carried out using the PRECISIOn set[31]. This is the set recommended for precision simulations. Particle transport thresholds are set to 100 keV (neutrons 10 µeV). The threshold for multiple scattering is set to the minimum allowed energy for all charged particles. Heavy fragment transport is activated (for full specifications, see [31]).The transport cutoffs for electrons, positrons and photons was set to 1 MeV total energy using EMFCUT.
The limit for maximum center of mass momentum was set to 55 TeV/c using the PHYSICS command to initialize the DPMJET-3 generator.
Biasing options were used to reduce the simulation time. The secondary particle production from hadronic events was reduced by a factor of 0.2 using the BIASING command, as suggested by the FLUKA simulation group at CERN. This is automatically accounted for in the weight (importance) of the particles produced. Biasing of the EM-cascade was done using EMF-BIAS with SDUM "LPBEMF" which activates leading particle bias (LPB). This was used for all EM-interactions for energies below 1 GeV. LPB randomly selects one of the produced particles in EM-interactions with a probability proportional to the particle energy. The other particles are then discarded. The weight of the retained particle is then adjusted accordingly.

4.2.2. Geometry

To simulate energy deposition in matter with spatial resolution, 3D-models of all devices are necessary. Since the main focus is the energy deposition in the magnets downstream of cell 8 and cell 10 in the DS-region, a simple model of this area was constructed including all necessary components.

General considerations

To reduce the simulation time the size of the overall geometry, and therefore the number of interactions to simulate, has to be chosen carefully. For this reason three assumptions, based on the tracking studies (see section 4.1), were made:

1. The first collimator in cell 8 will intercept almost all beam protons that would hit the magnets before the collimator in cell 10, which will intercept the remaining ones (fig. 4.3).

2. The particle showers induced by these protons will be lost within the next two magnets.

3. The problem in cell 8 and cell 10 is essentially the same, which more challenging impact parameters in cell 8 (see fig. 4.4). It is therefore sufficient to simulate only this case.

Magnets

Since the superconducting magnets were still in development, no real model was available for the quadrupoles and dipoles. For this reason a simplified model was created based on current LHC magnet designs [43] and available resources of proposed coil designs for the FCC-hh [44] and confirmed with the group working on magnet development. The quadrupole model has a length of 10 m and the dipole has a length of 15 m. In fig. 4.5 a visualization of the model is shown. The main geometric difference of the two magnets is the position of the coils (green). The material of the superconducting coils was assumed to be a mixture of 50 % Cu and 50 % Nb_3Sn. The spacers between the coils are made of titanium (dark grey). On the inside of the coils is the cold-bore, which separates the liquid helium from the beam pipe, and the beam pipe itself, which was assumed to be an elliptical cylinder (violet). Both are assumed to be steel (section 4.2.3). In reality the beam pipe is much more complex [45], but this has no impact on the energy deposition. Everything else on the outside of the coils was assumed to be iron as it is of low importance for the energy deposition in the coils.

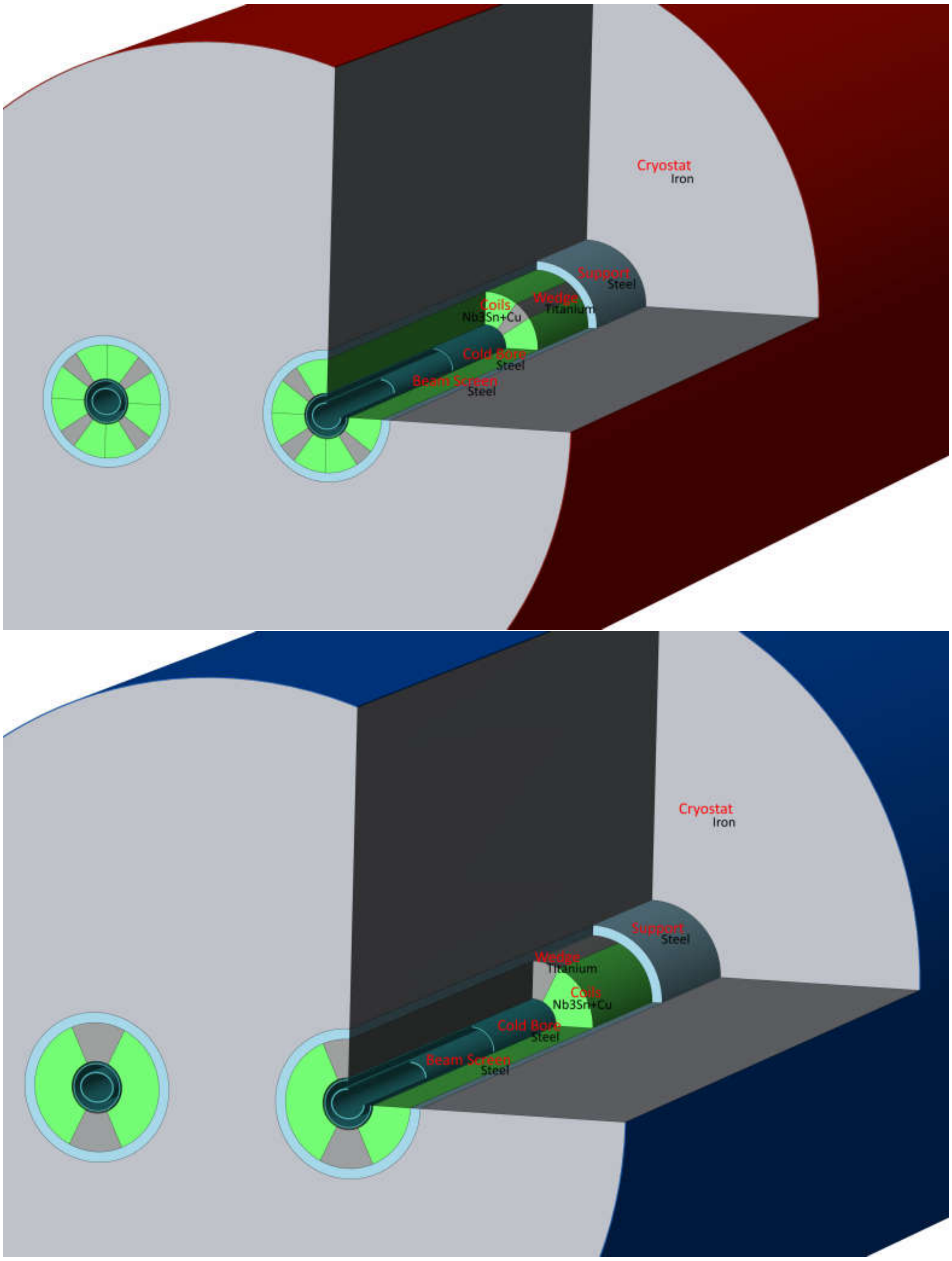


Figure 4.5.: Simplified 3D models of the quadrupole (top) and dipole (bottom) magnets. The green area represents the Nb_3Sn superconductor coils. Visualized with SimpleGeo [35].

Collimators and masks

The design of the collimators was taken to be two blocks of IT180(section 2.2.5) of varying length with a tapering of 10 cm^2 on each end (fig. 4.6, left). The shape of the masks was assumed to be a simple cylinder made of IT-180 with a hole the size of the beam pipe in it.

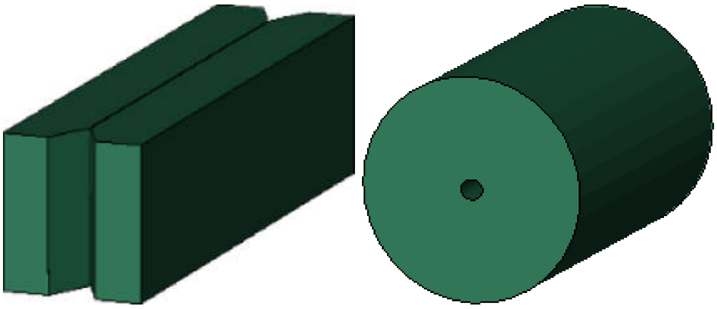


Figure 4.6.: Simplified 3D models of a collimator with tapering (left) and a mask (right). Visualized with SimpleGeo [35].

Visualization of the full geometry

The region of interest in cell 8 consists of 7.6 m empty drift space followed by a quadrupole, 3.6 m of drift space and a dipole. The whole geometry is enclosed in a box as required for FLUKA simulations (see section 2.4.3). The full geometry is pictured in Figure 4.7. The visualization, and a validation against overlap of solid bodies, was done using SimpleGeo.

[2]A collimator is tapered to reduce beam instabilities due to impedance.

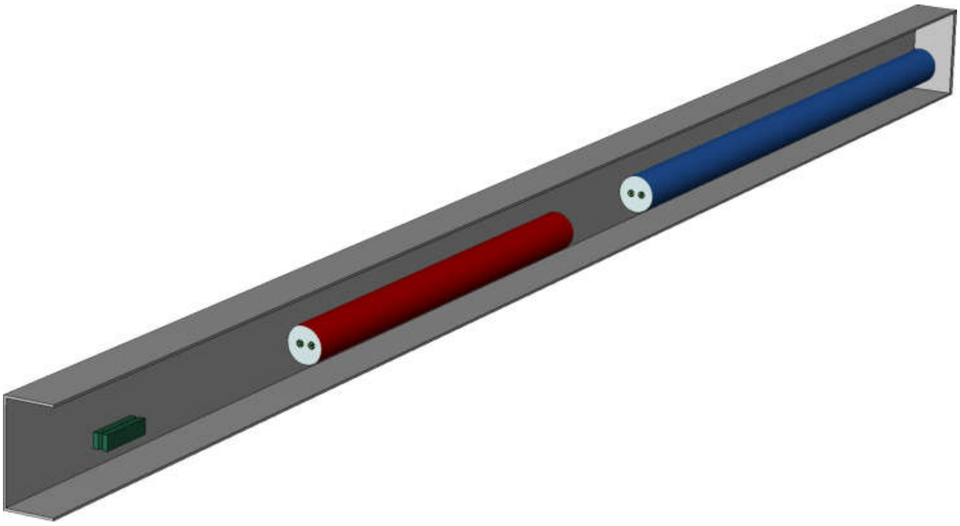


Figure 4.7.: Complete 3D geometry for simulations with the quadrupole in red, the dipole in blue and a collimator in green (the enclosing box is truncated). Visualized with SimpleGeo [35].

4.2.3. Material description in FLUKA

For the correct description of the materials used in the magnets and collimators the following materials and compounds were defined as explained in section 2.4.3:

- Manganese, Mn ($Z = 25$), $\rho = 7.21\,\mathrm{g/cm^3}$
- Chromium, Cr ($Z = 24$), $\rho = 7.18\,\mathrm{g/cm^3}$
- Niobium, Nb ($Z = 41$), $\rho = 8.57\,\mathrm{g/cm^3}$
- Steel, $\rho = 8.14\,\mathrm{g/cm^3}$, (56 % Fe, 20 % Cr, 12 % Mn, 11 % Ni, 1 % Cu)
- Inermet®-IT180, $\rho = 18.0\,\mathrm{g/cm^3}$, (95 % W, 2.5 % Ni, 1.5 % Cu)
- Nb_3Sn, $\rho = 8.95\,\mathrm{g/cm^3}$, (3 Nb, 1 Sn)
- Nb_3Sn+Cu, $\rho = 8.95\,\mathrm{g/cm^3}$, (50 % Nb_3Sn, 50 % Cu)

These materials are then assigned to the corresponding region using the ASSIGNMAT command.

4.2.4. USER routines

To further adjust the simulation it was necessary to use so called "user routines" to interface with the FLUKA code in a more advanced way. The user routines are FORTRAN77 interfaces that have access to some part of the FLUKA code. This was used for the generation of the starting particle distribution and for the definition of the magnetic field inside the quadrupole and dipole.

Magnetic fields

To set a magnetic field the commands MAGNFIELD and ASSIGNMAT are used. In ASSIGNMAT regions that contain a magnetic field can be defined, only these are considered for magnetic field tracking. MAGNFIELD is used to set parameters for tracking, like maximum angle traveled in one step (set here to 20 degrees), the maximum error in identifying a boundary crossing from one region to the next (set here to 0.2 cm) and the minimum step length, if the angle constraint forces a reduced step size (here 0.1 cm). It is possible to create a uniform magnetic field in all regions specified by ASSIGNMAT, but since different fields in different regions are needed, the user routine MAGFLD was used. Furthermore the user routine USRGCALL, which is called during the initialization process FLUKA, was used to pass parameters to MAGFLD, thus making it more interactive for changes in the input file. A perfect quadrupole field with a field gradient of 219.4 T/m was created in the vacuum, the beam screen and the cold bore inside the quadrupole magnet. In the titan wedges a residual, linear decreasing field was assumed. For the dipole a simple uniform 16 T dipole field was assumed in the vacuum, the beam screen and the cold bore of the dipole magnet. A sketch of the magnetic fields inside the dipole and the quadrupole can be seen in fig. 4.8. The user routine file can be found in appendix B.1.2

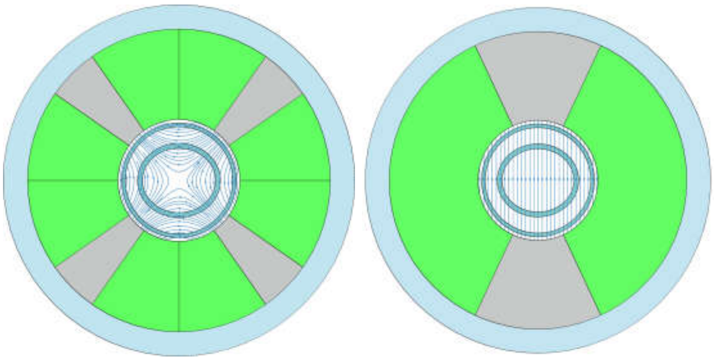


Figure 4.8.: Sketch of the quadrupole (left) and dipole (right) magnetic fields.

User defined source

Normally a particle beam is created in FLUKA using the BEAM and BEAMPOS commands. If the beam shape is more complex than Gaussian, the user routine SOURCE should be used. With the SOURCE routine each particle can be specified independently in terms of position (x,y,z), momentum direction(x',y',z') and absolute momentum (p). Since FORTRAN77 cannot read arbitrary formatted files, a C-program program was called from within the SOURCE routine, to act as a file-parser for the particle distribution file. To ensure a correct distribution sampling, independent from the number of requested primaries, the build in random generator of FLUKA is called by the C-program each time the SOURCE routine requests a particle. In appendix B.1.1 the file-parser and the user routine can be found.

4.2.5. Scoring in FLUKA

Scoring quantities in FLUKA is done via so called detectors. To score energy deposition with spatial resolution the USRBIN detector is very useful. USRBIN detectors create a 3D mesh of bins in which the desired quantity is scored, this is completely independent from the surrounding geometry. The mesh can be either a in Cartesian coordinates or in cylinder coordinates. To score the energy deposition in the quadrupole coils, a cylinder mesh was used. With 25 radial bins ranging from 2.5 cm to 15.0 cm, 180 angular bins and 200 bins along the magnet from 0 to 10 m. For scoring in the dipoles, a similar mesh was used, but with 150 bins along the magnet from 0 to 15 m. The resulting energy deposition in each bin is given in GeV/cm^3 per primary particle. A visualization of such a binning is shown in fig. 4.9

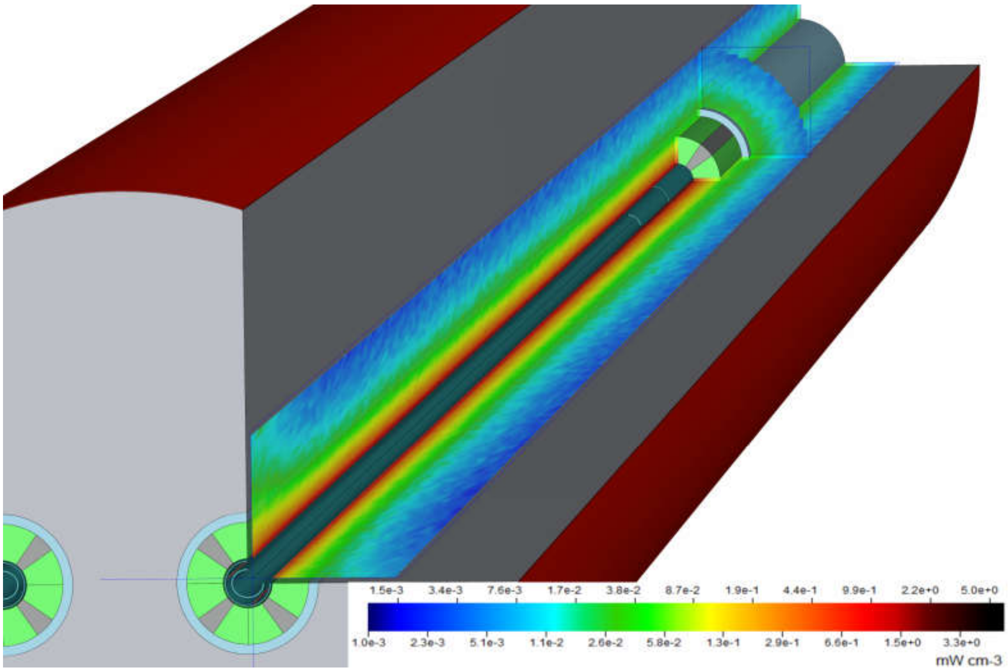


Figure 4.9.: A 3D representation of the meshgrid energy deposition binning in the quadrupole magnet. For this representation the radial binning was translated into cartesian coordinates. Visualized with SimpleGeo [35].

4.2.6. Number of primary particles and Independent simulations

To decrease fluctuations in the simulation results it is necessary to track a "large enough" number of particles. "Large enough" in this case strongly depends on the simulation setup and geometry of the problem. To estimate the statistical error on these fluctuations (and in turn get an estimate of the number of primary particles needed), it is important to run several independent simulations. This is done by starting a number of FLUKA simulation runs with the same input file, but different random seeds for each run. The scoring result in one detector bin is then an average over all primaries from one run. To get the final result, the results of all runs are then averaged. For the presented simulations the number of primaries was first set to 400 and increased to 1000 for subsequent simulations. The number of statistically independent runs was set to 1000 and later to 4000.

5. Evaluation and Performance

The main loss scenario considered for the DS protection is the case of 12 min beam lifetime. Faster losses are not expected to occur at this position (see also section 2.2.1). A circulating beam with 12 min of lifetime will only be kept in the machine for a few seconds at most until it is either stabilized or dumped. During this time the losses occur continuously, which makes peak power density inside the coils the key parameter for the protection evaluation. A 12 min beam lifetime translates to 1.47×10^{12} lost protons per second. Combining this with the cleaning efficiency of TCLD.8 of 1.11×10^{-4} from section 4.1, gives a loss rate of 1.64×10^{8} protons per second for cell 8 in the DS region. With this, the energy deposition values can be normalized to give the energy deposition in mW/cm^3.

5.1. Initial simulations

This was done at first for three different cases:

1. Using only a single 1 m collimator (TCLD.8) at the beginning of the drift space.
2. Including a 0.5 m mask in front of the quadrupole and dipole.
3. Introducing a secondary collimator (TCLDS.8) in front of the quadrupole with a gap of double the size of the TCLD.8 (2.60 mm or 79.22 σ).

The last case is shown in fig. 5.1. The FLUKA simulations for case 1 and case 2 were done using 1000 runs with 400 primaries = $4 \cdot 10^5$ primary particles. For case 3 1000 runs with 1000 primaries = $1 \cdot 10^6$ primary particles were used.

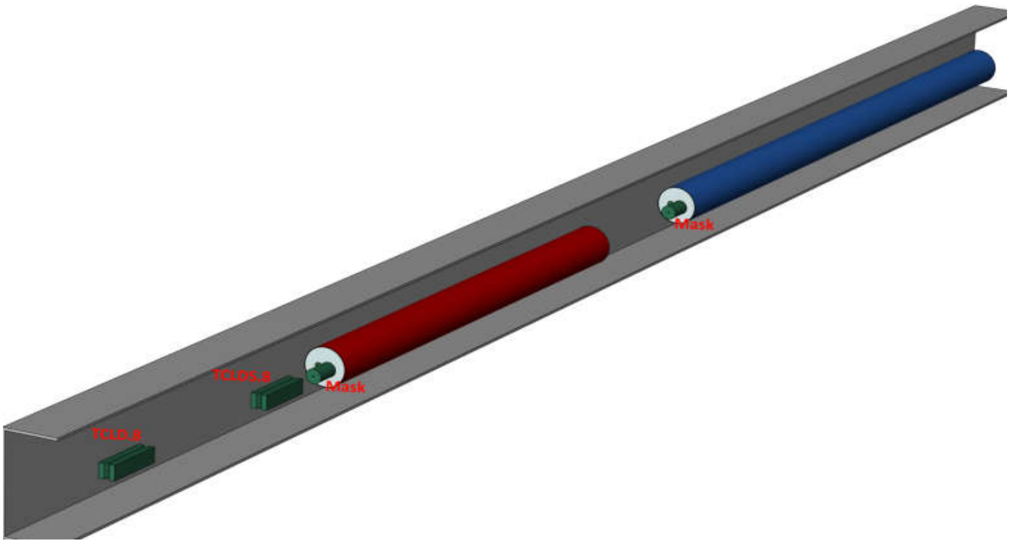


Figure 5.1.: Geometry for first energy deposition studies with 1 m TCLD.8, 1 m TCLDS.8, and 0.5 m masks, (the enclosing box is truncated). Visualized with SimpleGeo [35].

The peak power density along the magnets was taken to be the highest power density value of all bins for every 5 cm along the quadrupole and 10 cm along the dipole. The resulting peak power density diagrams for the quadrupole (top) and dipole (bottom) are shown in fig. 5.2. As stated before in section 3.4 the estimated quench limit is 5 mW/cm^3. The deposition is highest in the beginning of the quadrupole when no mask is present (red curve, top plot). With a mask included these front losses can be reduced by a factor 5 (black curve, top plot). The addition of the TCLDS.8 collimator further reduces the losses over the whole magnet (blue curve, top plot). In the dipole the deposition is generally less than in the quadrupole, because it is much further away from the TCLD.8. Adding a mask in the front still reduces the losses (black curve, bottom plot), but the inclusion of the TCLDS.8 has no impact on the dipole energy deposition (blue curve, bottom plot). This is of no surprise, as the shower particles that reach the dipole are very close to the ideal beam, until the magnetic field in the dipole bends them into the wall of the beamscreen. Considering the magnet limits (shaded area in both plots) case 2 and case 3 seem at first sufficient to protect the magnets.

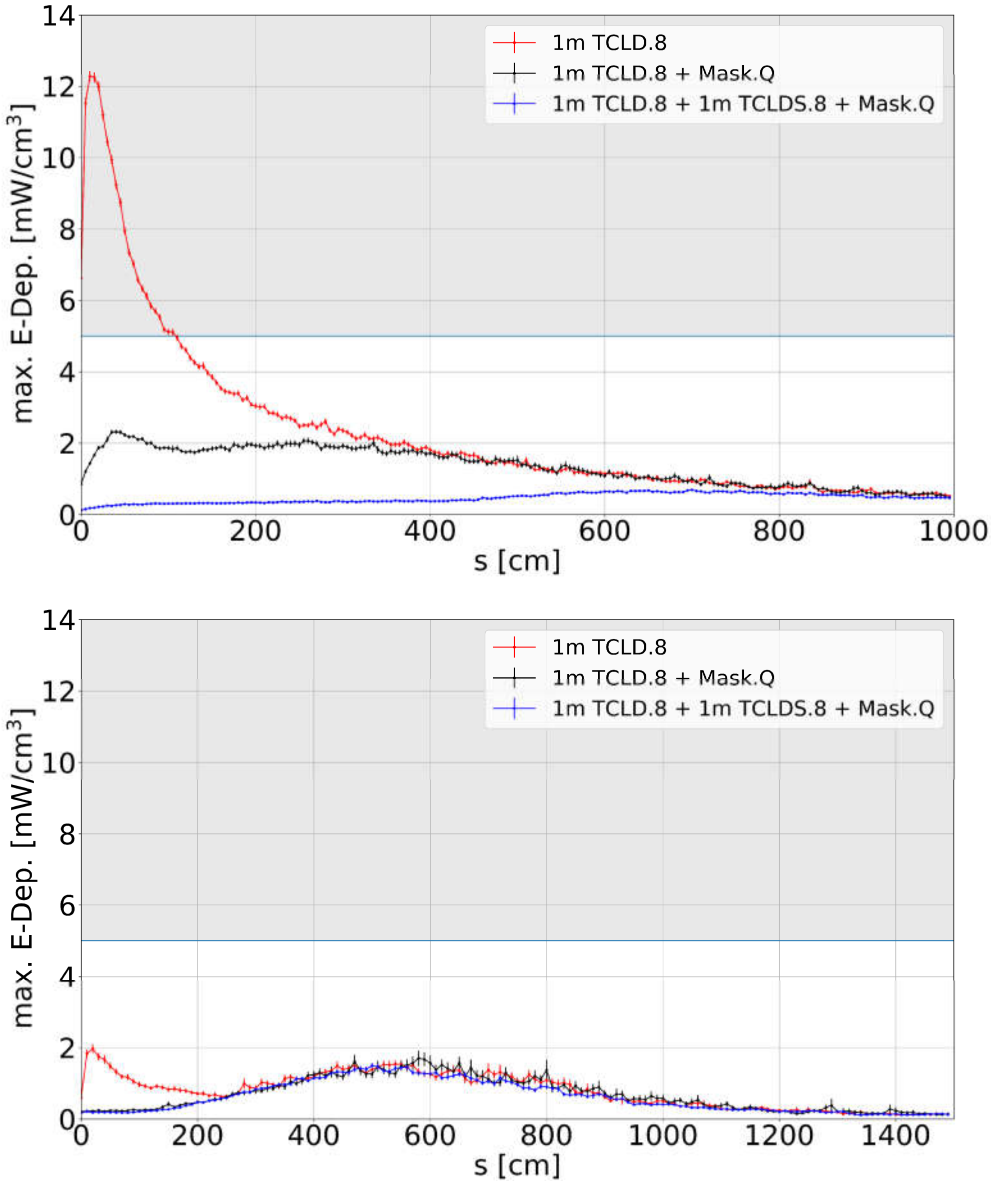


Figure 5.2.: Peak power density inside the coils along the quadrupole (top) and dipole (bottom) in cell 8 for different protection designs. The quench limit is shaded in gray. The energy deposition is normalized for a loss rate of 1.64×10^8 protons per second on the TCLD.8 collimator.

5.2. Correction for underestimations

A study done by R. Bruce et. al. [10] comparing the measured loss rates in the LHC to lossmap simulations found an underestimation by a factor 2 to 3 in the DS regions. Also, for the current simulations a perfect machine is considered, without any misalignment of collimators, imperfections, or magnetic field deviations. It was therefore deemed necessary to include a factor 4 as safety margin on top of the energy deposition results. This increase is plotted as shaded areas in fig. 5.3. The safety margin is only plotted for case 2 and 3, as case 1 did not meet the requirements without it. Case 3 with the TCLDS.8 included manages to keep the peak power deposition below 3 mW/cm^3 (blue shaded area, left plot), whereas in case 2 the maximum power deposition is 9 mW/cm^3 (dark grey shaded area, left plot). In the dipole (right plot) case 2 (blue shaded area) and case 3 (dark grey shaded area) remain very similar and for both cases the peak deposition is above the lower quench limit of 5 mW/cm^3.

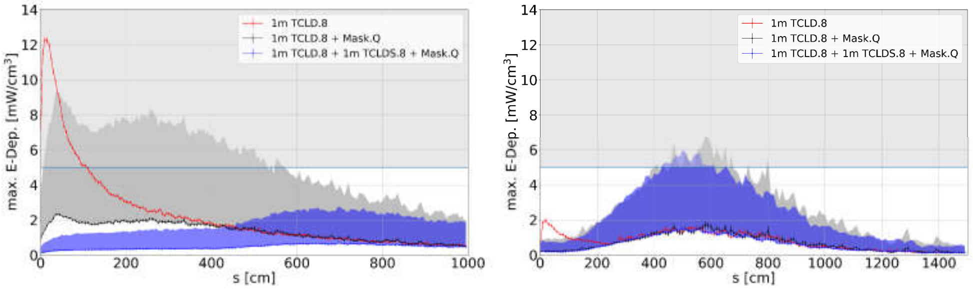


Figure 5.3.: Peak power density inside the coils along the quadrupole (left) and dipole (right) in cell 8 for different protection designs for a loss rate of 1.64×10^8 protons per second on the TCLD.8 collimator, with included safety margin of a factor 4 (shaded areas).

5.3. Further corrections and final protection design

Before the development of an updated protection design, to reduce the losses in the dipole, it was suggested by R. Bruce from the CERN collimation group, who was also working on FCC-hh collimation and HE-LHC collimation at that time, to further increase the safety margin to a factor of 8 [46]. The reason for this was that more comparisons had be done between loss measurements and simulations which now showed a difference of a factor 5 to 6 in the DS regions instead of the aforementioned factor of 2 to 3. This was therefore considered in the development of the final protection design. Another restriction for the new design was that the TCLD.8 should not be changed, if possible, as this would affect other studies which are based on the FCC-hh layout with 1 m TCLD collimators. The changes for the final protection design are as follows:

- The TCLDS.8 length was increased to 1.5 m, and the gap reduced to 1.40 mm (42.66 σ). This does not affect the FCC-hh layout as this is still a factor 1.5 larger than the largest collimator gap in the cleaning sections (table 4.1).
- A new collimator (TCLDT.8) was introduced between the quadrupole and the dipole with a length of 1.5 m and a gap size of 1.45 mm (48.62 σ).
- To create enough space for the TCLDT.8, the mask in front of the dipole was shortened to 0.15 m.

The complete geometry is shown in fig. 5.4 and the parameters of all protection devices are given in table 5.1. This geometry was simulated in FLUKA using 4570 runs with 1000 primaries = $4.57 \cdot 10^6$ primary particles. Figure 5.5 shows the resulting peak power depositions in the quadrupole and dipole (blue curves). These include the proposed safety margin of a factor 8. The previous design is also included for comparison (red curves). The improved design is able to reduce the peak deposition in the quadrupole by another factor of 2 in the most exposed area (top plot, red vs blue curve). With the factor 8 safety margin the previous design is not sufficient to keep the deposition below 5 mW/cm^3. A even bigger difference can be seen in the dipole magnet (bottom plot). Here the peak power deposition is a factor 3 lower for the updated design (bottom plot, red vs blue curve), reducing the deposition to values below the lower quench limit. The updated protection design is sufficient to protect the superconducting magnets in the dispersion suppressor region of the FCC-hh.

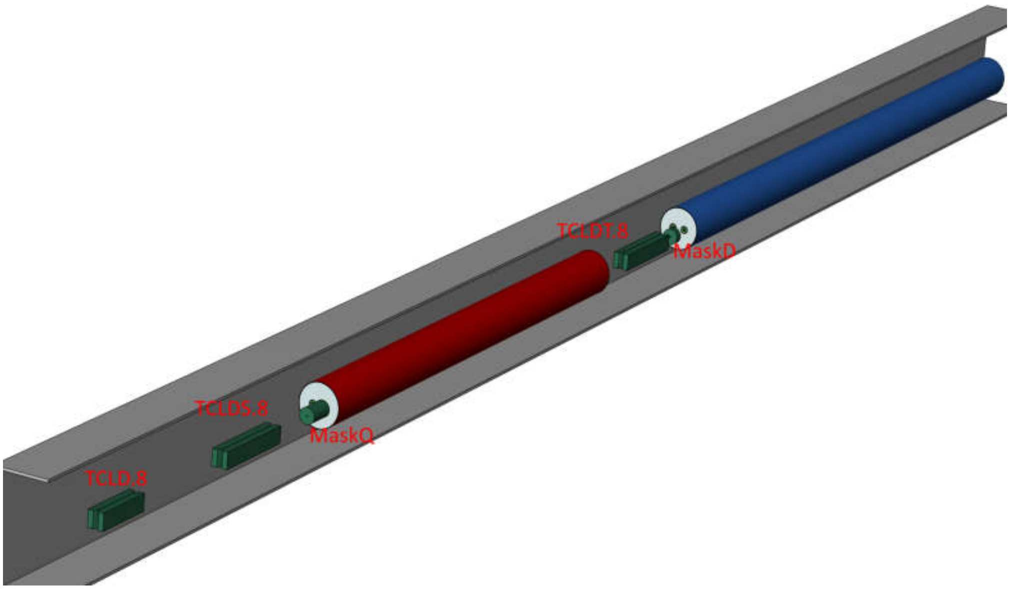


Figure 5.4.: Geometry with the final DS protection design. With 1 m TCLD.8, 1.5 m TCLDS.8, 0.5 m MaskQ, 1.5 m TCLDT.8 and 0.15 m MaskD, (the enclosing box is truncated). Visualized with SimpleGeo [35].

Table 5.1.: Parameters of the protection devices used in the final design. It can be seen from the gap sizes in σ that the collimation hierarchy is not violated.

Device	length / m	nom. gap / mm	nom. gap / σ [1]
TCLD.8	1.0	1.30	35.14
TCLDS.8	1.5	1.40	42.66
MaskQ	0.5	150.00	
TCLDT.	1.5	1.45	48.62
MaskD	0.15	150.00	

[1] (There are no σ values for the gaps of the masks, as they have the full aperture of the beampipe as opening.)

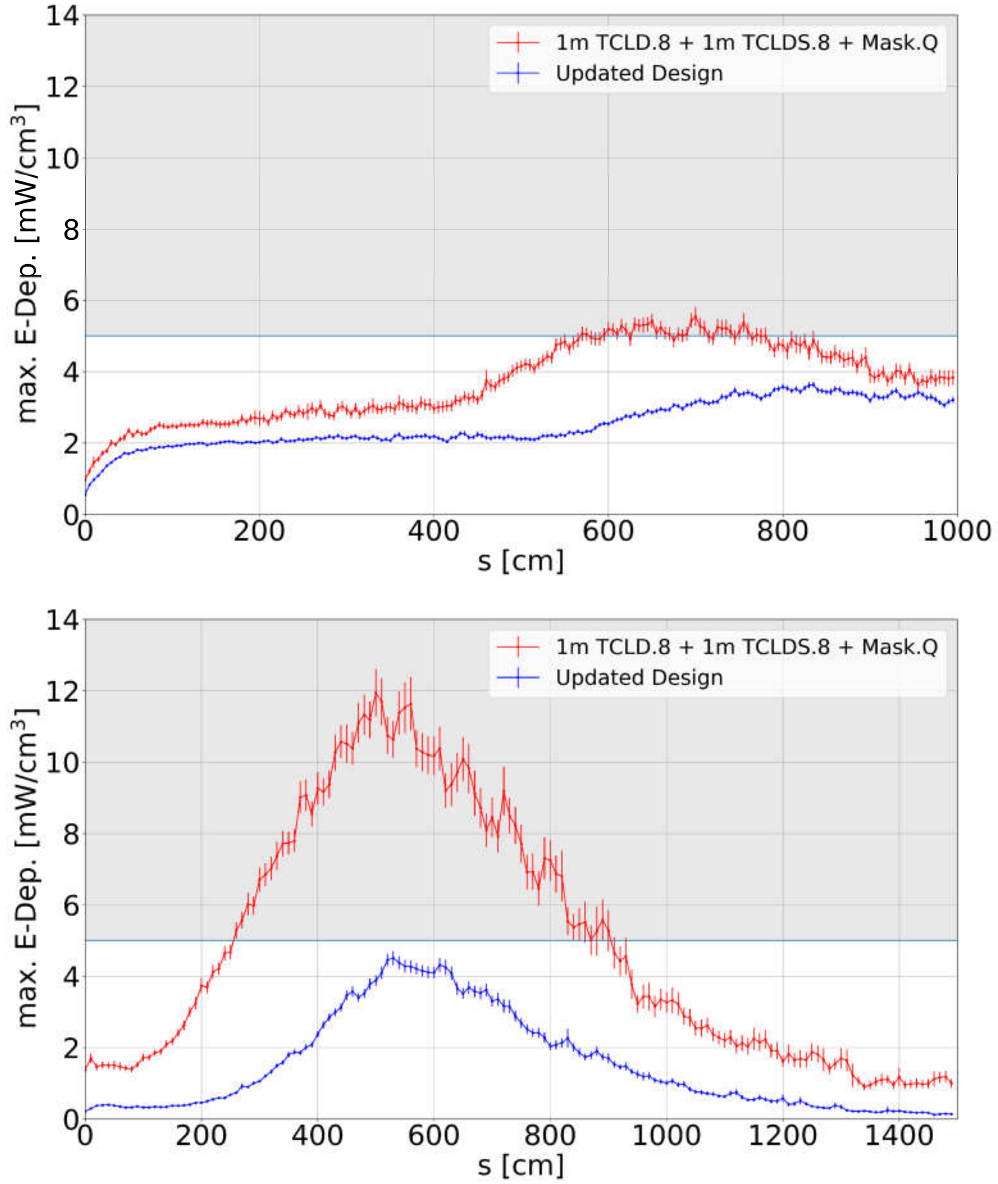


Figure 5.5.: Peak power density along the quadrupole (top) and dipole (bottom) in cell 8 for the final protection design and the previous solution with a factor 8 safety margin included. Normalized for a loss rate of 1.64×10^8 protons per second on the TCLD.8 collimator.

6. Conclusion

To protect superconducting magnets from beam losses is of grave importance. An efficiently performing protection design is necessary for save operation of an accelerator.
In this thesis the special case of beam losses in the dispersion suppressor regions of the FCC-hh was investigated. Lossmaps were created using MERLIN to estimate the severity of particles lost in the dispersion suppressor. Furthermore the effects of beam optics in that regions on particle losses were investigated and suitable locations for protection placement was determined. A protection design was developed consisting of a 1 m collimator (TCLD) with a gap of 35.14 σ, a 1.5 m secondary collimator (TCLDS) with a gap of 42.66 σ, a 50 cm mask (MaskQ) in front of the first quadrupole downstream of the collimators, a 1.5 m tertiary collimator (TCLDT) between the quadrupole and the subsequent dipole and a 15 cm mask (MaskD) in front of the dipole. The material for all collimator yaws and masks was taken to be Inermet®-IT180 [22] (see fig. 5.4). This design was then tested using FLUKA simulations for energy deposition estimations. This was done using particle distributions created from tracking studies with MERLIN and assuming a beam loss scenario of 12 min beam lifetime for the FCC beam with $1.06 \cdot 10^{15}$ protons per beam with an energy of 50 TeV. As suggested by R. Bruce [46], a safety margin of a factor 8 was then added on top of the results to account for differences in measured beam losses and simulated beam losses (as experienced in the LHC), as well as to take into account possible machine imperfections. The resulting peak power deposition in the quadrupole and dipole magnets is below the tentative minimum quench limit for the superconducting FCC-hh magnets of 5 mW/cm^3 (see section 3.4). The peak power deposition in the quadrupole is (3.64 ± 0.08) mW/cm^3 and in the dipole (4.51 ± 0.19) mW/cm^3. Hence, the design is sufficient to protect the magnets from power deposition induced quenches and shows that the presented design is well

suited for protecting the dispersion suppressor regions.

Bibliography

[1] Georges Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Phys. Lett.* B716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020. arXiv: 1207.7214 [hep-ex].

[2] Serguei Chatrchyan et al. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Phys. Lett.* B716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].

[3] *FCC Homepage*. Feb. 2017. URL: https://fcc.web.cern.ch.

[4] O. Bruening et al. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. URL: https://cds.cern.ch/record/782076.

[5] Daniel Schulte. *Preliminary Collider Baseline Parameters: Deliverable D1.1*. Tech. rep. CERN-ACC-2015-0132. Geneva: CERN, Sept. 2015. URL: https://cds.cern.ch/record/2059230.

[6] Antoine Chancé et al. "Updates on the Optics of the Future Hadron-Hadron Collider FCC-hh". In: *Proceedings, 8th International Particle Accelerator Conference (IPAC 2017): Copenhagen, Denmark, May 14-19, 2017*. 2017, TUPVA002. DOI: 10.18429/JACoW-IPAC2017-TUPVA002.

[7] G. Apollinari et al. "Chapter 1: High Luminosity Large Hadron Collider HL-LHC. High Luminosity Large Hadron Collider HL-LHC". In: *CERN Yellow Report* arXiv:1705.08830. 5 (May 2017). 21 pages, chapter in High-Luminosity Large Hadron Collider (HL-LHC) : Preliminary Design Report, 1–19. 21 p. URL: https://cds.cern.ch/record/2120673.

[8] J. B. Jeanneret et al. *Quench levels and transient beam losses in LHC magnets*. Tech. rep. LHC-Project-Report-44. CERN-LHC-Project-Report-44. Geneva: CERN, May 1996. URL: https://cds.cern.ch/record/308241.

[9] T. Trenkler and J. B. Jeanneret. "The Principles of Two Stage Betatron and Momentum Collimation in Circular Accelerators". In: *Part. Accel.* 50.CERN-SL-95-03-AP. LHC-NOTE-312. CERN-LHC-Note-312. 4 (Feb. 1995), 287–311. 23 p. URL: https://cds.cern.ch/record/277606.

[10] R. Bruce et al. "Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider". In: *Phys. Rev. Spec. Top. Accel. Beams* 17.arXiv:1409.3123 (Sept. 2014), 081004. 16 p. URL: https://cds.cern.ch/record/1755099.

[11] Maria Fiascaris et al. "First Design of a Proton Collimation System for 50 TeV FCC-hh". In: *Proceedings, 7th International Particle Accelerator Conference (IPAC 2016): Busan, Korea, May 8-13, 2016*. 2016, WEPMW006. DOI: 10.18429/JACoW-IPAC2016-WEPMW006.

[12] K. Wille and J. McFall. *The Physics of Particle Accelerators: An Introduction*. Oxford University Press, 2000. ISBN: 9780198505495.

[13] Helmut Wiedemann. *Particle accelerator physics; 4th ed.* Berlin: Springer, 2015.

[14] K. Flottmann. "Some basic features of the beam emittance". In: *Phys. Rev. ST Accel. Beams* 6 (2003), p. 034202. DOI: 10.1103/PhysRevSTAB.6.079901,10.1103/PhysRevSTAB.6.034202.

[15] James Molson et al. "Status of the FCC-hh Collimation System". In: *Proceedings, 8th International Particle Accelerator Conference (IPAC 2017): Copenhagen, Denmark, May 14-19, 2017*. 2017, MOPAB001. DOI: 10.18429/JACoW-IPAC2017-MOPAB001.

[16] Daniel Schulte. "FCC-hh Design Highlights". In: *ICFA Beam Dyn. Newslett.* 72 (2017), pp. 99–109.

[17] R. Schmidt et al. "Beam Loss Scenarios and Strategies for Machine Protection at the LHC". In: *AIP Conf. Proc.* 693.LHC-Project-Report-665. CERN-LHC-Project-Report-665 (Aug. 2003). revised version number 1 submitted on 2003-08-25 11:18:46, 184–187. 5 p. URL: https://cds.cern.ch/record/638593.

[18] *A cutaway of the LHC Secondary Collimator*. accessed on August 28th, 2017. URL: https://lhc-collimation-project.web.cern.ch/lhc-collimation-project/coll-design-meeting.htm.

[19] E. Krzyzak. *Properties of AC150k*. Tech. rep. CERN, May 21, 2014. URL: https://indico.cern.ch/event/283843/contributions/644152/attachments/523657/722288/Col.Wor.Gr.-CFC_results.pdf.

[20] E. Quaranta. *Status of materials R&D*. Tech. rep. CERN, July 21, 2014. URL: https://indico.cern.ch/event/330359/.../Status_of_material_RD_AdColMat_revEQ.pptx.

[21] Elena Quaranta et al. "Towards Optimum Material Choices for the HL-LHC Collimator Upgrade". In: *Proceedings, 7th International Particle Accelerator Conference (IPAC 2016): Busan, Korea, May 8-13, 2016*. 2016, WEPMW031. DOI: 10.18429/JACoW-IPAC2016-WEPMW031.

[22] *Inermet®-IT180, material composition*. accessed on September 12th, 2017. URL: https://www.plansee.com/en/materials/tungsten-heavy-metal.html.

[23] Elena Quaranta et al. "Investigation of collimator materials for the High Luminosity Large Hadron Collider". Presented 18 Jul 2017. PhD thesis. Milan Polytechnic, June 2017. URL: https://cds.cern.ch/record/2276659.

[24] M. Cauchi and W. Hohenauer. *Thermal characterization of INERMET180*. Tech. rep. Austrian Institue Of Technology, Sept. 2013. URL: https://cds.cern.ch/record/1600002.

[25] Halina Abramowicz. "Diffractive scattering". In: *eConf* C0406271 (2004), MONT04. arXiv: hep-ex/0410002 [hep-ex].

[26] James Molson. “Proton scattering and collimation for the LHC and LHC luminosity upgrade”. PhD thesis. University of Manchester, Nov. 2014.

[27] *Particle Data Group, Atomic and Nuclear Properties*. Feb. 2018. URL: http://pdg.lbl.gov/2018/AtomicNuclearProperties/.

[28] *Methodical Accelerator Design (MAD) program, CERN,* accessed on August 24th, 2017. URL: http://mad.web.cern.ch/mad.

[29] *FCC-lattice repository*. URL: fcc-optics-admin@cern.ch.

[30] *MERLIN on github:* accessed on August 24th, 2017. URL: https://github.com/MERLIN-Collaboration/MERLIN.

[31] A. Ferrari et al. *FLUKA: a multi-particle transport code*. Tech. rep. CERN Report No. 2005-010. Geneva: CERN, 2005. DOI: 10.5170/CERN-2005-010.

[32] T. T. Böhlen et al. “The FLUKA Code: Developments and Challenges for High Energy and Medical Applications”. In: *Nuclear Data Sheets* 120 (2014), pp. 211–214. DOI: 10.1016/j.nds.2014.07.049.

[33] Stefan Roesler, Ralph Engel, and Johannes Ranft. “The Monte Carlo event generator DPMJET-III”. In: *Advanced Monte Carlo for radiation physics, particle transport simulation and applications. Proceedings, Conference, MC2000, Lisbon, Portugal, October 23-26, 2000*. 2000, pp. 1033–1038. DOI: 10.1007/978-3-642-18211-2_166. arXiv: hep-ph/0012252 [hep-ph]. URL: http://www-public.slac.stanford.edu/sciDoc/docMeta.aspx?slacPubNumber=SLAC-PUB-8740.

[34] *FLUKA Homepage*. accessed on October 20th, 2018. URL: http://www.fluka.org.

[35] Theis C. et al. “Interactive three dimensional visualization and creation of geometries for Monte Carlo calculations”. In: *Nuclear Instruments and Methods in Physics Research A* 562.2 (2006), pp. 827–829. DOI: 10.1016/j.nima.2006.02.125.

[36] Roderik Bruce, Aurelien Marsili, and Stefano Redaelli. “Cleaning Performance with 11T Dipoles and Local Dispersion Suppressor Collimation at the LHC”. In: *Proceedings, 5th International Particle Accelerator Conference (IPAC 2014): Dresden, Germany, June 15-20, 2014*. 2014, MOPRO042. DOI: 10.18429/JACoW-IPAC2014-MOPRO042. URL: http://jacow.org/IPAC2014/papers/mopro042.pdf.

[37] Anton Lechner et al. “Power Deposition in LHC Magnets With and Without Dispersion Suppressor Collimators Downstream of the Betatron Cleaning Insertion”. In: *Proceedings, 5th International Particle Accelerator Conference (IPAC 2014): Dresden, Germany, June 15-20, 2014*. 2014, MOPRO021. DOI: 10.18429/JACoW-IPAC2014-MOPRO021. URL: http://jacow.org/IPAC2014/papers/mopro021.pdf.

[38] M. Fiascaris, R. Bruce, and S. Redaelli. “A conceptual solution for a beam halo collimation system for the Future Circular hadron-hadron Collider (FCC-hh)”. In: *Nucl. Instrum. Methods Phys. Res. A* 894 (2018), pp. 96–106. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2018.03.042. URL: http://www.sciencedirect.com/science/article/pii/S0168900218303759.

[39] B. Auchmann et al. “Testing beam-induced quench levels of LHC superconducting magnets. Testing Beam-Induced Quench Levels of LHC Superconducting Magnets in Run 1”. In: *Phys. Rev. Spec. Top. Accel. Beams* 18.arXiv:1502.05261 (Feb. 2015). 19 pages, 061002. 21 p. URL: https://cds.cern.ch/record/1993038.

[40] P. Granieri and R. van Weelderen. “Deduction of Steady-State Cable Quench Limits for Various Electrical Insulation Schemes with Application to LHC and HL-LHC Magnets”. In: *IEEE Trans. Appl. Supercond.* 24.CERN-ACC-2014-0035 (Feb. 2014), 4802806. 6 p. URL: https://cds.cern.ch/record/1662721.

[41] Ezio Todesco. *Personal correspondence, regarding 16T magnet limits*. CERN, TE-MSC-MDT. 2017. URL: Ezio.Todesco@cern.ch.

[42] Daniel Schulte. “FCC-hh machine layout and optics”. In: *Future Circular Collider Week 2016, Rome*. 2016. URL: https://indico.

cern.ch/event/438866/contributions/1085167/attachments/1255574/1854314/Rome_DS_final2.pdf.

[43] Amos Breskin and Rüdiger Voss. *The CERN Large Hadron Collider: Accelerator and Experiments*. Geneva: CERN, 2009. URL: https://cds.cern.ch/record/1244506.

[44] Davide Tommasini et al. "Status of the 16 T Dipole Development Program for a Future Hadron Collider". In: *IEEE Trans. Appl. Supercond.* 28.3 (2018), 4001305. 5 p. URL: https://cds.cern.ch/record/2302336.

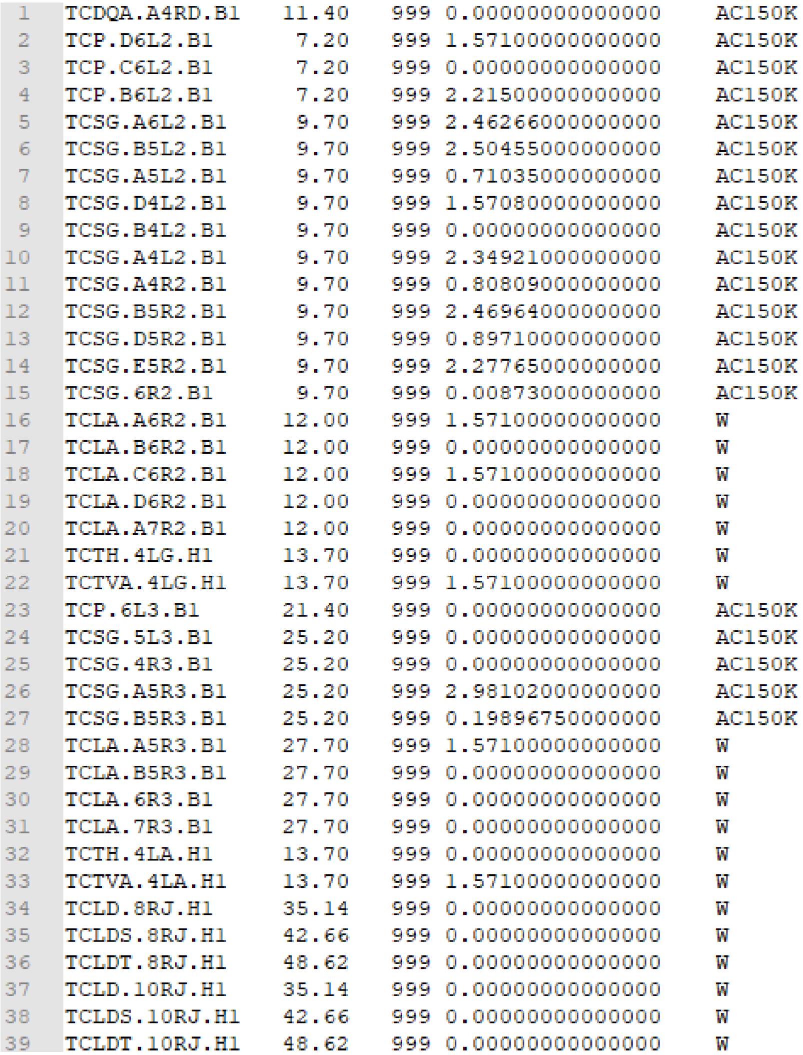
[45] Sergey Arsenyev and Daniel Schulte. "Broadband Impedance of Pumping Holes and Interconnects in the FCC-hh Beamscreen". In: *J. Phys. : Conf. Ser.* 1067.2 (2018), MOPMF030. 5 p. URL: https://cds.cern.ch/record/2647705.

[46] Roderik Bruce. *Personal correspondence, regarding differences between measured losses and simulation at the LHC.* CERN, BE-ABP-HSS. 2017. URL: Roderik.Bruce@cern.ch.

Appendix A.

MERLIN input files

A.1. Collimator input



```
TCDQA.A4RD.B1   11.40   999 0.00000000000000    AC150K
TCP.D6L2.B1      7.20   999 1.57100000000000    AC150K
TCP.C6L2.B1      7.20   999 0.00000000000000    AC150K
TCP.B6L2.B1      7.20   999 2.21500000000000    AC150K
TCSG.A6L2.B1     9.70   999 2.46266000000000    AC150K
TCSG.B5L2.B1     9.70   999 2.50455000000000    AC150K
TCSG.A5L2.B1     9.70   999 0.71035000000000    AC150K
TCSG.D4L2.B1     9.70   999 1.57080000000000    AC150K
TCSG.B4L2.B1     9.70   999 0.00000000000000    AC150K
TCSG.A4L2.B1     9.70   999 2.34921000000000    AC150K
TCSG.A4R2.B1     9.70   999 0.80809000000000    AC150K
TCSG.B5R2.B1     9.70   999 2.46964000000000    AC150K
TCSG.D5R2.B1     9.70   999 0.89710000000000    AC150K
TCSG.E5R2.B1     9.70   999 2.27765000000000    AC150K
TCSG.6R2.B1      9.70   999 0.00873000000000    AC150K
TCLA.A6R2.B1    12.00   999 1.57100000000000    W
TCLA.B6R2.B1    12.00   999 0.00000000000000    W
TCLA.C6R2.B1    12.00   999 1.57100000000000    W
TCLA.D6R2.B1    12.00   999 0.00000000000000    W
TCLA.A7R2.B1    12.00   999 0.00000000000000    W
TCTH.4LG.H1     13.70   999 0.00000000000000    W
TCTVA.4LG.H1    13.70   999 1.57100000000000    W
TCP.6L3.B1      21.40   999 0.00000000000000    AC150K
TCSG.5L3.B1     25.20   999 0.00000000000000    AC150K
TCSG.4R3.B1     25.20   999 0.00000000000000    AC150K
TCSG.A5R3.B1    25.20   999 2.98102000000000    AC150K
TCSG.B5R3.B1    25.20   999 0.19896750000000    AC150K
TCLA.A5R3.B1    27.70   999 1.57100000000000    W
TCLA.B5R3.B1    27.70   999 0.00000000000000    W
TCLA.6R3.B1     27.70   999 0.00000000000000    W
TCLA.7R3.B1     27.70   999 0.00000000000000    W
TCTH.4LA.H1     13.70   999 0.00000000000000    W
TCTVA.4LA.H1    13.70   999 1.57100000000000    W
TCLD.8RJ.H1     35.14   999 0.00000000000000    W
TCLDS.8RJ.H1    42.66   999 0.00000000000000    W
TCLDT.8RJ.H1    48.62   999 0.00000000000000    W
TCLD.10RJ.H1    35.14   999 0.00000000000000    W
TCLDS.10RJ.H1   42.66   999 0.00000000000000    W
TCLDT.10RJ.H1   48.62   999 0.00000000000000    W
```

Figure A.1.: Input of collimator settings for MERLIN. The rows are: collimator name, sigma gap x, sigma gap y, tilt angle, material (AC150K is a carbon composite and W denotes tungsten).

A.2. MERLIN simulation code

```
//
//        V14 lattice sixtrack comparison
//
#include <iostream>
#include <fstream>
#include <sstream>
#include <ctime>
#include <unistd.h>
#include <sys/stat.h>

//MPI includes
#ifdef ENABLE_MPI
#include <mpi.h>
#endif

#include "BeamDynamics/ParticleTracking/ParticleBunchConstructor.h"
#include "BeamDynamics/ParticleTracking/ParticleTracker.h"
#include "BeamDynamics/ParticleTracking/ParticleBunchTypes.h"

#include "MADInterface/MADInterface.h"

#include "Random/RandomNG.h"

#include "NumericalUtils/PhysicalUnits.h"
#include "NumericalUtils/PhysicalConstants.h"

#include "BeamDynamics/ParticleTracking/SynchRadParticleProcess.h"

#include "Collimators/CollimateParticleProcess.h"
#include "Collimators/CollimatorDatabase.h"
#include "Collimators/MaterialDatabase.h"
#include "Collimators/ApertureConfiguration.h"
```

```
#include "AcceleratorModel/Components.h"
#include "AcceleratorModel/AcceleratorErrors.h"
#include "AcceleratorModel/Apertures/CollimatorAperture.h"
#include "AcceleratorModel/Aperture.h"
#include "AcceleratorModel/Apertures/RectEllipseAperture.h"
#include "AcceleratorModel/ControlElements/Klystron.h"

#include "RingDynamics/BetatronTunes.h"
#include "RingDynamics/Dispersion.h"
#include "RingDynamics/LatticeFunctions.h"
#include "RingDynamics/Dispersion.h"
#include "RingDynamics/EquilibriumDistribution.h"

#include "AcceleratorModel/AcceleratorModel.h"

#include "BeamDynamics/ParticleTracking/BunchFilter.h"
#include "Collimators/Output/LossMapCollimationOutput.h"
#include "Collimators/CollimateProtonProcess.h"
#include "Collimators/ScatteringModelsMerlin.h"

#include "DumpOutput.h"

using namespace PhysicalUnits;
using namespace PhysicalConstants;
using namespace ParticleTracking;
using namespace Collimation;

int main(int argc, char* argv[])
{
        /*
         #ifdef ENABLE_MPI
         //Start up the MPI communicator
         MPI::Init(argc, argv);

         //Get this processes rank
         int MPI_RANK = MPI::COMM_WORLD.Get_rank();
```

```
 int MPI_SIZE = MPI::COMM_WORLD.Get_size();
 #endif
 */

/*************************************************
 **
 **
 **         GENERAL SETTINGS
 **
 **
 *************************************************/
//Be nice: do we re-nice the process to a lower
 ↪  priority?
bool be_nice = false;
int pri = 18;

string twiss_path = "./";

//Enable RF
bool enable_RF = true;
//Set the phase
double RF_phase = pi / 2;
//Set the voltage in MV
double RF_voltage = 2.0;

//Enable random errors
bool enable_errors = false;
bool SaveInputBunch = true;

//Beam energy (GeV) 7000,3500,450 etc
double beam_energy = 50000.0;

//Number of particles
int nParticles = 100;

//Number of turns around the ring
int nturns = 10;
```

```
//Save progress every save_turn_count turns?
bool save_progress = false;
size_t save_turn_count = 10;

//Load a saved bunch?
bool load_bunch = false;
size_t turn = 1;
size_t initial_turn = 1;

//Number of "real" particles in each bunch.
double beamcharge = 1e11;//6e10;

//Normalized emittance of the beam
double normalized_emittance = 2.2e-6;

//Initial seed, node seeds will be this + MPI rank.
int seed = 6;

//Output various statistics on the bunch at the end
 ↪  of each revolution
bool output_statistics = false;

//Print out lots of information, can get messy with
 ↪  lots of nodes.
bool be_verbose = false;

double beam_momentum = sqrt(pow(beam_energy, 2) -
 ↪  pow(ProtonMassMeV * MeV, 2));
double beam_kinetic_energy = beam_energy -
 ↪  ProtonMassMeV * MeV;
double gamma_rel = beam_energy /
 ↪  PhysicalConstants::ProtonMassMeV /
 ↪  PhysicalUnits::MeV;
//double beta = sqrt(1.0-(1.0/pow(gamma_rel,2)));
double beta = LorentzBeta(gamma_rel);
```

```
        double emittance = normalized_emittance /
         ↪  (gamma_rel * beta);

        int precision = cout.precision(16);
#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        {
                std::cout << std::endl;
                std::cout << "SUMMARY OF INPUT SETTINGS" <<
                 ↪  std::endl;
                std::cout << "Proton total energy: " <<
                 ↪  beam_energy << " GeV/c^2" << std::endl;
                std::cout << "Proton kinetic energy: " <<
                 ↪  beam_kinetic_energy << " GeV/c^2" <<
                 ↪  std::endl;
                std::cout << "Proton momentum: " <<
                 ↪  beam_momentum << " GeV/c" << std::endl;
                std::cout.precision(precision);
                std::cout << "Bunch real particle count: "
                 ↪  << beamcharge << std::endl;
                std::cout.precision(16);
                std::cout << "Normalized emittance: " <<
                 ↪  normalized_emittance << " m rad" <<
                 ↪  std::endl;
                std::cout << "Gamma factor: " << gamma_rel
                 ↪  << std::endl;
                std::cout << "Relativistic beta: " << beta
                 ↪  << std::endl;
                std::cout << "Geometric emittance: " <<
                 ↪  emittance << std::endl;
                std::cout << std::endl;
        }

        /****************************************************************/

        //Lower priority if requried
        if (be_nice)
        {
                int error = nice(pri);
                if (error == -1)
                {
#ifdef ENABLE_MPI
                        cout << "Node: " << MPI_RANK << "
                          ↪ Could not drop priority" <<
                          ↪ endl;
#else
                        cout << "Could not drop priority"
                          ↪ << endl;
#endif
                }

                if (be_verbose && error != -1)
                {
                        cout << "Re-niced to " << pri <<
                          ↪ endl;
                }
        }

        //All threads do the same thing, except for initial
          ↪ bunch creation, and final bunch collection
        int opt = 0;
        while ((opt = getopt(argc, argv, "s:d:p:t:")) !=
          ↪ -1)
        {
                switch (opt)
                {
                case 's':
                        seed = atoi(optarg);
                        break;
                case 'd':
                        twiss_path = optarg;
                        break;
                case 'p':
                        nParticles = atoi(optarg);
                        break;
                case 't':
                        nturns = atoi(optarg);
                        break;
                default: /* '?' */
                        exit(EXIT_FAILURE);
                }
        }

#ifdef ENABLE_MPI
        if (MPI_RANK == 0 && be_verbose)
        {
                cout << "Random Seed: " << seed << endl;
        }

        seed += MPI_RANK;
#else
        cout << "Random Seed: " << seed << endl;
#endif

        //Initialise Random number generator
        RandomNG::init(seed);

        /*************************************************
         **
         **
         **        ACCELERATOR MODEL LOADING
         **
         **
         *************************************************/

        MADInterface* myMADinterface;

#ifdef ENABLE_MPI
        if (MPI_RANK == 0 && be_verbose)
        std::cout << "Loading optics" << std::endl;
```

```
#endif

        //Load accelerator optics file.
        myMADinterface = new MADInterface(twiss_path +
         ↪  "FCC_ring.b1.V8_1.tfs", beam_energy);

        //Set the elements to be treated as drift
        myMADinterface->TreatTypeAsDrift("RFCAVITY");
        myMADinterface->TreatTypeAsDrift("HMONITOR");
        myMADinterface->TreatTypeAsDrift("VMONITOR");

        //Set MADInterface log file
        std::ofstream MADLog("output/MADlog.txt");
        myMADinterface->SetLogFile(MADLog);

        //Enable Logging
        myMADinterface->SetLoggingOn();
        myMADinterface->ConstructApertures(false);

        //Build accelerator FCChh
        AcceleratorModel* FCChh =
         ↪  myMADinterface->ConstructModel();

#ifdef ENABLE_MPI
        if(be_verbose)
        {
                cout << "Rank:" << MPI_RANK << " Built
                 ↪  MADInterface" << endl;
        }
#endif

        //Output the accelerator FCChh component statistics
#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        {
                std::ofstream
                 ↪  myoutfile("output/MAD_FCChh.txt");
                FCChh->ReportModelStatistics(myoutfile);
        }

        std::vector<RFStructure*> cavities;
        FCChh->ExtractTypedElements(cavities, "ACS*");
        Klystron* KLY_ACS_B1 = new Klystron("ACS_B1",
         ↪  cavities);
        if (enable_RF)
        {
                KLY_ACS_B1->SetVoltage(0);
//                        KLY_ACS_B1-
 ↪  >SetVoltage(RF_voltage);
                KLY_ACS_B1->SetPhase(RF_phase);
        }

        int prec = std::cout.precision(16);
        for (size_t len = 0; len < cavities.size(); len++)
        {
                std::cout <<
                 ↪  cavities[len]->GetQualifiedName() <<
                 ↪  "\t" << cavities[len]->GetLength() <<
                 ↪  "\t" << cavities[len]->GetFrequency()
                 ↪  << std::endl;
        }
        std::cout.precision(prec);

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        std::cout << "Loading optics complete" <<
         ↪  std::endl;

        /**************************************************
         **
         **
         **         BEAM PIPE LOADING
         **
         **
         ************************************************/
        ApertureConfiguration* apc = new
         ↪  ApertureConfiguration(twiss_path +
         ↪  "FCC_ring_aperture.b1.V8_1.tfs");

        OctagonalAperture* DefaultOctagon = new
         ↪  OctagonalAperture(0.0148, 0.0132, 0.378536,
         ↪  0.910831);
        apc->SetDefaultAperture(DefaultOctagon);
        apc->EnableDefaultAperture(true);

        std::ofstream* ApertureConfigurationLog;
#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        {
                ApertureConfigurationLog = new
                 ↪  std::ofstream("output/ApertureConfiguration.log");
                apc->SetLogFile(*ApertureConfigurationLog);
                apc->EnableLogging(true);
        }
        apc->ConfigureElementApertures(FCChh);

        /************************************************
         **
         **
         **         CLOSED ORBIT AND
         ** LATTICE FUNCTIONS CALCULATION
         **
         ************************************************/

        /*
         (1,0,0); // closed orbit: x
         (2,0,0); // closed orbit: px
         (3,0,0); // closed orbit: y
         (4,0,0); // closed orbit: py
         (5,0,0); // closed orbit: ct
         (6,0,0); // closed orbit: dp
         (1,1,1); // beta_x
         (1,2,1); // -alfa_x
         (3,3,2); // beta_y
         (3,4,2); // -alfa_y

         We add functions to give us the dispersion.
         Dx  = beta(1,6,3)/beta(6,6,3)
         Dpx = beta(2,6,3)/beta(6,6,3)
         etc.
         */
#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        std::cout << "Pre twiss calculation" << std::endl;

        //The lattice functions need to be calculated for
         ↪  protons that radiate.
        //i.e the closed orbit needs to be calculated with
         ↪  radiating protons and this must be fed to the
         ↪  LatticeFunctionTable class

        LatticeFunctionTable* twiss = new
         ↪  LatticeFunctionTable(FCChh, beam_energy);
        /* The following are needed for a dispersion
         ↪  calculation */

        twiss->AddFunction(1, 6, 3);  //Dx
        twiss->AddFunction(2, 6, 3);  //Dxp
        twiss->AddFunction(3, 6, 3);  //Dy
        twiss->AddFunction(4, 6, 3);  //Dyp
        twiss->AddFunction(6, 6, 3);  //dn
        twiss->AddFunction(0, 0, 1);        //mu x
        twiss->AddFunction(0, 0, 2);        //mu y
```

```
//        twiss->SetDelta(1e-15);
//        twiss->MakeTMSymplectic(true);

        double bscale1 = 0;
        while (true)
        {
                std::cout << "Trying bscale: " << bscale1
                 ↪  << "\t";
                std::cout.flush();
                twiss->ScaleBendPathLength(bscale1);
                std::cout << "Calculating" << std::endl;
                try
                {
                        twiss->Calculate();
                } catch (std::exception &e)
//                catch(...)
                {
                        std::cout << "exception: " <<
                         ↪  e.what() << std::endl;
                }

                std::cout << "Done calculating" << endl;
                if (!std::isnan(twiss->Value(1, 1, 1, 0)))
                {
                        cout << "Success!" << std::endl;
                        break;
                }
                if (bscale1 == 0)
                {
                        bscale1 = 5e-34;
                }
                bscale1 *= 2;
                std::cout << "Fail :(" << std::endl;
        }

#ifdef ENABLE_MPI
```

```
        if(MPI_RANK == 0)
#endif
        std::cout << "Post twiss" << std::endl;

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        {
                ofstream latticeFunction-
                 ↪  Log("output/LatticeFunctions.out");
                latticeFunctionLog.precision(16);
                twiss->PrintTable(latticeFunctionLog);
        }

        if (enable_RF)
        {
                KLY_ACS_B1->SetVoltage(RF_voltage);
                KLY_ACS_B1->SetPhase(RF_phase);
        }

        /*************************************************
         **
         **
         ** Collimation
         ** aperture configuration
         **
         *************************************************/

        MaterialDatabase* Materials = new
         ↪  MaterialDatabase();
        CollimatorDatabase* CollimatorDB = new
         ↪  CollimatorDatabase(twiss_path +
         ↪  "collimators_7.2.sigma_top", Materials, true);

        std::string StartElement = "TCP.C6L2.B1";  //
         ↪  TCP.C6: HORIZONTAL COLLIMATION (x)  //TCP.D6:
         ↪  VERTICAL COLLIMATION (y)
        typedef size_t Index;
        std::vector<Index> Ind;
        size_t nfound = FCChh->GetIndecies("Collimator." +
         ↪  StartElement, Ind);
        std::cout << "Found " << nfound << " pattern
         ↪  matches" << std::endl;

        if (Ind.size() != 1)
        {
                std::cout << "Non-unique index - exiting"
                 ↪  << std::endl;
                exit(EXIT_FAILURE);
        }
        size_t StartElementNumber = Ind[0];
        std::cout << StartElementNumber << std::endl;

        std::ofstream* CollimationConfigurationLog;
#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
        {
#endif
        CollimationConfigurationLog = new
         ↪  std::ofstream("output/CollimationConfig.log");
        CollimatorDB-
         ↪  >SetLogFile(*CollimationConfigurationLog);
        CollimatorDB->EnableLogging(true);
#ifdef ENABLE_MPI
}
else
{
        CollimatorDB->EnableLogging(false);
}
#endif

        CollimatorDB->MatchBeamEnvelope(false);
        CollimatorDB->SelectImpactFactor(StartElement,
         ↪  1.0e-6);
        double impact;
        //Setup the collimator jaws to appropriate sizes
        try
        {
                impact = CollimatorDB-
                 ↪  >ConfigureCollimators(FCChh, emittance,
                 ↪  emittance, twiss);
        } catch (exception& e)
        {
                std::cout << "Exception caught: " <<
                 ↪  e.what() << std::endl;
                exit(EXIT_FAILURE);
        }

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        std::cout << "Impact factor number of sigmas: " <<
         ↪  impact << endl;

        /*************************************************
         **
         **
         **          BEAM SETTINGS
         **
         **
         *************************************************/

        //Create a beam
        BeamData mybeam;

        //Default values are 0.0
        //The charge of the particles in the beam.
        //  <0 for electrons, >0 for positrons/protons.
        mybeam.charge = beamcharge / nParticles;
```

```
//Beam energy (momentum).
mybeam.p0 = beam_energy;

//TWISS beam parameters
/*
 (1,0,0); // closed orbit: x
 (2,0,0); // closed orbit: px
 (3,0,0); // closed orbit: y
 (4,0,0); // closed orbit: py
 (5,0,0); // closed orbit: ct
 (6,0,0); // closed orbit: dp
 (1,1,1); // beta_x
 (1,2,1); // -alfa_x
 (3,3,2); // beta_y
 (3,4,2); // -alfa_y
 */

mybeam.beta_x = twiss->Value(1, 1, 1,
 ↪  StartElementNumber) * meter;
mybeam.beta_y = twiss->Value(3, 3, 2,
 ↪  StartElementNumber) * meter;
mybeam.alpha_x = -twiss->Value(1, 2, 1,
 ↪  StartElementNumber);
mybeam.alpha_y = -twiss->Value(3, 4, 2,
 ↪  StartElementNumber);

//Dispersion
mybeam.Dx = twiss->Value(1, 6, 3,
 ↪  StartElementNumber) / twiss->Value(6, 6, 3,
 ↪  StartElementNumber);
mybeam.Dy = twiss->Value(2, 6, 3,
 ↪  StartElementNumber) / twiss->Value(6, 6, 3,
 ↪  StartElementNumber);
mybeam.Dxp = twiss->Value(3, 6, 3,
 ↪  StartElementNumber) / twiss->Value(6, 6, 3,
 ↪  StartElementNumber);
```

```
        mybeam.Dyp = twiss->Value(4, 6, 3,
         ↪  StartElementNumber) / twiss->Value(6, 6, 3,
         ↪  StartElementNumber);

//        impact = 10;
        mybeam.emit_x = impact * impact * emittance *
         ↪  meter;
        mybeam.emit_y = impact * impact * emittance *
         ↪  meter;

        //Beam length.
        //mybeam.sig_z = 75.5*millimeter;
//        mybeam.sig_z = SpeedOfLight * 1.0 * nanosecond /
 ↪  2.0; //0.299792458 meters
        mybeam.sig_z = 0;

        //Beam centroid
        mybeam.x0 = twiss->Value(1, 0, 0,
         ↪  StartElementNumber);
        mybeam.xp0 = twiss->Value(2, 0, 0,
         ↪  StartElementNumber);
        mybeam.y0 = twiss->Value(3, 0, 0,
         ↪  StartElementNumber);
        mybeam.yp0 = twiss->Value(4, 0, 0,
         ↪  StartElementNumber);
        mybeam.ct0 = twiss->Value(5, 0, 0,
         ↪  StartElementNumber);

        //Relative energy spread of beam.
//        mybeam.sig_dp = 4.0e-5;
        mybeam.sig_dp = 0;
        //mybeam.sig_dp = twiss->Value(6,0,0,0);

        //X-Y coupling
        mybeam.c_xy = 0.0;
        mybeam.c_xyp = 0.0;
        mybeam.c_xpy = 0.0;
        mybeam.c_xpyp = 0.0;

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        {
                //Check if beam parameters are ok.
                //cout << "\nusing:\nbeta x:\t" <<
                 ↪  twiss->Value(1,1,1,StartElementNumber)
                 ↪  << "\nbeta y:\t" <<
                 ↪  twiss->Value(3,3,2,StartElementNumber)
                 ↪  \
                << "\nalpha x: " <<
 ↪  -twiss->Value(1,2,1,StartElementNumber) << "\nalpha y:
 ↪  " << -twiss->Value(3,4,2,StartElementNumber);
                std::cout << "\nusing:\nbeta x:\t" <<
                 ↪  mybeam.beta_x << "\nbeta y:\t" <<
                 ↪  mybeam.beta_y << "\nalpha x: " <<
                 ↪  mybeam.alpha_x << "\nalpha y: "
                                << mybeam.alpha_y;
                cout << "\nDx:\t" << mybeam.Dx << "\nDy:\t"
                 ↪  << mybeam.Dy << "\nDxp:\t" <<
                 ↪  mybeam.Dxp << "\nDyp:\t" << mybeam.Dyp
                 ↪  << endl << std::endl;
                std::cout << std::endl;
                std::cout << "Initial reference particle:"
                 ↪  << std::endl;
                std::cout << "x: " << mybeam.x0 <<
                 ↪  std::endl;
                std::cout << "x': " << mybeam.xp0 <<
                 ↪  std::endl;
                std::cout << "y: " << mybeam.y0 <<
                 ↪  std::endl;
                std::cout << "y': " << mybeam.yp0 <<
                 ↪  std::endl;
                std::cout << "ct: " << mybeam.ct0 <<
                 ↪  std::endl;
                std::cout << "dp: " << mybeam.sig_dp <<
                 ↪  std::endl;
                //std::cout << "dp: " << mybeam.dp0 <<
                 ↪  std::endl;

                if (!mybeam.ok())
                {
                        std::cerr << "Bad beam parameters:
                         ↪  Check emittance and beta." <<
                         ↪  std::endl;
                        std::cout << "Beta x\t" <<
                         ↪  mybeam.beta_x << std::endl;
                        std::cout << "Beta y\t" <<
                         ↪  mybeam.beta_y << std::endl;
                        std::cout << "Alpha x\t" <<
                         ↪  mybeam.alpha_x << std::endl;
                        std::cout << "Alpha y\t" <<
                         ↪  mybeam.alpha_y << std::endl;
                        std::cout << "Gamma x\t" <<
                         ↪  mybeam.gamma_x() << std::endl;
                        std::cout << "Gamma y\t" <<
                         ↪  mybeam.gamma_y() << std::endl;
                        std::cout << "Emittance x\t" <<
                         ↪  mybeam.emit_x << std::endl;
                        std::cout << "Emittance y\t" <<
                         ↪  mybeam.emit_y << std::endl;

#ifdef ENABLE_MPI
                        MPI::COMM_WORLD.Abort(1);
#endif

                        exit(EXIT_FAILURE);
                }

                std::cout << "Beam parameters OK." <<
                 ↪  std::endl;
        }
        /**************************************************
         **
         **
         **        BUNCH SETTINGS
         **
         **
         **************************************************/

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        std::cout << "Bunch generation" << std::endl;
        ProtonBunch* myBunch;

#ifdef ENABLE_MPI
        int node_particles = nParticles/MPI_SIZE;
        if((MPI_RANK == 0) && (nParticles % MPI_SIZE != 0))
        {
                node_particles += (nParticles -
                 ↪  (node_particles * MPI_SIZE));
        }
        std::cout << "MPI_RANK: " << MPI_RANK << " making "
         ↪  << node_particles << " particles" << endl;
#else
        int node_particles = nParticles;
#endif

        //Somewhere in here there needs to be the filter.
        vector<Collimator*> TCP;
        int Tsize = FCChh->ExtractTypedElements(TCP,
         ↪  StartElement);

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        std::cout << "Found TCP?: " << Tsize << std::endl;
        /*
         Aperture* cap = NULL;

         cap = (TCP[0])->GetAperture();
         if(!pot || Tsize == 0)
         {
         std::cerr << "Could not get the TCP aperture" <<
↪   std::endl;
         abort();
         }
         */
        CollimatorAperture* CollimatorJaw =
         ↪  dynamic_cast<CollimatorAperture*>((TCP[0])-
         ↪  >GetAperture());
        if (!CollimatorJaw)
        {
                std::cerr << "Could not cast TCP collimator
                 ↪  jaw aperture" << std::endl;
                abort();
        }

        double h_orbit = twiss->Value(1, 0, 0,
         ↪  StartElementNumber);
        double JawPosition = CollimatorJaw->GetFullWidth()
         ↪  / 2.0;

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        {
                std::cout << "Jaw Width: " <<
                 ↪  CollimatorJaw->GetFullWidth() / 2.0 <<
                 ↪  std::endl;
                std::cout << "Jaw Height: " <<
                 ↪  CollimatorJaw->GetFullHeight() / 2.0 <<
                 ↪  std::endl;
                std::cout << "Orbit offset: " << h_orbit <<
                 ↪  std::endl;
        }


        HorizontalHaloParticleBunchFilter *hFilter;

        hFilter = new HorizontalHaloParticleBunchFilter();
        hFilter->SetHorizontalLimit(JawPosition);
        hFilter->SetHorizontalOrbit(h_orbit);

        ParticleBunchConstructor* constructor = new
         ↪  ParticleBunchConstructor(mybeam,
         ↪  node_particles, horizontalHaloDistribution1);

        constructor->SetFilter(hFilter);

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        std::cout << "Start bunch construction" <<
         ↪  std::endl;
        myBunch = constructor-
         ↪  >ConstructParticleBunch<ProtonBunch>();
        delete constructor;

#ifdef ENABLE_MPI
        if(MPI_RANK == 0)
#endif
        std::cout << "End bunch construction" << std::endl;

        myBunch->SetMacroParticleCharge(mybeam.charge);

        SaveInputBunch = false;
        if (SaveInputBunch)
        {
                ostringstream BunchOutputFile;
```

```
#ifdef ENABLE_MPI
                BunchOutputFile << "output/Node_" <<
                 ↪  MPI_RANK << "_bunch_in.txt";
#else
                BunchOutputFile << "output/bunch_in.txt";
#endif

                std::ofstream* bunch_output = new
                 ↪  std::ofstream(BunchOutputFile.str().c_str());
                myBunch->Output(*bunch_output);
                bunch_output->close();
                delete bunch_output;
        }

        /*************************************************
         **
         **
         **         PARTICLE TRACKER
         **
         **
         *************************************************/

        std::cout << "Constructing main tracker" <<
         ↪  std::endl;
        ParticleTracker* tracker = new
         ↪  ParticleTracker(FCChh-
         ↪  >GetRing(StartElementNumber),
         ↪  myBunch);

        /*************************************************
         **
         **
         **         SYNCHROTRON RADIATION
         **
         **
         *************************************************/
        SynchRadParticleProcess* srad = new
         ↪  SynchRadParticleProcess(10, true);
        //Include radiation effects in Quadrupoles and Skew
         ↪  Quadrupoles. Bool switch.
        srad->IncludeQuadRadiation(true);

        //Set photon generation type HBSpectrumGen or
         ↪  AWSpectrumGen
        srad->SetPhotonGenerator(HBSpectrumGen);

        //Set number of steps though each component,
         ↪  default = 1
        srad->SetNumComponentSteps(1);

        srad->AdjustBunchReferenceEnergy(false);
        //tracker->AddProcess(srad);
        //ofstream* srlog = new ofstream("output/sradlog");
        //srlog->precision(16);

        /*************************************************
         **
         **
         **        COLLIMATION
         **
         **
         **************************************************/

#ifdef ENABLE_MPI
        std::stringstream MPIlossOutput;
        MPIlossOutput << "output/LossLocations_Node_" <<
         ↪  MPI_RANK;
        std::ofstream* CollimationOutput = new
         ↪  std::ofstream(MPIlossOutput.str().c_str());
#else
        std::ofstream* CollimationOutput = new
         ↪  std::ofstream("output/LossLocations");
#endif
```

```
        Collimation::ScatteringModelMerlin* sm = new
         ↪  Collimation::ScatteringModelMerlin();
        //sm->SetScatterType(4);
        CollimateProtonProcess* colp = new
         ↪  CollimateProtonProcess(1, 4,
         ↪  CollimationOutput);
        colp->ScatterAtCollimator(true);
        colp->SetOutputBinSize(0.1);
        colp->SetScatteringModel(sm);

        LossMapCollimationOutput* ColOut = new
         ↪  LossMapCollimationOutput();
        ColOut->SetWarmRegion(std::pair<double,
         ↪  double>(47985.716815190164,
         ↪  49385.716815190273));
        ColOut->SetWarmRegion(std::pair<double,
         ↪  double>(77590.591277852014,
         ↪  80335.111277851640));

        colp->SetCollimationOutput(ColOut);
//---------------------------------------------------------
 ↪  -------------------------------
        //initial_tracker->AddProcess(colp);
//---------------------------------------------------------
 ↪  -------------------------------
        tracker->AddProcess(colp);

//---------------------------------------------------------
 ↪  -------------------------------
        DumpOutput* dumpTCP = new DumpOutput();
        //dumpTCP->AddIdentifier("Collimator.TCP.C6L2.B1");
        dumpTCP->AddIdentifier("SectorBend.MB.D8RJ.H1");
        dumpTCP->AddIdentifier("SectorBend.MB.D810RJ.H1");

//---------------------------------------------------------
 ↪  -------------------------------
```

```
//        initial_tracker->SetOutput(dumpALL);
//---------------------------------------------------------
 ↪  -------------------------------

        tracker->SetOutput(dumpTCP);


        /*************************************************
         **
         **
         **        TRACKING RUN
         **
         **
         *************************************************/

//---------------------------------------------------------
 ↪  -------------------------------
        //initial_tracker->Track(myBunch);
//---------------------------------------------------------
 ↪  -------------------------------
        int f = 0;
#ifdef ENABLE_MPI
        double ff = MPI_SIZE;
#else
        double ff = 1;
#endif
        while (ff > 1)
        {
                ff /= 10.0;
                f++;
        }
        int RankPrintSize = f + 2;

        f = 0;
        ff = nturns;
        while (ff > 1)
        {
                ff /= 10.0;
                f++;
        }
        int TurnPrintSize = f + 2;

        initial_turn = 1;
        // Do the loop for nturns times
        for (turn = initial_turn; turn <= nturns; turn++)
        {
                dumpTCP->SetTurn(turn);


#ifdef ENABLE_MPI
                std::cout << "Rank: " << std::left <<
                 ↪  std::setw(RankPrintSize) << MPI_RANK <<
                 ↪  "Turn: " << std::setw(TurnPrintSize) <<
                 ↪  turn <<"Particle number: " <<
                 ↪  myBunch->size() << std::endl;
#else
                std::cout << "Turn " << turn << "\tParticle
                 ↪  number: " << myBunch->size() <<
                 ↪  std::endl;
#endif

                tracker->Track(myBunch);
                if (myBunch->size() <= 1)
                {
#ifdef ENABLE_MPI
                        std::cout << "Rank: " << std::left
                         ↪  << std::setw(RankPrintSize) <<
                         ↪  MPI_RANK << "lost all particles
                         ↪  on turn: " << turn <<
                         ↪  std::endl;
#endif
                        break;
                }
```

```
                 //Do the tracking
                 if(turn == initial_turn)
                 {
                 tracker->Run();
                 }
                 else
                 {
                 tracker->Continue();
                 }

#ifdef ENABLE_MPI
        cout << "Rank: " << MPI_RANK << "
         ↪  finished.\tParticle number: " <<
         ↪  myBunch->size() << endl;
        nice(19);
        myBunch->gather();
        if(MPI_RANK == 0)
#endif
        {
                cout << "nParticles: " << nParticles <<
                 ↪  endl;
                cout << "left: " << myBunch->size() <<
                 ↪  endl;
                cout << "absorbed: " << nParticles -
                 ↪  myBunch->size() << endl;
        }

        ostringstream LossMapOutputFile;
#ifdef ENABLE_MPI
        LossMapOutputFile << "output/Node_" << MPI_RANK <<
         ↪  "_LossMap.out";
#else
        LossMapOutputFile << "output/LossMap.out";
#endif
```

```
        std::ofstream* colloutput = new
         ↪  std::ofstream(LossMapOutputFile.str().c_str());
        ColOut->Finalise();
        ColOut->Output(colloutput);

#ifdef ENABLE_MPI
        //Deleting the bunch will also terminate MPI, hence
         ↪  this should be the last thing we do..
        myBunch->MPI_Finalize();
#endif

        //std::cout << myBunch->GetParticles()[1].x() <<
         ↪  "\t" << (1 + myBunch->GetParticles()[1].dp()) *
         ↪  50000.0 << std::endl;

        //PSmoments result;
        //myBunch->GetMoments(result);
        //double tempp = myBunch->GetReferenceMomentum();
        //pair<double,double> dpp =
         ↪  myBunch->GetMoments(ps_DP);
        //std::cout << "Mean energy: " << 50000 * dpp.first
         ↪  << std::endl;
        //std::cout << "Loss per turn: " << 50000 * 1000 *
         ↪  dpp.first/nturns  << " MeV" << std::endl;

        delete myBunch;

        return EXIT_SUCCESS;
}
//The end
```

Appendix B.

FLUKA Files

B.1. User routines

B.1.1. user source input

file parser

```
/*
 * merlinsource.cpp
 *
 *  Created on: Feb 6, 2017
 *      Author: alkraine
 */

#include "merlinsource.h"

std::vector<C6Vector> g_particles;

void initsource_(char* filename, double *u1, double *u2)
{

        using namespace std;
        using namespace stltoolbox;

        vector<Point> b12;
        string line;
        vector<string> wordList;
```

```
wordList = TokenizeString(filename, " ");
string file = "../" + wordList.at(0);

ifstream inStream;

cout << "------------------------------------------
 ↪  ---------------------------" <<
 ↪  endl;
cout << "---                     Merlin Source
 ↪  Parser                      ---" << endl;
cout << "---           (based on Sixtrack Source
 ↪  Parser 1.1)              ---" << endl;
cout << "---             Version 1.0, Feb. 2017,
 ↪  A. Krainer               ---" << endl;
cout << "------------------------------------------
 ↪  ---------------------------" <<
 ↪  endl;

inStream.open(file);

double x = 0, xp = 0, y = 0, yp = 0, z = 0, p = 0;
clog << "m1 is: " << *u1 << endl;
clog << "m2 is: " << *u2 << endl;
if (!inStream)
{
        cerr << "ERROR: Error while opening source
         ↪  input file!" << endl;
        cerr << "ERROR: Filename: " << file <<
         ↪  endl;
        exit(1);
}

cout << "LOG: reading source from input file: " <<
 ↪  file << endl;

while (getline(inStream, line))
{
        if (!line.empty() &&
         ↪  line.at(line.find_first_not_of(" ")) !=
         ↪  '#')
        {
                wordList = TokenizeString(line, "
                 ↪  ");
                y = FromString<double>(wordList[2])
                 ↪  * 100;
                yp = From-
                 ↪  String<double>(wordList[3]);
                x = FromString<double>(wordList[4])
                 ↪  * 100;
                xp = From-
                 ↪  String<double>(wordList[5]);
                z = FromString<double>(wordList[6])
                 ↪  * 100 - *u1;
                p = From-
                 ↪  String<double>(wordList[1])*(1+FromString<double>(wo

                x = x + *u2 * tan(xp);
                y = y + *u2 * tan(yp);

                xp = sin(xp);
                yp = sin(yp);
                //zp: Calculated by FLUKA so xp,yp
                 ↪  and zp are normalized



                g_particles.push_back(C6Vector(x,
                 ↪  xp, y, yp, z, p));

        }
}

inStream.close();
}

void getparticle_(double* WT, double* pX, double* pY,
 ↪  double* pZ, double* dX, double* dY, double* dZ, double*
 ↪  PB)
{
        *WT = 1.0;
        int randomIndex = (GetRndNumber() *
         ↪  g_particles.size());
        C6Vector particle = g_particles.at(randomIndex);
        *pX = particle.m_X;
        *pY = particle.m_Y;
        *pZ = particle.m_Z;

        *dX = particle.m_pX;
        *dY = particle.m_pY;
        *dZ = 0;

        *PB = particle.m_P;
}
double GetRndNumber()
{
        double Dummy;

        //call FLUKA's random number generator
        return flrndm_(&Dummy);
}
```

user source routine

```
*$ CREATE SOURCE.FOR
*COPY SOURCE
*
*=== source
 ↪  ============================================================*
*
      SUBROUTINE SOURCE ( NOMORE )
      INCLUDE '(DBLPRC)'
      INCLUDE '(DIMPAR)'
      INCLUDE '(IOUNIT)'
*
*----------------------------------------------------------------
 ↪  -------------*
*
 ↪  *
*     Copyright (C) 1990-2010      by    Alfredo Ferrari &
 ↪  Paola Sala   *
*     All Rights Reserved.
 ↪  *
*
 ↪  *
*
 ↪  *
*     New source for FLUKA9x-FLUKA20xy:
 ↪  *
*
 ↪  *
*     Created on 07 January 1990   by    Alfredo Ferrari &
 ↪  Paola Sala   *
*                                                    Infn -
 ↪  Milan        *
*
 ↪  *
*     Last change on  17-Oct-10    by    Alfredo Ferrari
 ↪  *
*
 ↪  *
*  This is just an example of a possible user written
 ↪  source routine.  *
*  note that the beam card still has some meaning - in the
 ↪  scoring the *
*  maximum momentum used in deciding the binning is taken
 ↪  from the      *
```

```
*  beam momentum.  Other beam card parameters are obsolete.
 ↪   *
*
 ↪   *
*       Output variables:
 ↪   *
*
 ↪   *
*              Nomore = if > 0 the run will be terminated
 ↪   *
*
 ↪   *
*-----------------------------------------------------------
 ↪   ------------*
*
      INCLUDE '(BEAMCM)'
      INCLUDE '(FHEAVY)'
      INCLUDE '(FLKSTK)'
      INCLUDE '(IOIOCM)'
      INCLUDE '(LTCLCM)'
      INCLUDE '(PAPROP)'
      INCLUDE '(SOURCM)'
      INCLUDE '(SUMCOU)'
*
      LOGICAL LFIRST
*
      SAVE LFIRST
      DATA LFIRST / .TRUE. /
*=======================================================================*
*
 ↪   *
*                 BASIC VERSION
 ↪   *
*
 ↪   *
*=======================================================================*
      NOMORE = 0
```

```
*  +-------------------------------------------------------
 ↪  ------------*
*  |  First call initializations:
      IF ( LFIRST ) THEN
*  |  *** The following 3 cards are mandatory ***
         TKESUM = ZERZER
         LFIRST = .FALSE.
         LUSSRC = .TRUE.
*  |  *** User initialization ***
         CALL initsource(SDUSOU, WHASOU(1), WHASOU(2))
      END IF
*  |
*  +-------------------------------------------------------
 ↪  ------------*
*  Push one source particle to the stack. Note that you
 ↪  could as well
*  push many but this way we reserve a maximum amount of
 ↪  space in the
*  stack for the secondaries to be generated
*  Npflka is the stack counter: of course any time source
 ↪  is called it
*  must be =0
      CALL getparticle(WT, PX, PY, PZ, DX, DY, DZ, PB)
      NPFLKA = NPFLKA + 1
*  Wt is the weight of the particle
      WTFLK  (NPFLKA) = ONEONE
      WEIPRI = WEIPRI + WTFLK (NPFLKA)
*  Particle type (1=proton.....). Ijbeam is the type set by
 ↪  the BEAM
*  card
*  +-------------------------------------------------------
 ↪  ------------*
*  |  (Radioactive) isotope:
      IF ( IJBEAM .EQ. -2 .AND. LRDBEA ) THEN
         IARES  = IPROA
         IZRES  = IPROZ
         IISRES = IPROM
         CALL STISBM ( IARES, IZRES, IISRES )
         IJHION = IPROZ  * 1000 + IPROA
         IJHION = IJHION * 100 + KXHEAV
         IONID  = IJHION
         CALL DCDION ( IONID )
         CALL SETION ( IONID )
*  |
*  +------------------------------------------------------
 ↪  ------------*
*  |  Heavy ion:
      ELSE IF ( IJBEAM .EQ. -2 ) THEN
         IJHION = IPROZ  * 1000 + IPROA
         IJHION = IJHION * 100 + KXHEAV
         IONID  = IJHION
         CALL DCDION ( IONID )
         CALL SETION ( IONID )
         ILOFLK (NPFLKA) = IJHION
*  |  Flag this is prompt radiation
         LRADDC (NPFLKA) = .FALSE.
*  |  Group number for "low" energy neutrons, set to 0
 ↪  anyway
         IGROUP (NPFLKA) = 0
*  |
*  +------------------------------------------------------
 ↪  ------------*
*  |  Normal hadron:
      ELSE
         IONID = IJBEAM
         ILOFLK (NPFLKA) = IJBEAM
*  |  Flag this is prompt radiation
         LRADDC (NPFLKA) = .FALSE.
*  |  Group number for "low" energy neutrons, set to 0
 ↪  anyway
         IGROUP (NPFLKA) = 0
      END IF
*  |
*  +--------------------------------------------------------
 ↪  ------------*
*  From this point .....
*  Particle generation (1 for primaries)
      LOFLK  (NPFLKA) = 1
*  User dependent flag:
      LOUSE  (NPFLKA) = 0
*  No channeling:
      LCHFLK (NPFLKA) = .FALSE.
      DCHFLK (NPFLKA) = ZERZER
*  User dependent spare variables:
      DO 100 ISPR = 1, MKBMX1
         SPAREK (ISPR,NPFLKA) = ZERZER
 100  CONTINUE
*  User dependent spare flags:
      DO 200 ISPR = 1, MKBMX2
         ISPARK (ISPR,NPFLKA) = 0
 200  CONTINUE
*  Save the track number of the stack particle:
      ISPARK (MKBMX2,NPFLKA) = NPFLKA
      NPARMA = NPARMA + 1
      NUMPAR (NPFLKA) = NPARMA
      NEVENT (NPFLKA) = 0
      DFNEAR (NPFLKA) = +ZERZER
*  ... to this point: don't change anything
*  Particle age (s)
      AGESTK (NPFLKA) = +ZERZER
      AKNSHR (NPFLKA) = -TWOTWO
*  Kinetic energy of the particle (GeV)
      TKEFLK (NPFLKA) = SQRT ( PB**2 + AM (IONID)**2 ) - AM
 ↪  (IONID)
*  Particle momentum
      PMOFLK (NPFLKA) = PB
*     PMOFLK (NPFLKA) = SQRT ( TKEFLK (NPFLKA) * ( TKEFLK
 ↪  (NPFLKA)
*    &                       + TWOTWO * AM (IONID) ) )
*  Cosines (tx,ty,tz)
```

```
      TXFLK  (NPFLKA) = DX
      TYFLK  (NPFLKA) = DY
*     TZFLK  (NPFLKA) = DZ
      TZFLK  (NPFLKA) = SQRT ( ONEONE - TXFLK (NPFLKA)**2
     &                       - TYFLK (NPFLKA)**2 )
*  Polarization cosines:
      TXPOL  (NPFLKA) = -TWOTWO
      TYPOL  (NPFLKA) = +ZERZER
      TZPOL  (NPFLKA) = +ZERZER
*  Particle coordinates
      XFLK   (NPFLKA) = PX
      YFLK   (NPFLKA) = PY
      ZFLK   (NPFLKA) = PZ
*  Calculate the total kinetic energy of the primaries:
 ↪  don't change
      IF ( ILOFLK (NPFLKA) .EQ. -2 .OR. ILOFLK (NPFLKA)
 ↪  .GT. 100000 )
     &   THEN
         TKESUM = TKESUM + TKEFLK (NPFLKA) * WTFLK (NPFLKA)
      ELSE IF ( ILOFLK (NPFLKA) .NE. 0 ) THEN
         TKESUM = TKESUM + ( TKEFLK (NPFLKA) + AMDISC
 ↪  (ILOFLK(NPFLKA)) )
     &          * WTFLK (NPFLKA)
      ELSE
         TKESUM = TKESUM + TKEFLK (NPFLKA) * WTFLK (NPFLKA)
      END IF
      RADDLY (NPFLKA) = ZERZER
*  Here we ask for the region number of the hitting point.
*     NREG (NPFLKA) = ...
*  The following line makes the starting region search much
 ↪  more
*  robust if particles are starting very close to a
 ↪  boundary:
      CALL GEOCRS ( TXFLK (NPFLKA), TYFLK (NPFLKA), TZFLK
 ↪  (NPFLKA) )
      CALL GEOREG ( XFLK  (NPFLKA), YFLK  (NPFLKA), ZFLK
 ↪  (NPFLKA),
     &              NRGFLK(NPFLKA), IDISC )
*  Do not change these cards:
      CALL GEOHSM ( NHSPNT (NPFLKA), 1, -11, MLATTC )
      NLATTC (NPFLKA) = MLATTC
      CMPATH (NPFLKA) = ZERZER
      CALL SOEVSV
      RETURN
*=== End of subroutine Source
 ↪  ========================================*
      END
```

B.1.2. user magnet field input

```
*$ CREATE MAGFLD.FOR
*COPY MAGFLD
*
*===magfld============================================================*
*
      SUBROUTINE MAGFLD ( X, Y, Z, BTX, BTY, BTZ, B, NREG,
 ↪  IDISC )

      INCLUDE '(DBLPRC)'
      INCLUDE '(DIMPAR)'
      INCLUDE '(IOUNIT)'
*
*----------------------------------------------------------
 ↪  ------------*
*
 ↪  *
*     Copyright (C) 1988-2010      by Alberto Fasso` &
 ↪  Alfredo Ferrari *
*     All Rights Reserved.
 ↪  *
*
 ↪  *
*
 ↪  *
```

```
*     Created  in     1988         by    Alberto Fasso`
 ↪  *
*
 ↪  *
*
 ↪  *
*     Last change on 06-Nov-10     by    Alfredo Ferrari
 ↪  *
*
 ↪  *
*     Input variables:
 ↪  *
*            x,y,z = current position
 ↪  *
*            nreg  = current region
 ↪  *
*     Output variables:
 ↪  *
*            btx,bty,btz = cosines of the magn. field
 ↪  vector           *
*            B = magnetic field intensity (Tesla)
 ↪  *
*            idisc = set to 1 if the particle has to be
 ↪  discarded      *
*
 ↪  *
*----------------------------------------------------------
 ↪  ------------*
*
*      INCLUDE '(CMEMFL)'
*      INCLUDE '(CSMCRY)'
*
*  +-------------------------------------------------------
 ↪  ------------*
*  |  Earth geomagnetic field:
*      IF ( LGMFLD ) THEN
*         CALL GEOFLD ( X, Y, Z, BTX, BTY, BTZ, B, NREG,
 ↪  IDISC )
*         RETURN
*      END IF
*  |
*  +----------------------------------------------------------
 ↪  ------------*
      INCLUDE '(USRVAR)'
      IDISC = 0

      IF ( NREG.GE.FIREQM .AND. NREG.LE.LAREQM) THEN
         BTX = GRADQM * Y
         BTY = GRADQM * X
         BTZ = ZERZER
         B = SQRT(BTX**2 + BTY**2 + BTZ**2)
      ELSE IF (NREG.GE.FIREOM .AND. NREG.LE.LAREOM) THEN
         BTX = GRADQM * (APERQM/SQRT(X**2 +
 ↪  Y**2)-APERQM/MAXLOM) * Y
         BTY = GRADQM * (APERQM/SQRT(X**2 +
 ↪  Y**2)-APERQM/MAXLOM) * X
         BTZ = ZERZER
         B = SQRT(BTX**2 + BTY**2 + BTZ**2)
      ELSE IF (NREG.GE.FIREDM .AND. NREG.LE.LAREDM) THEN
         BTX = XFSTDM
         BTY = YFSTDM
         BTZ = ZFSTDM
         B = SQRT(BTX**2 + BTY**2 + BTZ**2)
      END IF
      IF (B.gt.1E-12) THEN
         BTX = BTX / B
         BTY = BTY / B
         BTZ = BTZ / B
      ELSE
         BTZ = ZERZER
         BTY = ZERZER
         BTZ = ONEONE
```

```
          B = ZERZER
       END IF
       RETURN
*=== End of subroutine Magfld ==================================================*
       END
```

B.2. FLUKA input file

```
TITLE
DIS Collimation COL_QM_DIM
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
GLOBAL       10000.0       0.0       0.0       0.0       1.0        0.
*
DEFAULTS                                                              PRECISIO
*
* Inermet 180: Real material for tungsten jaw collimators
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
MATERIAL        25.0    54.938      7.21                              MANGANES
MATERIAL        24.0   51.9961      7.18                              CHROMIUM
MATERIAL                            18.0                              INERM180
MATERIAL                            8.14                              STEEL
MATERIAL        41.0  92.90637      8.57                              NIOBIUM
MATERIAL                            8.95                              NB3SN
MATERIAL                            8.95                              NB3SNCU
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
COMPOUND       -0.95  TUNGSTEN    -0.035    NICKEL    -0.015    COPPERINERM180
COMPOUND        -56.      IRON      -20.  CHROMIUM      -11.    NICKELSTEEL
COMPOUND        -12.  MANGANES       -1.    COPPER                    STEEL
COMPOUND         3.0   NIOBIUM       1.0       TIN                    NB3SN
COMPOUND        -50.    COPPER      -50.     NB3SN                    NB3SNCU
LOW-MAT         IRON      26.0      -2.0     296.0
LOW-MAT     MANGANES      25.0      55.0     296.0
LOW-MAT     CHROMIUM      42.0      -2.0     296.0
*
GEOBEGIN
    0     0                  MC-CAD
* DBORE1
  REC   DBORE1   0.0000000000000000   0.0000000000000000 2074.0080000000000000
```

```
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   2.3500000000000000   0.0000000000000000
                 2.3500000000000000   0.0000000000000000   0.0000000000000000
* DBORE2
  REC   DBORE2   0.0000000000000000   0.0000000000000000 2074.0080000000000000
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   2.2000000000000000   0.0000000000000000
                 2.2000000000000000   0.0000000000000000   0.0000000000000000
* DBORE3
  REC   DBORE3   0.0000000000000000 -25.0000000000000000 2074.0080000000000000
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   2.3500000000000000   0.0000000000000000
                 2.3500000000000000   0.0000000000000000   0.0000000000000000
* DBORE4
  REC   DBORE4   0.0000000000000000 -25.0000000000000000 2074.0080000000000000
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   2.2000000000000000   0.0000000000000000
                 2.2000000000000000   0.0000000000000000   0.0000000000000000
* DBS1_O
  REC   DBS1_O   0.0000000000000000   0.0000000000000000 2074.0080000000000000
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   1.7000000000000000   0.0000000000000000
                 1.5200000000000000   0.0000000000000000   0.0000000000000000
* DBS1_V
  REC   DBS1_V   0.0000000000000000   0.0000000000000000 2074.0080000000000000
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   1.5000000000000000   0.0000000000000000
                 1.3200000000000000   0.0000000000000000   0.0000000000000000
* DBS2_O
  REC   DBS2_O   0.0000000000000000 -25.0000000000000000 2074.0080000000000000
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   1.7000000000000000   0.0000000000000000
                 1.5200000000000000   0.0000000000000000   0.0000000000000000
* DBS2_V
  REC   DBS2_V   0.0000000000000000 -25.0000000000000000 2074.0080000000000000
                 0.0000000000000000   0.0000000000000000 1500.0000000000000000
                 0.0000000000000000   1.5000000000000000   0.0000000000000000
                 1.3200000000000000   0.0000000000000000   0.0000000000000000
* DSTIN1
  RCC   DSTIN1     0.0000     0.0000  2074.0080     0.0000     0.0000  1500.0000
              6.2000
* DSTIN2
  RCC   DSTIN2     0.0000   -25.0000  2074.0080     0.0000     0.0000  1500.0000
              6.2000
```

```
* DSTOUT1
  RCC  DSTOUT1     0.0000     0.0000  2074.0080     0.0000     0.0000  1500.0000
              7.2000
* DSTOUT2
  RCC  DSTOUT2     0.0000   -25.0000  2074.0080     0.0000     0.0000  1500.0000
              7.2000
* Dapert_1
  RCC Dapert_1     0.0000     0.0000  2074.0080     0.0000     0.0000  1500.0000
              2.5000
* Dapert_2
  RCC Dapert_2     0.0000   -25.0000  2074.0080     0.0000     0.0000  1500.0000
              2.5000
* Dpc2_3
  PLA   Dpc2_3    0.50000    0.86603    0.00000      0.000    -25.500    640.000
* Dpc2_4
  PLA   Dpc2_4    0.50000   -0.86603    0.00000      0.000    -24.500    640.000
* Dpc2_7
  PLA   Dpc2_7   -0.50000   -0.86603    0.00000      0.000    -24.500    640.000
* Dpc2_8
  PLA   Dpc2_8   -0.50000    0.86603    0.00000      0.000    -25.500    640.000
* Dpc3
  PLA     Dpc3    0.50000    0.86603    0.00000      0.000     -0.500    640.000
* Dpc4
  PLA     Dpc4    0.50000   -0.86603    0.00000      0.000      0.500    640.000
* Dpc7
  PLA     Dpc7   -0.50000   -0.86603    0.00000      0.000      0.500    640.000
* Dpc8
  PLA     Dpc8   -0.50000    0.86603    0.00000      0.000     -0.500    640.000
* Dyk
  RCC      Dyk     0.0000   -12.5000  2074.0080     0.0000     0.0000  1500.0000
             40.0000
* QBORE1
  REC   QBORE1   0.0000000000000000   0.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   2.3500000000000000   0.0000000000000000
                 2.3500000000000000   0.0000000000000000   0.0000000000000000
* QBORE2
  REC   QBORE2   0.0000000000000000   0.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   2.2000000000000000   0.0000000000000000
                 2.2000000000000000   0.0000000000000000   0.0000000000000000
* QBORE3
  REC   QBORE3   0.0000000000000000 -25.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   2.3500000000000000   0.0000000000000000
                 2.3500000000000000   0.0000000000000000   0.0000000000000000
* QBORE4
  REC   QBORE4   0.0000000000000000 -25.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   2.2000000000000000   0.0000000000000000
                 2.2000000000000000   0.0000000000000000   0.0000000000000000
* QBS1_O
  REC   QBS1_O   0.0000000000000000   0.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   1.7000000000000000   0.0000000000000000
                 1.5200000000000000   0.0000000000000000   0.0000000000000000
* QBS1_V
  REC   QBS1_V   0.0000000000000000   0.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   1.5000000000000000   0.0000000000000000
                 1.3200000000000000   0.0000000000000000   0.0000000000000000
* QBS2_O
  REC   QBS2_O   0.0000000000000000 -25.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   1.7000000000000000   0.0000000000000000
                 1.5200000000000000   0.0000000000000000   0.0000000000000000
* QBS2_V
  REC   QBS2_V   0.0000000000000000 -25.0000000000000000 707.0000000000000000
                 0.0000000000000000   0.0000000000000000 1000.0000000000000000
                 0.0000000000000000   1.5000000000000000   0.0000000000000000
                 1.3200000000000000   0.0000000000000000   0.0000000000000000
* QSTIN1
  RCC   QSTIN1     0.0000     0.0000   707.0000     0.0000     0.0000   1000.0000
              6.2000
* QSTIN2
  RCC   QSTIN2     0.0000   -25.0000   707.0000     0.0000     0.0000   1000.0000
              6.2000
* QSTOUT1
  RCC  QSTOUT1     0.0000     0.0000   707.0000     0.0000     0.0000   1000.0000
              7.2000
* QSTOUT2
  RCC  QSTOUT2     0.0000   -25.0000   707.0000     0.0000     0.0000   1000.0000
              7.2000
* Qapert_1
  RCC Qapert_1     0.0000     0.0000   707.0000     0.0000     0.0000   1000.0000
              2.5000
* Qapert_2
  RCC Qapert_2     0.0000   -25.0000   707.0000     0.0000     0.0000   1000.0000
```

```
              2.5000
* Qpc1
  PLA     Qpc1    0.50000   -0.86603    0.00000      0.000      0.500      0.000
* Qpc2
  PLA     Qpc2   -0.86603    0.50000   -0.00000      0.500      0.000      0.000
* Qpc2_1
  PLA   Qpc2_1    0.50000   -0.86603    0.00000      0.000    -24.500      0.000
* Qpc2_2
  PLA   Qpc2_2   -0.86603    0.50000   -0.00000      0.500    -25.000      0.000
* Qpc2_3
  PLA   Qpc2_3    0.86603    0.50000    0.00000     -0.500    -25.000      0.000
* Qpc2_4
  PLA   Qpc2_4   -0.50000   -0.86603    0.00000      0.000    -24.500      0.000
* Qpc2_5
  PLA   Qpc2_5    0.86603   -0.50000   -0.00000     -0.500    -25.000      0.000
* Qpc2_6
  PLA   Qpc2_6   -0.50000    0.86603    0.00000      0.000    -25.500      0.000
* Qpc2_7
  PLA   Qpc2_7   -0.86603   -0.50000    0.00000      0.500    -25.000      0.000
* Qpc2_8
  PLA   Qpc2_8    0.50000    0.86603    0.00000      0.000    -25.500      0.000
* Qpc3
  PLA     Qpc3    0.86603    0.50000    0.00000     -0.500      0.000      0.000
* Qpc4
  PLA     Qpc4   -0.50000   -0.86603    0.00000      0.000      0.500      0.000
* Qpc5
  PLA     Qpc5    0.86603   -0.50000   -0.00000     -0.500      0.000      0.000
* Qpc6
  PLA     Qpc6   -0.50000    0.86603    0.00000      0.000     -0.500      0.000
* Qpc7
  PLA     Qpc7   -0.86603   -0.50000    0.00000      0.500      0.000      0.000
* Qpc8
  PLA     Qpc8    0.50000    0.86603    0.00000      0.000     -0.500      0.000
* Qyk
  RCC      Qyk     0.0000   -12.5000   707.0000     0.0000     0.0000   1000.0000
             40.0000
* Dump
*  RPP     Dump     -90.00      90.00     -90.00      90.00    3600.00    3700.00
* Vac
  RPP      Vac  -100.0000   100.0000  -100.0000   100.0000  -150.0000  3750.0000
* blk
  RPP      blk  -105.0000   105.0000  -105.0000   105.0000  -155.0000  3755.0000
*-------------------------------------------------------------------------------------
* colPL
```

```
  RPP     colPL  -15.0000   15.0000  -15.130041   -0.130041  -10.0000  110.0000
* colPR
  RPP     colPR  -15.0000   15.0000    0.130041   15.130041  -10.0000  110.0000
* pa7L
  PLA     pa7L    0.00000    0.89509   -0.44589      0.000     -0.130041      0.000
* pa7R
  PLA     pa7R    0.00000    0.89509   -0.44589      0.000      0.130041    100.000
* pa8L
  PLA     pa8L    0.00000    0.89509    0.44589      0.000     -0.130041    100.000
* pa8R
  PLA     pa8R    0.00000    0.89509    0.44589      0.000      0.130041      0.000
*--------------------------------------------------------------------------------------
* colSL
  RPP     colSL  -15.0000   15.0000  -15.140000   -0.140000  365.0000  535.0000
* colSR
  RPP     colSR  -15.0000   15.0000    0.140000   15.140000  365.0000  535.0000
* pb7L
  PLA     pb7L    0.00000    0.89509   -0.44589      0.000     -0.140000    375.000
* pb7R
  PLA     pb7R    0.00000    0.89509   -0.44589      0.000      0.140000    525.000
* pb8L
  PLA     pb8L    0.00000    0.89509    0.44589      0.000     -0.140000    525.000
* pb8R
  PLA     pb8R    0.00000    0.89509    0.44589      0.000      0.140000    375.000
*-----------------------------------------------------------------------------------
*--------------------------------------------------------------------------------------
* colTL
  RPP     colTL  -15.0000   15.0000  -15.145000   -0.145000  1807.0000  1977.0000
* colTR
  RPP     colTR  -15.0000   15.0000    0.145000   15.145000  1807.0000  1977.0000
* pc7L
  PLA     pc7L    0.00000    0.89509   -0.44589      0.000     -0.145000   1817.00
* pc7R
  PLA     pc7R    0.00000    0.89509   -0.44589      0.000      0.145000   1967.00
* pc8L
  PLA     pc8L    0.00000    0.89509    0.44589      0.000     -0.145000   1967.00
* pc8R
  PLA     pc8R    0.00000    0.89509    0.44589      0.000      0.145000   1817.00
*--------------------------------------------------------------------------------------
* MA_IN
  REC    MA_IN   0.0000000000000000   0.0000000000000000 647.0000000000000000
                 0.0000000000000000   0.0000000000000000  50.0000000000000000
                 0.0000000000000000   1.5000000000000000   0.0000000000000000
                 1.3200000000000000   0.0000000000000000   0.0000000000000000
```

```
* MA_OUT
  RCC   MA_OUT     0.0000     0.0000   647.0000     0.0000     0.0000    50.0000
             15.0000
* MA_IN2
  REC   MA_IN2   0.0000000000000000   0.0000000000000000 2014.0000000000000000
                 0.0000000000000000   0.0000000000000000   15.0000000000000000
                 0.0000000000000000   1.5000000000000000    0.0000000000000000
                 1.3200000000000000   0.0000000000000000    0.0000000000000000
* MA_OUT2
  RCC  MA_OUT2     0.0000     0.0000  2014.0000     0.0000     0.0000    15.0000
             15.0000
  END
* Reg # 1
* BLKOUT; assigned material: Blackhole; mat # (1)
BLKOUT   5   +blk  -Vac
* Reg # 2
* VacOut; assigned material: Vacuum; mat # (2)
VacOut  15   +Vac  -Qyk  -Dyk
           -( +MA_OUT  -MA_IN )
           -( +MA_OUT2  -MA_IN2 )
           -( +colPL +pa7L +pa8L)
           -( +colPR -pa7R -pa8R)
           -( +colSL +pb7L +pb8L)
           -( +colSR -pb7R -pb8R)
           -( +colTL +pc7L +pc8L)
           -( +colTR -pc7R -pc8R)
* Reg # 3
* PriJawL; assigned material: INERM180; mat # (35)
PriJawL   5   +colPL +pa7L +pa8L
* Reg # 4
* PriJawR; assigned material: INERM180; mat # (35)
PriJawR   5   +colPR -pa7R -pa8R
* Reg # 5
* SecJawL; assigned material: INERM180; mat # (35)
SecJawL   5   +colSL +pb7L +pb8L
* Reg # 6
* SecJawR; assigned material: INERM180; mat # (35)
SecJawR   5   +colSR -pb7R -pb8R
* Reg # 7
* Mask; assigned material: INERM180; mat # (35)
Mask   5   +MA_OUT  -MA_IN
* Reg # 8
* Mask; assigned material: INERM180; mat # (35)
Mask2   5   +MA_OUT2  -MA_IN2
```

```
* Reg # 9
* QSteel_1; assigned material: STEEL; mat # (34)
QSteel_1   5   +QSTOUT1  -QSTIN1
* Reg # 10
* QCoils_1; assigned material: NB3SNCU; mat # (36)
QCoils_1   5   +QSTIN1  -( +Qpc1  +Qpc2 ) -( +Qpc3  +Qpc4 ) -( +Qpc5  +Qpc6 ) -(
           +Qpc7  +Qpc8 ) -Qapert_1
* Reg # 11
* QSteel_2; assigned material: STEEL; mat # (34)
QSteel_2   5   +QSTOUT2  -QSTIN2
* Reg # 12
* QCoils_2; assigned material: NB3SNCU; mat # (36)
QCoils_2   5   +QSTIN2  -( +Qpc2_1  +Qpc2_2 ) -( +Qpc2_3  +Qpc2_4 ) -( +Qpc2_5  +Qpc2_6 ) -(
           +Qpc2_7  +Qpc2_8 ) -Qapert_2
* Reg # 13
* QYoke; assigned material: Iron; mat # (11)
QYoke   5   +Qyk  -QSTOUT1  -QSTOUT2
* Reg # 14
* DSteel_1; assigned material: STEEL; mat # (34)
DSteel_1   5   +DSTOUT1  -DSTIN1
* Reg # 15
* DCoils_1; assigned material: NB3SNCU; mat # (36)
DCoils_1   5   +DSTIN1  -( +Dpc3  +Dpc4 ) -( +Dpc7  +Dpc8 ) -Dapert_1
* Reg # 16
* DTitan_1; assigned material: Titanium; mat # (24)
DTitan_1   5   +( +( +DSTIN1  +Dpc3  +Dpc4 )| +( +DSTIN1  +Dpc7  +Dpc8 )) -Dapert_1
* Reg # 17
* DTitan_2; assigned material: Titanium; mat # (24)
DTitan_2   5   +( +( +DSTIN2  +Dpc2_3  +Dpc2_4 )| +( +DSTIN2  +Dpc2_7  +Dpc2_8 )) -Dapert_2
* Reg # 18
* DSteel_2; assigned material: STEEL; mat # (34)
DSteel_2   5   +DSTOUT2  -DSTIN2
* Reg # 19
* DCoils_2; assigned material: NB3SNCU; mat # (36)
DCoils_2   5   +DSTIN2  -( +Dpc2_3  +Dpc2_4 ) -( +Dpc2_7  +Dpc2_8 ) -Dapert_2
* Reg # 20
* DYoke; assigned material: Iron; mat # (11)
DYoke   5   +Dyk  -DSTOUT1  -DSTOUT2
* Reg # 21
* QM1_BS1; assigned material: STEEL; mat # (34)
QM1_BS1   5   +QBS1_O  -QBS1_V
* Reg # 22
* QCB_1; assigned material: STEEL; mat # (34)
QCB_1   5   +QBORE1  -QBORE2
```

```
* Reg # 23
* QM1_Vac; assigned material: Vacuum; mat # (2)
QM1_Vac   5   +QBS1_V | +( +QBORE2  -QBS1_O )| +( +Qapert_1  -QBORE1 )
* Reg # 24
* QTitan_1; assigned material: Titanium; mat # (24)
QTitan_1   5   +( +( +QSTIN1  +Qpc1  +Qpc2 )| +( +QSTIN1  +Qpc3  +Qpc4 )| +( +QSTIN1
           +Qpc5  +Qpc6 )| +( +QSTIN1  +Qpc7  +Qpc8 )) -Qapert_1
* Reg # 25
* DM1_BS1; assigned material: STEEL; mat # (34)
DM1_BS1   5   +DBS1_O  -DBS1_V
* Reg # 26
* DCB_1; assigned material: STEEL; mat # (34)
DCB_1   5   +DBORE1  -DBORE2
* Reg # 27
* DM1_Vac; assigned material: Vacuum; mat # (2)
DM1_Vac   5   +DBS1_V | +( +DBORE2  -DBS1_O )| +( +Dapert_1  -DBORE1 )
* Reg # 28
* QM1_BS2; assigned material: STEEL; mat # (34)
QM1_BS2   5   +QBS2_O  -QBS2_V
* Reg # 29
* QCB_2; assigned material: STEEL; mat # (34)
QCB_2   5   +QBORE3  -QBORE4
* Reg # 30
* QM2_Vac; assigned material: Vacuum; mat # (2)
QM2_Vac   5   +QBS2_V | +( +QBORE4  -QBS2_O )| +( +Qapert_2  -QBORE3 )
* Reg # 31
* QTitan_2; assigned material: Titanium; mat # (24)
QTitan_2   5   +( +( +QSTIN2  +Qpc2_1  +Qpc2_2 )| +( +QSTIN2  +Qpc2_3  +Qpc2_4 )| +( +QST
           +Qpc2_5  +Qpc2_6 )| +( +QSTIN2  +Qpc2_7  +Qpc2_8 )) -Qapert_2
* Reg # 32
* DM1_BS2; assigned material: STEEL; mat # (34)
DM1_BS2   5   +DBS2_O  -DBS2_V
* Reg # 33
* DCB_2; assigned material: STEEL; mat # (34)
DCB_2   5   +DBORE3  -DBORE4
* Reg # 34
* DM2_Vac; assigned material: Vacuum; mat # (2)
DM2_Vac   5   +DBS2_V | +( +DBORE4  -DBS2_O )| +( +Dapert_2  -DBORE3 )
* Reg # 35
* Dump; assigned material: NB3SNCU; mat # (36)
*Dump   5   +Dump
* Reg # 36
* TerJawL; assigned material: INERM180; mat # (35)
TerJawL   5   +colTL +pc7L +pc8L
```

```
* Reg # 37
* TerJawR; assigned material: INERM180; mat # (35)
TerJawR   5   +colTR -pc7R -pc8R
  END
GEOEND
ASSIGNMAT   BLCKHOLE    BLKOUT
ASSIGNMAT     VACUUM    VacOut
ASSIGNMAT   INERM180   PriJawL
ASSIGNMAT   INERM180   PriJawR
ASSIGNMAT   INERM180   SecJawL
ASSIGNMAT   INERM180   SecJawR
ASSIGNMAT   INERM180      Mask
ASSIGNMAT   INERM180     Mask2
ASSIGNMAT       IRON  QSteel_1
ASSIGNMAT    NB3SNCU  QCoils_1
ASSIGNMAT       IRON  QSteel_2
ASSIGNMAT    NB3SNCU  QCoils_2
ASSIGNMAT       IRON     QYoke
ASSIGNMAT       IRON  DSteel_1
ASSIGNMAT    NB3SNCU  DCoils_1
ASSIGNMAT   TITANIUM  DTitan_1
ASSIGNMAT   TITANIUM  DTitan_2
ASSIGNMAT       IRON  DSteel_2
ASSIGNMAT    NB3SNCU  DCoils_2
ASSIGNMAT       IRON     DYoke
ASSIGNMAT      STEEL   QM1_BS1                           1.0
ASSIGNMAT      STEEL     QCB_1                           1.0
ASSIGNMAT     VACUUM   QM1_Vac                           1.0
ASSIGNMAT   TITANIUM  QTitan_1                           1.0
ASSIGNMAT      STEEL   DM1_BS1                           1.0
ASSIGNMAT      STEEL     DCB_1                           1.0
ASSIGNMAT     VACUUM   DM1_Vac                           1.0
ASSIGNMAT      STEEL   QM1_BS2
ASSIGNMAT      STEEL     QCB_2
ASSIGNMAT     VACUUM   QM2_Vac
ASSIGNMAT   TITANIUM  QTitan_2
ASSIGNMAT      STEEL   DM1_BS2
ASSIGNMAT      STEEL     DCB_2
ASSIGNMAT     VACUUM   DM2_Vac
*ASSIGNMAT    NB3SNCU      Dump
ASSIGNMAT   INERM180   TerJawL
ASSIGNMAT   INERM180   TerJawR
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
USRGCALL        21.0      23.0     2.194                              QPOLEFLD
USRGCALL        24.0      24.0       2.5       6.2                    OUTERFLD
USRGCALL        25.0      27.0       16.                              DPOLEFLD
MGNFIELD        20.0       0.2      0.10       0.0       0.0       0.0
*--------------------------------------------------------------------------------
PHYSICS      55000.0                                                  LIMITS
*********Discard Particles****************
*Discard neutrinos, antineutrinos
DISCARD          5.0       6.0
*--------------------------------------------------------------------------------
*BIASING
BIASING          1.0       0.2                  2.        36.
EMF-BIAS        255.        1.        1.        2.        36.          LPBEMF
*CUTOFFS
EMFCUT          1E-3      1E-3        0.        2.        36.
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

*--------------------------------------------------------------------------------
*                                        Scoring
*--------------------------------------------------------------------------------
*                                   Primary Collimator
*--------------------------------------------------------------------------------
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
*
*USRBIN          10.0    ENERGY      71.0      15.0   -0.1300      110.    E_ColL
*USRBIN         -15.0  -15.1300       -10.      120.       60.      120.         &
*
*USRBIN          10.0    ENERGY      72.0      15.0   15.1300      110.    E_ColR
*USRBIN         -15.0    0.1300       -10.      120.       60.      120.         &
*--------------------------------------------------------------------------------
*                                 Secondary Collimator
*--------------------------------------------------------------------------------
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
*
*USRBIN          10.0    ENERGY     101.0      15.0   -0.2600      585.   E_Col2L
*USRBIN         -15.0  -15.2600       465.      120.       60.      120.         &
*
*USRBIN          10.0    ENERGY     102.0      15.0   15.2600      585.   E_Col2R
*USRBIN         -15.0    0.2600       465.      120.       60.      120.         &
*
*--------------------------------------------------------------------------------
*--------------------------------------------------------------------------------
*                                      Quadrupole
*--------------------------------------------------------------------------------
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
```

```
USRBIN          10.0    ENERGY     -41.0       8.0       8.0     1707.     E1_MQ
USRBIN          -8.0      -8.0      707.       64.       64.      200.         &
*
USRBIN          11.0    ENERGY     -32.0      15.0       0.0     1707. EQ_RadOUT
USRBIN           2.5       0.0      707.      25.0      180.      200.         &
*
*--------------------------------------------------------------------------------
*                                        Masks
*--------------------------------------------------------------------------------
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
*USRBIN          11.0    ENERGY      81.0      15.0       0.0      697.  E_M1_Rad
*USRBIN           1.0       0.0      647.      28.0      180.      050.         &
*
*USRBIN          11.0    ENERGY      91.0      15.0       0.0     2064.  E_M2_Rad
*USRBIN           1.0       0.0     2014.      28.0      180.      050.         &
*--------------------------------------------------------------------------------
*                                        Dipole
*--------------------------------------------------------------------------------
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
USRBIN          11.0    ENERGY     -52.0      15.0       0.0     3574. EB_RadOUT
USRBIN           2.5       0.0     2074.      25.0      180.      150.         &
*
USRBIN          10.0    ENERGY     -61.0       8.0       8.0     3574.     EB_MQ
USRBIN          -8.0      -8.0     2074.       32.       32.      150.         &
*--------------------------------------------------------------------------------
*                                        Fluence
*--------------------------------------------------------------------------------
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
USRBIN          10.0  ALL-PART    -194.0       2.5       2.5      710.     FLUE1
USRBIN          -2.5      -2.5      100.       50.       50.      305.         &
*
USRBIN          10.0  ALL-PART    -195.0       2.5       2.5     2075.     FLUE2
USRBIN          -2.5      -2.5     1705.       50.       50.      370.         &
*
USRBIN          10.0  ALL-PART    -196.0       1.5       4.5     2475.     FLUE3
USRBIN          -1.5      -4.5     2075.       30.       90.      400.         &
*
USRBIN          10.0  ALL-PART    -197.0       1.5       4.5     2875.     FLUE4
USRBIN          -1.5      -4.5     2475.       30.       90.      400.         &
*
*--------------------------------------------------------------------------------
USERWEIG          0.        0.        4.        0.        0.        0.
SOURCE        63.508       0.0                                        M_1_noX
```

```
BEAM        -50000.0                                                  PROTON
RANDOMIZ         1.0
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
START            10.                                     0.0
STOP
```