Simulations of stored electrons in the Penning trap between the KATRIN spectrometers

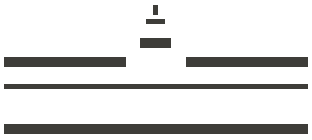
Jan D. Behrens

Diploma thesis

Institut für Kernphysik
Mathematisch-Naturwissenschaftliche Fakultät
Westfälische Wilhelms-Universität Münster

July 2012

revised edition (November 12, 2012)



Westfälische
Wilhelms-Universität
Münster

Referent: Prof. Dr. C. Weinheimer

Korreferent: Prof. Dr. J. Wessels

It is not unscientific to make a guess,
although many people who are not in science think it is.

— Richard P. Feynman

Contents

1. Introduction

Neutrinos are one of the most fascinating topics in modern particle physics. After their first theoretical description, a lot of progress has been made on both experimental and theoretical side. It was proven that neutrinos amazingly are not massless particles, and flavor oscillations are possible. As neutrinos play a significant role in the evolution of the universe – the neutrino density of the universe is $336\,\mathrm{cm}^{-3}$ – knowledge of the absolute neutrino mass scale is very important for cosmology, particle- and astrophysics.

KATRIN is a model-independent experiment to determine the mass of the electron neutrino by kinematic measurements of the Tritium beta-decay with a sensitivity of $0.2\,\mathrm{eV/c^2}$. To reach this sensitivity, a low background level of $10^{-2}\,\mathrm{s}^{-1}$ in the main spectrometer alone is crucial. An important part of the complex experimental setup is the so-called MAC-E filter, which allows to measure the energy spectrum of the Tritium beta-decay at high sensitivity and low background. This filter type is employed in both KATRIN spectrometers, namely the pre-spectrometer and the main spectrometer. The pre-spectrometer itself is supposed to keep uninteresting beta electrons from entering the analysis part of the experiment, and thus reduce background levels. However, the combination of two spectrometers at high negative potentials creates an intrinsic Penning trap in the experimental setup.

As this Penning trap will contribute to unwanted background if not taken care of, a series of investigations were conducted in the past. Two dissertations dealt with experimental studies of the trap, and proposed a number of countermeasures to reduce trap-generated background. The present diploma thesis is a direct continuation of the latest of these studies, and was aimed at further investigating the trap characteristics by computer simulations. Additionally, two of the three proposed countermeasures were examined by simulations: The “electron catcher”, a stationary pin that directly removes electrons from the trap; and the “pulsed coil”, which is supposed to remove electrons from the trap by applying an additional magnetic field.

The present diploma thesis is structured as follows:

- Chapter 2 gives an introduction to neutrino physics, and contains an overview of different approaches to determine the neutrino mass. It is focused on the method which is employed at the KATRIN experiment, but does not neglect the interesting results of neutrino disappearance experiments that were published earlier this year.
- In chapter 3 an overview to the structure of the KATRIN experiment is given, with focus on the MAC-E filter that is employed in both KATRIN spectrometers. The MAC-E filter is not directly relevant to this diploma thesis, but knowledge about the underlying principles is useful to understand the Penning trap between the spectrometers.
- Chapter 4 provides a general discussion of Penning traps with respect to the KATRIN experiment. The characteristic trap properties are explained, followed by a discussion of mechanisms that can induce unwanted background. The remainder of this chapter gives details on the Penning trap between the KATRIN spectrometers, which was investigated in this thesis. The chapter closes with a review of previous investigations of this specific trap, and discusses the proposed countermeasures.
- The particle simulation software and related tools which were employed in this work are discussed in chapter 5. A general introduction to the Kassiopeia simulation tool is given, followed by a detailed discussion of the employed electric and magnetic field algorithms, which are one of the most relevant parts to the particle simulation. The chapter then goes on to present a new interface that improves the management of Kassiopeia simulations, and introduces a first implementation of the geometry management tool Kreator. Thereafter, the newly implemented visualization techniques of Kassiopeia are explained.
- Finally, the simulations of the Penning trap between the KATRIN spectrometers and their results are presented in chapter 6. After a discussion of the trap characteristics, such as trap dimensions and particle frequencies, the chapter moves on to discuss the effects of the two proposed electron removal methods that were closely investigated in this thesis.
- The appendix chapters A to C contain a number of source code excerpts, and a discussion of the geometric setups that were used in the simulations.

2. Neutrino physics

2.1. The discovery of neutrinos

After the first studies of radioactivity by HENRI BECQUEREL in the late 19th century, the further investigation of radioactive decay by Becquerel together with MARIE and PIERRE CURIE and others lead to a much deeper understanding of the structure of matter than ever before. ERNEST RUTHERFORD established the separation of radioactive decay into two different types, namely α and β decay. These decay types emit different kinds of radiation with different abilities to penetrate objects. Later on, the γ radiation was identified as a third type of radiation.

The beta decay was then further investigated by JAMES CHADWICK, who for the first time in history achieved to measure the energy spectrum of decay electrons. He found that, in contrast to the already-known spectra of alpha and gamma decay, the beta spectrum was continuous. This is inconsistent with the assumption of a two-body decay, as it would imply a violation of the law of energy conservation. To solve this problem, the *neutrino* was postulated by WOLFGANG PAULI in 1930 as a neutral particle with roughly the same mass as the electron, and spin $1/2$ [Pau30]. With this assumption, the beta decay could be described as a three-body process, which would lead to the known spectrum as this neutrino could now carry away some of the decay energy, thus making the electron energy spectrum continuous. Pauli chose the name "neutron" for the proposed particle. After the neutral nucleon – which we today know as "neutron" – was discovered by Chadwick in 1932, it soon became clear that this could not be the particle postulated by Pauli, most notably for its mass being far too large. In response, the now-established name "neutrino" ("small neutral one" in Italian) was introduced two years later by ENRICO FERMI in his theory of the beta decay. At this time, it was not possible to directly measure these neutrinos. The weak interaction cross-section requires either a very strong neutrino source or a large-mass detector to compensate the low interaction cross-sections. Large neutrino detectors with high sensitivity were not available before the last third of the 20th century, but nuclear reactors provided the necessary strong neutrino sources. However, it took some time until the experimental verification that neutrinos do indeed exist.

The first experimental evidence for neutrinos was found more than 20 years after their prediction by FREDERICK REINES, CLYDE L. COWAN et al. in 1956, who were able to use the neutrino flux of the Savannah River Reactor (South Carolina, USA) [Rei59]. They used a target that consisted of water and cadmium chloride, where a neutrino can react with a proton of the water and create a positron and a neutron by inverse beta decay:

$$\mathrm{p} + \bar{\nu}_e \rightarrow \mathrm{n} + \mathrm{e}^+ . \tag{2.1}$$

The neutron would then be absorbed by the cadmium, which has a large cross-section for capturing neutrons. The excited daughter nucleus decays to its ground state after a characteristic delay of a few microseconds, emitting a photon with specific energy. Both the emitted 511 keV photon from the positron annihilation and the photon from the cadmium decay could then be detected by scintillators around the target, and the measurement of these signals with a characteristic time delay provided the first evidence for the existence of neutrinos.

Since then, a large number of experiments for further investigation of neutrino properties were performed. One of the most interesting results was the discovery that neutrinos carry a mass, which will be explained in more detail in the next section.

2.2. Massive neutrinos and neutrino oscillations

2.2.1. Solar neutrinos

It is well-known that the Sun is powered by nuclear fusion, mainly through the proton-proton chain reaction that transmutes four hydrogen nuclei:

$$4\mathrm{p} \rightarrow {}^4_2\mathrm{He} + 2\mathrm{e}^+ + 2\nu_e + 26.73\,\mathrm{MeV} . \tag{2.2}$$

The excess energy is radiated as thermal energy, while the neutrinos are emitted through a number of different processes. The energy spectrum of the emitted neutrinos can be calculated theoretically, and when neutrino detectors became sensitive to the energies involved, the theoretical spectrum was compared to the experimental results. It soon became obvious that there is a discrepancy between the theoretical predictions and the experimental measurements: the number of detected neutrinos was only about one third to one half as large as the predicted number.

The first experiment to successfully measure the solar neutrino flux and detect the discrepancy was the HOMESTAKE experiment by RAYMOND DAVIS, JOHN N. BACALL et al. in the late 1960s. This discrepancy became known as the *solar neutrino problem*. It was further confirmed by several follow-up experiments. A possible solution to this problem is provided by the theory of neutrino oscillations, which is explained in detail in the following section.

2.2.2. Neutrino oscillations

The theory of neutrino oscillation is based on the concept that the flavor eigenstates $\{\langle\nu_\alpha|\,,\ \ \alpha = e,\mu,\tau\}$ and the mass eigenstates $\{\langle\nu_i|\,,\ \ i = 1,2,3\}$ both create eigenbases of the parameter space, but are not identical. This leads to the unitarian transformation $\hat{U}$ between both bases: [Ott09]

$$\langle\nu_\alpha| = \sum_i \hat{U}_{\alpha i}\,\langle\nu_i|\,. \tag{2.3}$$

Therefore the flavor eigenstates can be written as a linear combination of the mass eigenstates, and vice versa. The transformation matrix U is called the PONTECORVO-MAKI-NAKAGAWA-SAKATA (PMNS) matrix, and with the mixing angles ϑ_{ij} and the abbreviations $c_{ij} = \cos\vartheta_{ij}$ and $s_{ij} = \sin\vartheta_{ij}$ it has the following form:

$$U = \begin{pmatrix} c_{12}c_{13} & c_{12}c_{13} & s_{13}e^{-\mathrm{i}\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{\mathrm{i}\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{\mathrm{i}\delta} & s_{23}c_{13} \\ s_{12}c_{23} - c_{12}c_{23}s_{13}e^{\mathrm{i}\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{\mathrm{i}\delta} & c_{23}c_{13} \end{pmatrix}. \tag{2.4}$$

The factor "CP-violating phase" $e^{\mathrm{i}\delta}$ takes into account the CP-violation probability of neutrino oscillations. It is worth mentioning that the PMNS matrix is fully defined by three mixing angles ϑ_{12}, ϑ_{13} , ϑ_{23} and the phase δ.

The time development of a mass eigenstate can then be retrieved by applying Schrödinger's equation (using $\hbar = c = 1$):

$$|\nu_i(t)\rangle = e^{-\mathrm{i}E_i t}\,|\nu_i\rangle\ , \tag{2.5}$$

with

$$E_i = \sqrt{p_i^2c^2 + m_i^2c^4} \tag{2.6}$$

$$\approx pc + \frac{m_i^2c^4}{2pc} \approx E + \frac{m_i^2c^4}{2E} \qquad \text{for} \quad p \gg m_i \tag{2.7}$$

denoting the neutrino energy.

The time development implies that a former pure flavor eigenstate $|\nu_\alpha\rangle$ gets an admixture of the state $|\nu_\beta\rangle$ after some time t has passed:

$$|\nu(t)\rangle = \sum_i U_{\alpha,i}\cdot e^{-\mathrm{i}E_i t}\,|\nu_i\rangle = \sum_{i,\beta} U_{\alpha,i}U^*_{\beta,i}\cdot e^{-\mathrm{i}E_i t}\,|\nu_\beta\rangle\ . \tag{2.8}$$

It can also be said that the three different mass eigenstates propagate with different phase velocities after a flavor eigenstate has been created, leading to a time-dependent oscillation probability for a neutrino along its flight path. As neutrinos are highly relativistic particles, the time dependency corresponds to a dependency on the path length the neutrino has traveled

($L \approx ct$). The oscillation probability for one flavor α transforming into another flavor β can then be written as:

$$P(\nu_\alpha \to \nu_\beta) = \left|\langle \nu_\beta(t) | \nu_\alpha(t) \rangle\right|^2 \tag{2.9}$$

$$\begin{aligned} &= \delta_{\alpha\beta} \\ &\quad - 4 \sum_{i>j} \Re(U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j}) \sin^2\left(\Delta m^2_{ij} \frac{L}{4E}\right) \\ &\quad + 2 \sum_{i>j} \Re(U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j}) \sin^2\left(\Delta m^2_{ij} \frac{L}{4E}\right) . \end{aligned} \tag{2.10}$$

Here, L denotes the distance from the neutrino-generating source and E the particle energy. The squared neutrino mass difference Δm^2_ν is given in eV. $\Re(x)$ is the real part of the complex number x.

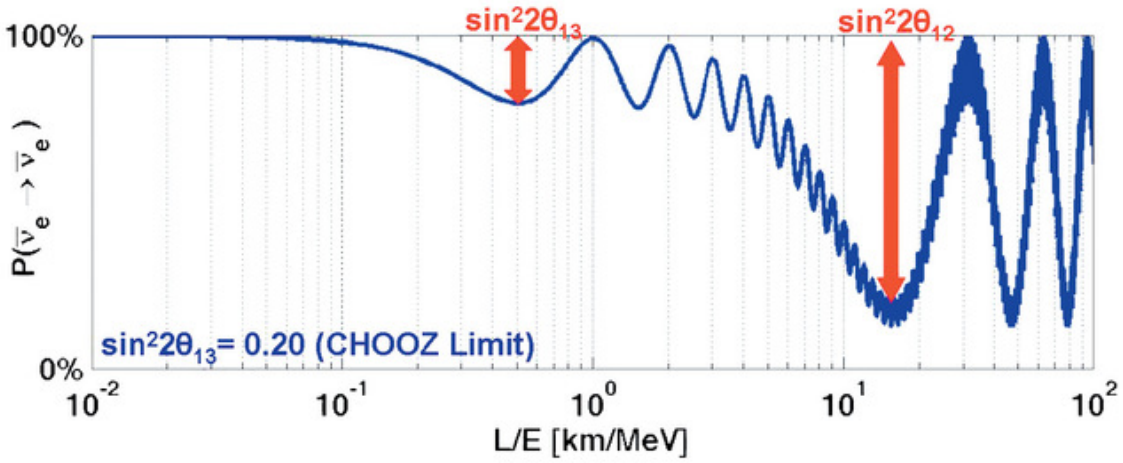


Figure 2.1.: Transition probability for the $\bar{\nu}_e \to \bar{\nu}_e$ process. The probability is plotted against L/E and remains almost constant for low path lengths, whereas for higher values the probability shows strong oscillations, allowing transitions into another flavor eigenstate in some cases. Figure taken from [KAT06].

A simplified model of neutrino oscillation, using only two flavor eigenstates α and β, can be implemented to show the oscillation effects. The transformation matrix (2.4) is reduced to a 4×4 matrix with a single mixing angle ϑ in this model, and the mass difference can be written as $\Delta m = |m_1^2 - m_2^2|$. The oscillation probability then simplifies to

$$P(\nu_\alpha \to \nu_\beta) = \langle \nu_\beta | \nu(t) \rangle = \sin^2 2\vartheta \cdot \sin^2\left(\Delta m^2 \frac{L}{4E}\right) . \tag{2.11}$$

Figure 2.1 illustrates the dependency of oscillation probability on the path length for the $\bar{\nu}_e \to \bar{\nu}_e$ process.

2.2.3. Evidence for neutrino oscillations

The theory of neutrino oscillation described in the previous section allows for two methods of detecting neutrino oscillations experimentally:

Disapperance experiments: The possible disappearance of a neutrino flavor which is emitted at the source can be measured at a certain distance at fixed energy. According to equation (2.9), the disappearance probability depends on the mixing angle and the squared-mass difference.

Apperance experiments: Another, similar method to detect neutrino oscillations would be to measure the appearance of certain neutrino flavors that are not created at the source.

Both methods, or combinations of them, were and still are used to investigate neutrino oscillations. Some of the results will be discussed in this section to give an overview of the current research topics.

Solar and atmospheric neutrinos

The KAMIOKANDE (Kamioka Nucleon Decay Experiment) experiment was one of the first neutrino experiments to measure the solar neutrino flux. Although it was originally designed to observe the possible decay of the proton, its huge water Čerenkov detector was also suited to detect electron (and muon) neutrinos. After some upgrades leading to the KAMIOKANDE-II experiment, it became indeed possible to measure the flux of solar electron neutrinos. However, the measured rate was only about $1/2$ of the rate predicted by the theoretical model of the Sun. The HOMESTAKE experiment, on the other hand, measured a neutrino flux of $1/3$ of the predicted value. This deficit was known as the *solar neutrino problem*.

Additionally the Kamiokande-II experiment was able to detect muon neutrinos created in the Earth's atmosphere. Although the results hinted at a deficit of these atmospheric muon neutrinos, the detector was not large enough to produce significant results. This problem became known as the *atmospheric neutrino deficit*.

After the theory of neutrino oscillation was formed, it became clear that the deficits in both solar and atmospheric neutrinos might be connected to the oscillation of neutrinos from one flavor into another. To investigate this idea, several experiments were conducted, and each required a large amount of target material to produce significant results.

One of the experiments of this kind is SUPER-KAMIOKANDE, which started from the original Kamiokande experiment. In 1998, it was the first experiment that could prove the oscillation of atmospheric neutrinos. The Super-Kamiokande experiment used a huge reservoir of pure water together with an array of photo-multipliers to detect atmospheric electron and muon neutrinos. The results showed a dependency of the number of detected neutrinos on the incident angle, i. e. less muon neutrinos were detected for incident angles that correspond to a longer path length (where neutrinos have to travel through the Earth, instead of just the atmosphere). With the theory of neutrino oscillation, this can be explained easily: if the neutrinos travel a longer distance before being detected, they have a higher probability to oscillate into tau neutrinos, which are not detected at the experiment.

Since neutrino oscillations as shown in section 2.2.2 imply a non-zero neutrino mass, Super-Kamiokande also was the first experiment to experimentally prove that massive neutrinos exist. The following limits to one of the mixing angles and the squared neutrino mass difference were established using a best-fit method: [SK08]

$$\tan^2 \vartheta_{23} = 0.40 \tag{2.12}$$

$$\Delta m^2_{23} = 6.03 \times 10^{-5}\,\mathrm{eV}^2\,. \tag{2.13}$$

The resulting angle $\vartheta_{23} = 45^\circ$ corresponds to maximal mixing.

Shortly thereafter, the SNO (Sudbury Neutrino Observatory) collaboration found evidence for oscillations in solar neutrinos that are produced from ^{8}B decay in the sun. These neutrinos have a very high energy of up to 14.6 MeV. With the SNO experiment it was possible to detect the neutrino flux for all three flavors, together with the flux of electron neutrinos only. For these measurements, three reactions were used:

$$\nu_e + \mathrm{d} \rightarrow \mathrm{p} + \mathrm{p} + \mathrm{e}^- \quad \text{charged current (CC)} \tag{2.14}$$

$$\nu_x + \mathrm{d} \rightarrow \mathrm{p} + \mathrm{n} + \nu_x \quad \text{neutral current (NC)} \tag{2.15}$$

$$\nu_x + \mathrm{e}^- \rightarrow \nu_x + \mathrm{e}^- \quad \text{elastic scattering (ES)}\,. \tag{2.16}$$

The recoil electrons from the ES and CC processes could be measured directly by their emission of Čerenkov light. The neutrons created in the NC process were detected indirectly by emission of characteristic γ-rays after their capture on deuterons or ^{35}Cl nuclei [SNO01].

The SNO results show that only about one third of the solar electron neutrinos reaching the Earth still have their initial flavor, while two thirds have oscillated into muon and tau neutrinos: [SNO11]

$$\frac{\varphi(\nu_e)}{\varphi(\nu_e) + \varphi(\nu_{\mu,\tau})} = 0.347 \pm 0.029\,. \tag{2.17}$$

As only electron neutrinos are generated by nuclear fusion inside the Sun, the appearance of other neutrino flavors proves the oscillation theory and thus provides evidence for massive neutrinos. These results explain the previously mentioned deficit of solar neutrinos e. g. at the Homestake experiment. The detected total neutrino flux is also consistent with the one predicted by the solar standard model (SSM).

Accelerator and reactor neutrinos

Apart from the naturally generated neutrinos, it is also possible to use particle accelerators and nuclear reactors as strong sources of neutrinos. One of the advantages of these methods is the very high neutrino flux that can be achieved.

Accelerator neutrinos are used e. g. at the MINOS and OPERA experiments. Both experiments use hadronic processes to create highly pure beams of muon neutrinos. While MINOS measures the disappearance of muon neutrinos, at OPERA the $\nu_\mu \rightarrow \nu_\tau$ channel is used, making it one of the few neutrino appearance experiments. Both experiments are also known for their time-of-flight measurements of neutrinos.

For reactor neutrinos, it is even possible to use modern-day nuclear power plants in normal operation as sources. This concept is indeed being used at various experimental sites, e. g. DOUBLE CHOOZ and DAYA BAY. Both experiments use reactor neutrinos for their measurements to detect the disappearance of electron neutrinos at different distances from the reactor core. While Double Chooz currently uses only one detector at 1050 m distance, as the other one is still under construction, Daya Bay uses a total of six detectors at three detector sites.

In both experiments, the survival probability of an electron neutrino is measured, i. e. the probability for a neutrino to arrive at the detector *without* changing its flavor. It depends on the mixing angles $\vartheta_{13,12}$, the squared-mass differences $\Delta m^2_{31,21}$, the neutrino energy and the path length (compare (2.9)): [DB12]

$$\begin{aligned} P_{surv} \approx 1 &- \sin^2 2\vartheta_{13} \cdot \sin^2\left(\Delta m^2_{31}\frac{L}{4E}\right) \\ &- \cos^4\vartheta_{13}\sin^2 2\vartheta_{12} \cdot \sin^2\left(\Delta m^2_{21}\frac{L}{4E}\right). \end{aligned} \tag{2.18}$$

The squared-mass differences $\Delta m_{31,21}$ and mixing angle ϑ_{12} are well known from previous experiments: [MIN11] [Kam08]

$$|\Delta m_{31}|^2 = 2.32^{+0.12}_{-0.08} \times 10^{-3}\,\mathrm{eV}^2 \tag{2.19}$$

$$|\Delta m_{21}|^2 = 7.59^{+0.21}_{-0.21} \times 10^{-5}\,\mathrm{eV}^2 \tag{2.20}$$

$$\tan^2\vartheta_{12} = 0.47^{+0.06}_{-0.005}. \tag{2.21}$$

Therefore, (2.18) allows to determine the ϑ_{13} mixing angle from measuring the survival probability.

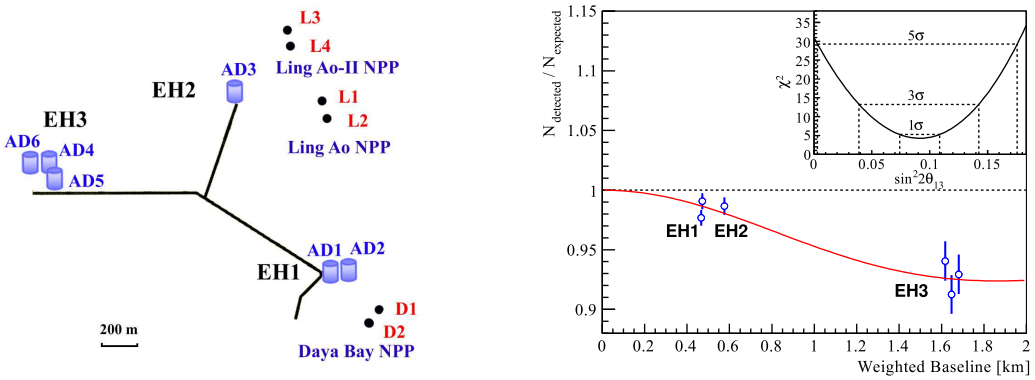
Lately, the Daya Bay collaboration was able to publish significant new results on the ϑ_{13} mixing angle. At the time the data was taken (Dec. 2011 to Feb. 2012), the experiment consisted of two near and one far detector that measure anti-electron neutrinos emitted from a total of six reactor cores. The combination of near and far detectors enabled the Daya Bay experiment to not only measure the amount of neutrinos in each detector, but also the difference in count-rate at different path lengths. Since all detectors are functionally identical, this very much nullifies the systematic uncertainties and minimizes uncorrelated reactor uncertainties, enabling the Daya Bay experiment to gain a very high sensitivity.

In figure 2.2 the results of the mentioned measurement period are shown together with a schematic view of the experimental setup. The results are plotted relative to the count-rate that would be expected if no oscillations did occur. It can clearly be seen that a quite large deficit of 6 % occurs at the EH3 detector site, and also matches the measurements from EH1 and EH2. The best fit gives a result on the mixing angle of

$$\sin^2 2\vartheta_{13} = 0.092 \pm 0.005(\mathrm{sys}) \pm 0.016(\mathrm{stat}), \tag{2.22}$$

with a significance of 5.2σ. These results indicate that the neutrino mixing is larger than estimated by previous studies [DB12].

The Daya Bay results are supported by measurements from Double Chooz and RENO [DC12] [REN12]. These latest measurements provide strong evidence for a fairly high mixing angle of $\vartheta_{13} \approx 9°$. This is much higher than previous, less-significant results from 2011 would anticipate.



(a) Setup of the Daya Bay experiment.

(b) Daya Bay results for ϑ_{13}.

Figure 2.2.: Latest measurements of the ϑ_{13} mixing angle by the Daya Bay experiment. On the left, the experimental setup is shown. Daya Bay uses a combination of near and far detectors (AD), located in at three detector sites (EH). The six reactor cores from which the neutrinos are emitted are grouped into three pairs. (Note that some reactors are horizontally displaced for better visibility in this plot.) The right figure shows the measurement results as published by the Daya Bay collaboration. Plotted is the number of events measured at each detector against the expected number without oscillations. Especially for EH3 a considerable deficit can be seen. The red curve represents a best fit on the data and strongly supports $\vartheta_{13} > 0$. The inset shows a plot of χ^2 against $\sin^2 2\vartheta_{13}$. A non-zero mixing angle can be deducted from the data with a significance of 5.2σ. Figures taken from [DB12].

2.3. The absolute neutrino mass scale

Although all of the methods mentioned above provided very important results and were able to determine or restrict the neutrino mixing parameters and measure the squared mass differences between the three mass eigenstates, some questions are still left open:

- The oscillation experiments are only able to measure mass differences (compare (2.9)), but not the absolute masses.
- Also the sign of the mass differences can not be measured by this kind of experiments, although the ordering of the neutrino masses has to be known to discriminate between normal hierarchy ($m_1 < m_2 < m_3$) and inverted hierarchy ($m_3 < m_1 < m_2$).

In conclusion, neutrino oscillation experiments can not answer all questions regarding neutrinos and have to be supported by other, alternative concepts. On the other hand, cosmological studies on neutrinos are always dependent of the theoretical model which is used. A model-independent approach to directly measure the neutrino mass is therefore favorable to provide answers to the questions given above, and to provide complementary results to the cosmological measurements. One model-independent approach is the direct measurement of the neutrino mass from beta-decay kinematics, which will be discussed in the next section.

A complementary method to gain knowledge of the neutrino mass is to look for the neutrino-less double-beta decay ($0\nu\beta\beta$). This process is not allowed in the Standard Model, but would exist if the neutrino is a so-called *Majorana fermion*, i. e. it is its own anti-particle. In this case the decay rate $\Gamma_{0\nu\beta\beta}$ would scale with an effective neutrino mass. This kind of experiments would measure a coherent sum of neutrino mass eigenstates, which may be subject to destructive interference due to the phase factors in the mixing matrix. Uncertainties arise from the nuclear matrix element of the neutrino-less double-beta decay, which is not known exactly.

2.3.1. Direct measurements of the neutrino mass

The direct measurement of the neutrino mass provides a model-independent result and can therefore support other measurements. The used method to determine the neutrino mass is to measure the energy spectrum of the beta decay. This method is completely based on relativistic kinematics and thus does not rely on any model-specific assumptions, and will be explained in the remainder of this section.

The beta-decay electron energy spectrum depends on the mass of the neutrino $m(\nu_e)$. Because the mass of the electron neutrino is not exactly defined (see above), the energy spectrum is a superposition of single spectra depending on the three mass eigenstates. Since the differences between these spectra are very small, an effective neutrino mass can be defined as the incoherent sum of neutrino eigenstates:

$$m(\nu_e)^2 = \sum_{i=1}^{3} \left|U_{ei}\right|^2 m(\nu_i)^2 \,. \tag{2.23}$$

To better understand the principle of these kinematic mass measurements, we shall investigate an electron emitted in nuclear beta decay. From the law of energy conservation it follows that the electron is missing the kinetic energy which is carried away by the neutrino. If the decay energy E_0 is known, the maximum possible electron energy can be written as

$$E_{max} = E_0 - m(\nu_e)c^2 \,. \tag{2.24}$$

The direct measurement of the spectrum endpoint would thus give away the neutrino mass. But as always, things are not that simple, and the low count rate (see figure 2.3) together with background events at the detector prevents such a measurement. Also the decay energy is not known with sufficient precision. A better approach therefore is to take multiple measurements *below* the end-point, which will then allow to reconstruct the form of the spectrum in this range.

Neglecting the final state distribution of the daughter nucleus, and applying Fermi's Golden Rule, the beta spectrum can be written as: [Ott09]

$$\frac{d\dot{N}}{dE} = R(E) \cdot \sqrt{(E_0 - E)^2 - m(\nu_e)^2 c^4} \cdot \Theta(E_0 - E - m(\nu_e)c^2)\,, \tag{2.25}$$

$$R(E) = \frac{G_F^2}{2\pi^3\hbar^7} \cos^2\vartheta_C \cdot |M|^2 F(Z,E) \cdot p(E + m_e c^2)(E_0 - E)\,. \tag{2.26}$$

Here, E is the energy and p the momentum of the decay electron. Z is the charge of the daughter nucleus, G_F the Fermi constant, ϑ_C the Cabibbo angle and M the nuclear matrix element. $F(Z,E)$ is the Fermi function that describes the electrostatic interaction. By fitting the spectrum to the experimental results, one can obtain the decay energy E_0 and the effective neutrino mass $m(\nu_e)$. An important aspect of this method is that it is independent of the theoretical model, and only relies on kinematics.

This method was successfully used in a number of experiments already. In most of these, Tritium was used as a source for decay electrons. The beta decay reaction of Tritium can be written as:

$$^3_1\mathrm{H} \rightarrow {}^3_2\mathrm{He}^+ + \mathrm{e}^- + \bar{\nu}_e\,. \tag{2.27}$$

Especially compared to Rhenium-187, which is also used as beta source for neutrino mass measurements, Tritium provides a number of advantages:

- The endpoint E_0 of the spectrum at 18.6 keV is very low compared to other possible sources. This means that more decay events fall in the range near the endpoint and are therefore relevant for the measurements, which in turn reduces the total measurement time. However, this is even more favorable for ^{187}Rh with $E_0 = 2.47\,\mathrm{keV}$.
- The half-life of $^3_1$H with 12.3 a is rather short compared to 4.3×10^{10} a for ^{187}Rh. This makes it possible to have strong source with a low column density that is still handleable.
- For Tritium, the beta decay is super-allowed and therefore the nuclear matrix element in (2.26) does not depend on the energy of the decay electron. This does not apply to Rhenium, where the beta decay is uniquely forbidden.
- The atomic structure of Tritium is simple enough for the atomic corrections to be calculated with fairly high accuracy. This also applies to its daughter nucleus $^3_1\mathrm{H}^+$.
- Finally, Tritium has a low nuclear charge, and thus the inelastic scattering of decay electrons inside the source is small.

For these reasons, Tritium will be used as an electron source at KATRIN. It was also already used in the two predecessor experiments in Mainz and Troitsk, which were able to provide upper limits on the neutrino mass of 2 eV [KAT04], and the KATRIN experiment can benefit from the experience gained in these experiments. The KATRIN experiment and its predecessors will be discussed in the next chapter.

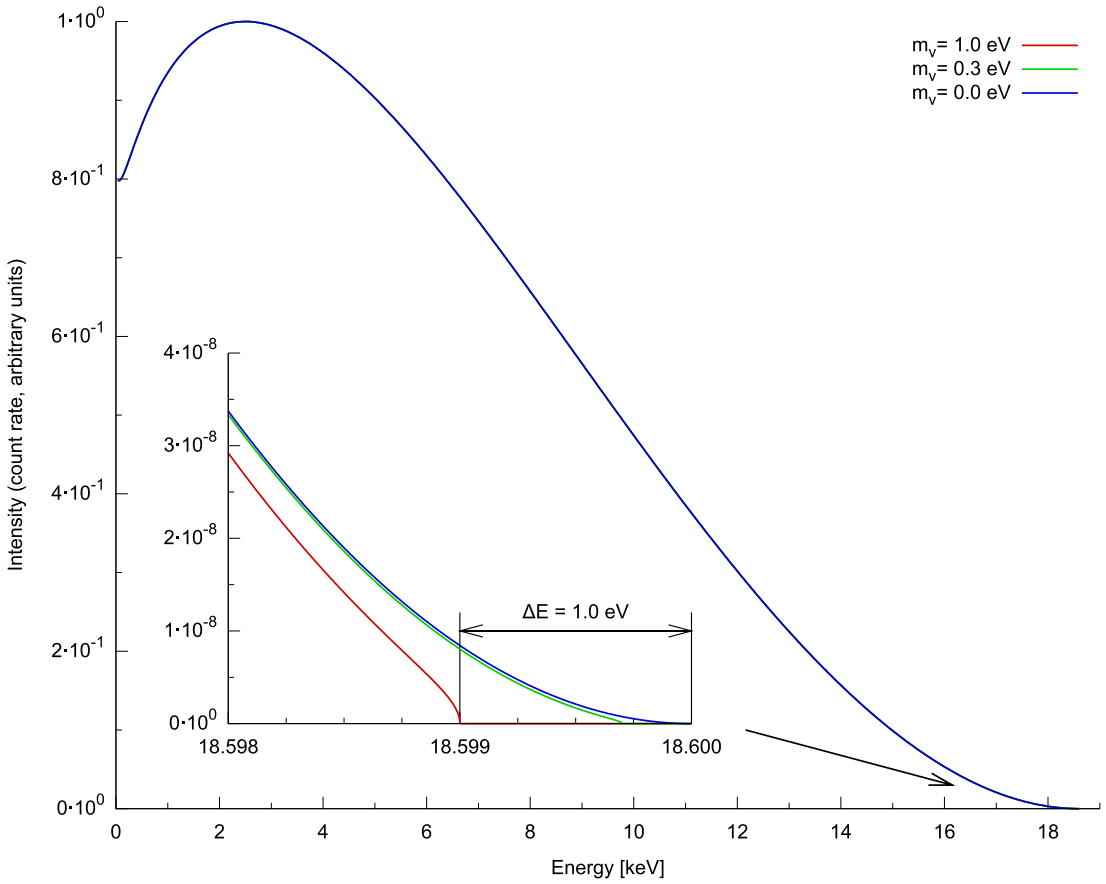


Figure 2.3.: Energy spectrum of the Tritium beta decay. The spectrum has an endpoint of $E_0 = 18.6\,\text{keV}$. Spectra for three different neutrino masses up to 1 eV are shown, which can best be seen within the region of a few eV around the endpoint (see inset). The count rate is given in arbitrary units.
(Note: This figure was published at the Wikimedia Commons, and is used in several Wikipedia articles.)

3. The KATRIN experiment

The main goal of the KATRIN experiment is to determine the neutrino mass as defined in (2.23) by precise measurements of Tritium beta-decay kinematics. Taking three years of measurement time (excluding overhead), KATRIN will be able to discover a neutrino mass of $m(\bar{\nu}_e) = 0.35\,\mathrm{eV/c^2}$ ($0.30\,\mathrm{eV/c^2}$) with 5σ (3σ) significance, or to place an upper limit of $m(\bar{\nu}_e) < 0.2\,\mathrm{eV/c^2}$ at 90% C.L. [KAT04].

The predecessor experiments of Mainz and Troitsk have yielded the lowest laboratory limits on the neutrino mass of today, with $2.3\,\mathrm{eV/c^2}$ and $2.05\,\mathrm{eV/c^2}$ at 95% C.L., respectively [Ott09]. The KATRIN experiment uses a similar concept of measuring the decay kinematics, but will achieve a 100-times higher sensitivity. One main aspect here is the increased size of the MAC-E filter, a design feature that is used in all three experiments. At KATRIN, the main filter has a length of about 23 m and a diameter of 9.8 m, and will help to increase the sensitivity of the experiment. However, also many other features of the experiment have to be improved for a successful measurement.

In the following sections, the general structure of the experiment will be discussed. The focus of this diploma thesis is on an unavoidable Penning trap between the two spectrometers, therefore this aspect will be explained in more detail in chapter 4.

3.1. Structure of the KATRIN experiment

The KATRIN experiment can be split up into two major sections: The source-and-transport section (STS) and the spectrometer-and-detector section (SDS). Each of these sections contains a number of components. An overview is given in figure 3.1.

Tritium source: The windowless gaseous Tritium source (WGTS) is the main electron source of the KATRIN experiment and also one of the most complex parts. It needs to achieve a high intensity, as the measured region of the decay spectrum only makes up a small part

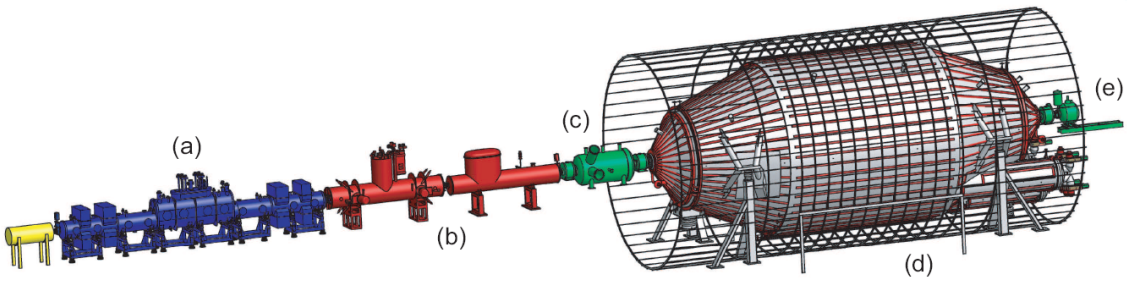


Figure 3.1.: Overview of the KATRIN experiment. (a) windowless gaseous Tritium source, **WGTS**; (b) differential and cryo pumping sections, **DPS/CPS**; (c) pre-spectrometer, **PS**; (d) main spectrometer, **MS**; (e) focal plane detector, **FPD**. Sections a,b are also called the source-transport-section, **STS**, while sections c,d,e are called the spectrometer-detector-section, **SDS**. Figure taken from [KAT04].

of the total number of emitted electrons, which results in a fairly low count-rate at the detector. It is also crucial to take into account all systematic effects.

A gaseous source is used at KATRIN, since a solid source with the required intensity of 10^{11} Bq/s would show charge effects of about 10 V [Ott09], and thus can not be used without affecting the high-sensitivity measurements that require an energy resolution of $\Delta E < 1$ V. The gaseous source is free of such an effect, but has a much more complex design.

The source has a length of 10 m with a diameter of 90 mm and is cooled down to 30 K. Tritium is injected in the middle of the tube and pumped to both ends, where it is removed again by turbomolecular pumps. With this concept a column density of 5×10^{17} molecules/cm^2 can be achieved. The setup implements a closed Tritium loop, which means that extracted Tritium is purified and then injected again. The intended flow rate of 5×10^9 molecules/s corresponds to 40 g T_2 throughput per day. This is one aspect that supported the experiment's location to be at the Forschungszentrum Karlsruhe (FZK), which hosts Europe's largest Tritium lab. The decay electrons emitted in the source are guided to the transport section by a homogeneous magnetic field of 3.6 T. The magnetic flux tube transported from the source to the detector has a size of 191 T cm^2.

Transport section: Before the decay electrons can be analyzed in the spectrometers, it is necessary to drastically reduce the remaining gas flow from the WGTS. With the high column density at the source and an ultra-high vacuum of 10^{-11} mbar in the spectrometers, a combination of two pumping sections is needed to reduce the gas flow into the spectrometer section by a factor 10^{14}.

In the first step, Tritium is removed from the beam tube by turbomolecular pumps in the differential pumping section (DPS). After that the remaining Tritium is removed in the cryo pumping section (CPS), by absorbing the gas molecules onto a frozen Argon layer on the surface of the beam tube. The pumping efficiency is improved by magnetically guiding the electron's flux tube through bends in the beam line. The neutral Tritium molecules remain moving in a straight line, and can be removed at the bends.

Pre-spectrometer: In addition to the main spectrometer that implements the high-resolution energy measurement, the pre-spectrometer is used beforehand to reject unwanted low-energy electrons. The pre-spectrometer is a MAC-E filter (see section 3.2) like the one used for the main spectrometer, but with smaller dimensions: It has a length of 3.4 m and a diameter of 1.7 m. The flux tube is shaped by two magnets that generate a field of 4.5 T each. This filter will reduce the electron flux to ca. $10^3\,\mathrm{s}^{-1}$ by applying a retarding potential roughly 300 V lower than the endpoint energy. As the real analysis is done in the main spectrometer, the energy resolution of the pre-spectrometer $\Delta E \lesssim 100\,\mathrm{V}$ is not very important.

The pre-spectrometer was also used to test mechanisms that are important to understand in these kind of spectrometers, e. g. tests of the inner electrode system and other methods to reduce background, or examinations of Penning traps in the spectrometer.

Another important aspect of the pre-spectrometer is that together with the main spectrometer it creates a deep Penning trap between the two spectrometers. This might create high background rates due to stored particles, which need to be removed for a successful measurement. One solution to this problem would be to operate the pre-spectrometer only at low potential, thus significantly reducing the trap depth [Glu08]. More details on this topic and other proposed solutions will be discussed in the next section.

Main spectrometer: The main spectrometer is the main tool for energy analysis at the KATRIN experiment. The spectrometer is a MAC-E filter with a length of over 23 m and a diameter of 9.8 m. The magnetic field from the pinch magnet at the detector side is decreased by a factor of 20 000 towards the analyzing plane, resulting in an energy resolution of 0.93 eV in the endpoint region of the beta-decay energy spectrum (see section 3.2 for details).

An inner wire electrode system is implemented to reduce background from cosmic muons and radioactive decay in the spectrometer surface. The spectrometer is also surrounded by air coils that correct for the Earth's magnetic field.

By varying the electric retarding potential U_0 at the analyzing plane and counting electrons at the detector, it is possible to measure an integrated spectrum of the Tritium beta-decay electrons. The main spectrometer further reduces the electron flux from $10^3\,\mathrm{s}^{-1}$ at the source side to $< 1\,\mathrm{s}^{-1}$ at the detector side. Obviously the exact count rate depends on the retarding potential. Additionally, the 6 T *pinch magnet* at the detector side of the spectrometer reflects electrons with large polar angles. This effect is further discussed in the next chapter.

Detector: The detector consists of a 148-pixel PIN diode. The segmentation of the detector allows a spatial resolution of counted electrons, which is necessary to correct for inhomogeneities in the retarding potential and the magnetic field in the analyzing plane. It also improves the analysis of background noise from the main spectrometer.

As the main spectrometer is responsible for the high-resolution analysis of decay electrons, the detector itself does not need to have a explicitly high energy resolution. However, an energy resolution of about 1 keV in the end-point region is implemented to improve the suppression of background. The detector also features active and passive shielding to lower the background, and can make use of a post-acceleration electrode with an applied

voltage of up to 30 kV. An additional detector magnet with a strength of up to 5.6 T creates a magnetic shielding effect and guides electrons to the detector surface. The magnet can also be used to adjust the flux tube diameter at the detector plane [Doe11].

Future enhancements of the KATRIN experiment are also taken into consideration. One of the ideas worth mentioning here is a time-of-flight (TOF) measurement mode that, if it proves to be practically achievable, will allow to measure differential spectra instead of integrated ones. The TOF measurement mode is based on the fact that the flight-time of an electron through the spectrometer is a function of its energy. While the integrated spectrum from normal measurement contains only the number of events per retarding potential, a TOF differential spectrum will include the time-of-flight for each single event, thus gaining more information from the same measurement. However, TOF measurements require both a start and a stop signal. While the detector provides a sufficient time resolution of ≤ 50 ns, there is currently no way to create a start signal from electrons entering the main spectrometer without interference or loss of precision. The "gated pre-spectrometer" approach, where the retarding potential of the pre-spectrometer is continuously pulsed to provide a trigger for TOF measurements, will in fact provide a sensitivity of 0.025 eV^2; hence the net benefit of the TOF mode will be negligible. However, with the "gated pre-spectrometer" method the TOF mode could provide a way to look for sterile, heavy neutrinos (i. e. warm dark matter) in the keV range. [Stei12]

In this diploma thesis the focus lies on the two spectrometers and especially the interface between the two. The source-transport section and the detector will therefore not be further discussed here. Instead, one of the most relevant features of the two spectrometers, the MAC-E filter, will be explained in the next section.

3.2. The KATRIN MAC-E filter

Both spectrometers of the KATRIN experiment make use of the MAC-E filter technique (MAC-E: Magnetic Adiabatic Collimation with Electrostatic filtering). This type of spectrometer is beneficial in the case of KATRIN for several reasons. The neutrino mass is best determined by measuring the end-point region of the Tritium beta-spectrum, where the count-rate is very small. Therefore, the combination of a source with high intensity and a spectrometer with large angular acceptance is necessary to achieve the high statistics required at KATRIN. The MAC-E filter design fulfills these requirements, and was indeed used successfully in the two predecessor experiments. In the next section, this type of filter is explained in detail.

A MAC-E filter is shown schematically in figure 3.2. It consists of two main parts:

- A magnetic field that is created by superconducting solenoids at the entrance and exit of the spectrometer. The maximum field strength B_{max} is achieved at the center of the pinch magnet at the spectrometer exit. It drops to its minimal value B_{min} in the analyzing plane in the middle of the spectrometer.

- An electric retarding potential is created by the spectrometer's electrode system. Its maximum value U_0 is reached in the analyzing plane.

Electrons that enter the spectrometer undergo a cyclotron motion along the magnetic field. Since this motion is adiabatic, the magnetic flux enclosed by the electron's trajectory remains constant. This implies that the transversal energy of an electron entering the spectrometer is transformed into longitudinal energy in the analyzing plane (see below for details). As the magnetic field lines are parallel to the electric field lines in the analyzing plane, only electrons with a high enough longitudinal energy can pass the spectrometer and finally arrive at the detector. The transformation of transversal into longitudinal energy ensures a high angular acceptance, since electrons with high starting angles are not rejected automatically. This is one of the main features of a MAC-E filter type filter. In conclusion, the spectrometer can be described as integrating high-pass filter, and the beta spectrum can be measured by scanning an energy range with the retarding potential.

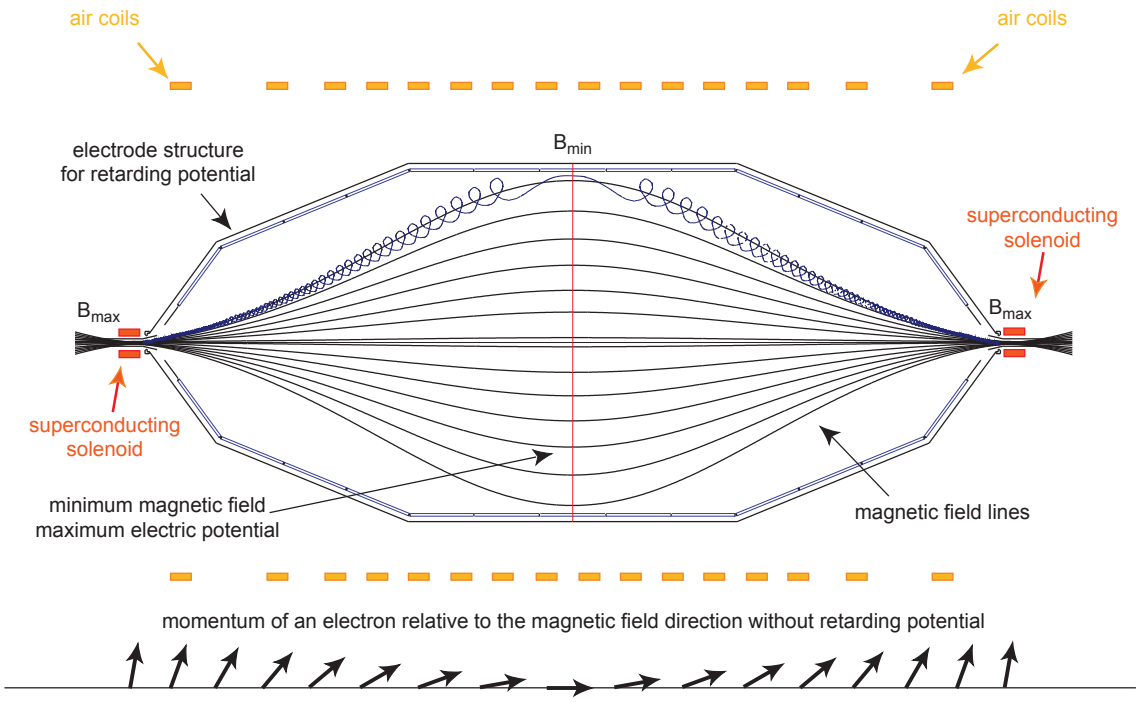


Figure 3.2.: Schematic view of the MAC-E filter used in the KATRIN main spectrometer. Shown are the spectrometer hull with the wire electrode system, along with the solenoids (red) and the air coils to compensate magnetic fields from the outside (orange). The cyclotron motion of a particle (blue) around a magnetic field line (black) is drawn exaggerated. At the bottom of the figure the changing direction of momentum for an electron is visible, as it moves through the spectrometer. It is obvious how the conservation of magnetic moment transfers transversal momentum into longitudinal momentum towards the analysis plane in the center of the spectrometer (vertical red line). Here the magnetic field has its minimum, while the electric field is maximal. Figure taken from [Hug08].

3.2.1. Adiabatic transformation of kinetic energy

To further discuss the adiabatic energy transformation in the MAC-E filter, we will investigate the trajectory of an electron that is started inside one of the solenoids.

For the case of Tritium beta-decay electrons at the endpoint, the according Lorentz factor is

$$\gamma = \frac{1}{\sqrt{1 - v^2/c^2}} = 1.04 \approx 1\,, \tag{3.1}$$

and therefore a non-relativistic calculation will be used.

The kinetic energy E of an electron entering the MAC-E filter with an angle ϑ relative to the magnetic field line can be split up into transversal and longitudinal components:

$$E = E_\perp + E_\parallel\,, \tag{3.2}$$

with

$$E_\perp = E \cdot \sin^2\vartheta \tag{3.3}$$

$$E_\parallel = E \cdot \cos^2\vartheta\,. \tag{3.4}$$

The transversal component $E_\perp$ is also the energy of the cyclotron motion. The cyclotron angle ϑ describes the angle of the electron momentum relative to the local magnetic field.

To calculate the cyclotron radius r_c of the electron, we consider that the centripetal force F_c of the electron must be equal to the Lorentz force F_L induced by the magnetic field B:

$$F_L = eBv_\perp \doteq \frac{m_e v_\perp^2}{r_c} = F_c\,. \tag{3.5}$$

Here $v_\perp$ is the transversal speed of the electron. The kinetic energy in this direction is then

$$E_\perp = \frac{1}{2} m_e v_\perp^2\,. \tag{3.6}$$

Putting (3.6) into (3.5) gives

$$r_c = \frac{\sqrt{2 m_e E_\perp}}{eB} \tag{3.7}$$

for the cyclotron radius. It depends on both magnetic field and transversal kinetic energy (and thus, cyclotron angle) of the electron.

The magnetic moment of the electron due to its cyclotron motion is

$$\mu = |\boldsymbol{\mu}| = \frac{E_\perp}{B}\,. \tag{3.8}$$

In the case of adiabatic guiding, i. e. the magnetic field changes only slightly in one cyclotron rotation, the magnetic flux enclosed by the cyclotron motion, and therefore the magnetic moment, is conserved:

$$\Phi = \int B dA = \text{const.} \tag{3.9}$$

$$\gamma\mu \approx \mu = \frac{E_\perp}{B} = \text{const.}. \tag{3.10}$$

When the magnetic field strength drops to its minimum in the analyzing plane, the transversal kinetic energy will be reduced, too. Due to energy conservation the transversal kinetic energy is transformed into longitudinal kinetic energy, according to equation (3.2). This is also shown schematically at the bottom of figure 3.2, where the momentum vector of the electron changes its orientation slowly while the electron is passing through the spectrometer.

Equation (3.8) also allows us to derive the energy resolution of a MAC-E type filter. We assume that an electron starts at maximum field strength B_{max} in the center of the solenoid with maximum kinetic energy E_{start}^{max} fully stored in the transversal component $E_{start,\perp}^{max}$. At the analyzing plane, the magnetic field drops to B_{min}, and the electron still retains some amount of transversal energy $E_{ap,\perp}^{max}$ which is defined by the magnetic field strength:

$$\mu = \frac{E_{start}^{max}}{B_{max}} = \frac{E_{start,\perp}^{max}}{B_{max}} = \frac{E_{ap,\perp}^{max}}{B_{min}} = \text{const.}. \tag{3.11}$$

In the analyzing plane only the longitudinal kinetic energy determines if an electron can pass through the spectrometer, and the remaining transversal energy is not taken into account. This "missing energy" therefore defines the energy resolution of an ideal MAC-E filter:

$$\Delta E = E_{ap,\perp}^{max} = E_{start}^{max} \cdot \frac{B_{min}}{B_{max}}. \tag{3.12}$$

In the case of KATRIN, the maximum starting kinetic energy of $E_0 \approx 18.6\,\text{keV}$ is given by the endpoint energy of the Tritium spectrum. The magnetic field drops from $B_{max} = 6\,\text{T}$ at the pinch magnet to $B_{min} = 3\,\text{G} = 3 \times 10^{-4}\,\text{T}$ at the analyzing plane. The energy resolution can therefore be calculated to

$$\Delta E = E_0 \cdot \frac{B_{min}}{B_{max}} = 18.6\,\text{keV} \cdot \frac{3 \times 10^{-4}\,\text{T}}{6\,\text{T}} = 0.93\,\text{eV}. \tag{3.13}$$

The relative energy resolution is then

$$\frac{\Delta E}{E_0} = 5 \times 10^{-5}. \tag{3.14}$$

3.2.2. Transmission function

As stated in the last section, the kinetic energy is mostly transformed into longitudinal energy in the analyzing plane, and only this component is relevant for electrons to pass the retarding potential U_0. Because the initial transversal energy depends on the starting angle relative to the magnetic field line, electrons with different starting angles, but fixed kinetic energy, will have different longitudinal energies in the analyzing plane. Therefore the retarding potential at which electrons with a fixed kinetic energy are allowed to pass through the filter is influenced by the starting angle. This then has an impact on the achievable energy resolution.

For an electron to pass through the filter, its longitudinal energy in the analyzing plane has to be greater than zero:

$$E_{ap,\parallel} > 0\,. \tag{3.15}$$

This condition for the longitudinal energy can be rewritten as

$$\begin{aligned} E_{ap,\parallel} &= E_{ap} - E_{ap,\perp} && (3.16)\\ &= E_{ap} - E_{start,\perp} \cdot \frac{B_{min}}{B_{max}} && (3.17)\\ &= \left(E_{start} - qU_0\right) - E_{start} \sin^2 \vartheta_{start} \cdot \frac{B_{min}}{B_{max}} > 0\,. && (3.18) \end{aligned}$$

Solving to ϑ_{start} gives the transmission condition

$$\vartheta_{start} \leq \vartheta_{max} = \arcsin \sqrt{\frac{E_{start} - qU_0}{E_{start}} \cdot \frac{B_{min}}{B_{max}}}\,, \tag{3.19}$$

i. e. only electrons with a small enough starting angle can pass the filter. The maximum starting angle therefore limits the theoretical 2π-acceptance of the MAC-E filter to a smaller fraction. The relation of the solid angle accepted by the MAC-E filter $\Delta\Omega$ and the theoretical, full acceptance of 2π is

$$\frac{\Delta\Omega}{2\pi} = 1 - \cos\vartheta_{max}\,. \tag{3.20}$$

Together with equation (3.19), the transmission function of the MAC-E filter can then be written as

$$T(E,U_0) = \begin{cases} 0 & , E \leq qU_0 \\ 1 - \sqrt{1 - \frac{E-qU_0}{E} \cdot \frac{B_{max}}{B_{min}}} & , qU_0 < E \leq qU_0 + \Delta E \\ 1 & , qU_0 + \Delta E \leq E \end{cases}, \tag{3.21}$$

with ΔE as defined in equation (3.12). The width of the transmission function is defined by the magnetic field ratio.

At the KATRIN experiment, the starting magnet of the WGTS does not provide the maximum magnetic field, but only a fraction $B_{source} = 3.6\,\mathrm{T} < B_{max}$. Electrons emitted from the source therefore have to travel into a region with a higher magnetic field of $B_{max} = 6\,\mathrm{T}$ at the

spectrometer, and are thus subject to the magnetic mirror effect. This effect reflects any electrons whose angle is higher than the critical angle, since here longitudinal energy is transformed into transversal energy for the same reasons as above, thus limiting the number of electrons that can pass through:

$$\vartheta_{mirror} = \arcsin\sqrt{\frac{B_{source}}{B_{max}}} \approx 50.77^{\circ}\,. \tag{3.22}$$

As electrons with a higher starting angle also have a longer path length in the WGTS due to their cyclotron motion, and therefore have a higher probability for scattering processes, this limit on the starting angle provides a useful feature of the KATRIN setup and helps to reduce background.

The analytic transmission function of the MAC-E filter (3.21) has to be rewritten to include this additional effect:

$$T(E,U_0) = \begin{cases} 0 & , E \leq qU_0 \\ \dfrac{1-\sqrt{1-\frac{E_{start}-qU_0}{E_{start}}\cdot\frac{B_{source}}{B_{min}}}}{1-\sqrt{1-\frac{\Delta E}{E_{start}}\cdot\frac{B_{source}}{B_{max}}}} & , qU_0 < E \leq \Delta E\,. \\ 1 & , \Delta E \leq E \end{cases} \tag{3.23}$$

In figure 3.3, this modified transmission function is shown after it has been normalized to 1 in the region of maximal transmission.

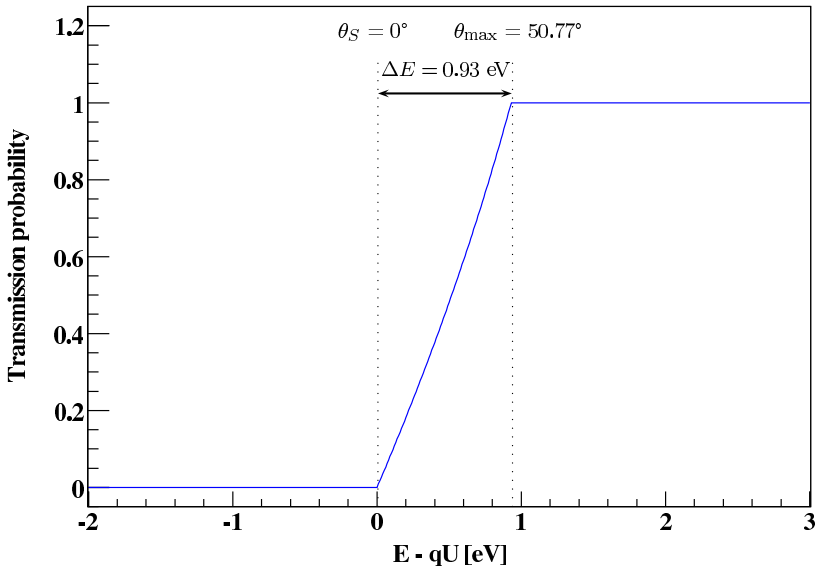


Figure 3.3.: The transmission function of the KATRIN MAC-E filter. The plot is a direct result of equation (3.23). The width of the transmission function is given by the starting angle ϑ_{start} and defines the energy resolution of the filter. Figure taken from [Wol08].

4. Penning traps at KATRIN

A low background count-rate is crucial for the success of the KATRIN experiment. Only a small amount of electrons from the Tritium beta spectrum with energies near the endpoint passes both spectrometers, leading to a small number of events that provide data relevant to the neutrino mass measurement. To achieve a high signal-to-noise ratio it is therefore necessary to keep the number of background events as small as possible.

One source for unwanted background are Penning traps in the experimental setup. These traps can store electrons either from Tritium decay in the source or from external sources like radioactive decays in the electrode materials, field emission, or muons hitting the electron surfaces. These stored electrons are then able to create secondary electrons and ions by scattering. While secondary electrons are kept in the trap volume, ions can travel further to the detector region and create secondary electrons on their own. These electrons are then likely to reach the detector and produce unwanted background signals. On the other hand, electrons which are stored directly in the Penning trap between the spectrometers can build up a space-charge in the trap volume and disturb signal electrons. Depending on the properties of each trap, different ways exist to either get rid of the trap in a way that it is no longer relevant to the experiment, or to remove the stored electrons from the trap.

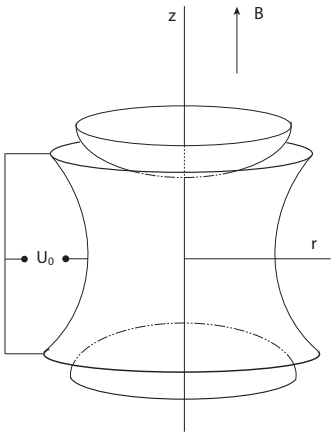
This chapter starts with a general discussion of Penning traps and why they are relevant to the KATRIN experiment. Finally, the previous studies on the Penning trap between the spectrometers – which is also the topic of this work – are explained, together with the proposed ways to counteract the background that would be generated by this trap.

4.1. General discussion of Penning traps

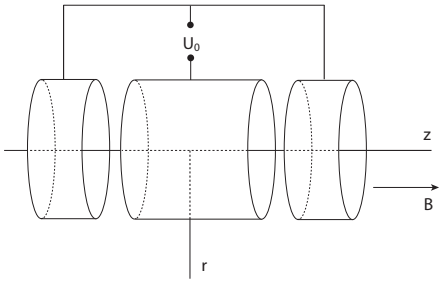
Generally speaking, a Penning trap consists of a combination of electric and magnetic fields that is able to store charged particles inside a closed volume. A particle inside the trap will move around in a specific motion, but it can not leave the trap if its energy is small enough. A

typical hyperbolic Penning trap uses a cylindrical magnetic coil that provides a homogeneous axial magnetic field and constrains particles in radial direction (see figure 4.1a). An additional electric quadrupole field then further constrains particles axially. In a classical Penning trap, a ring electrode and two electrode caps are used to create the quadrupole field. The particle then moves around in the trap volume according to the Lorentz equation

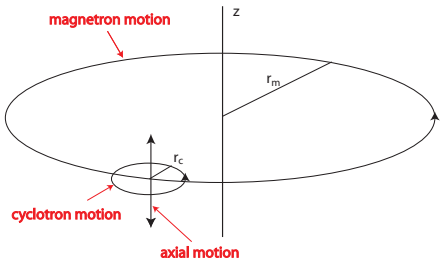
$$\boldsymbol{F}_L = q \cdot (\boldsymbol{E} + \boldsymbol{B} \times \boldsymbol{v}) \,. \tag{4.1}$$



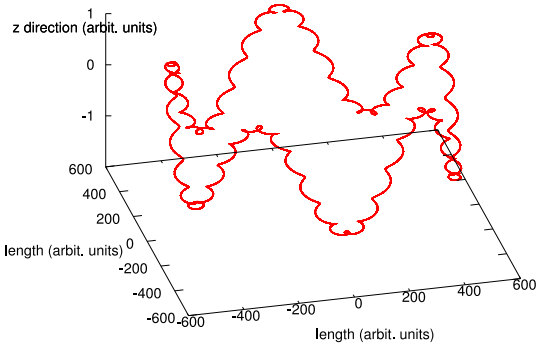
(a) Hyperbolic Penning trap.



(b) Cylindrical Penning trap.



(c) Motion of charged particle in a Penning trap.



(d) Electron motion in a hyperbolic Penning trap.

Figure 4.1.: Typical realization of Penning traps (a) In the hyperbolic Penning trap, the electrodes correspond to equipotential surfaces of an electric quadrupole field. A magnetic field is superimposed to the electric one along the z-axis. (b) In a cylindrical trap, an electric field that matches the field of the hyperbolic trap can be achieved by careful design of the electrode segments. (c,d)The motion of a trapped particle can be split into an axial oscillation, a magnetron and a cyclotron motion. The three components of motion of a particle stored in a hyperbolic trap are shown, together with the corresponding total motion with all three components superimposed. (Fig. a,b,c taken from [Ess04]; fig. d taken from [Hil11].)

In a simple, homogeneous magnetic field, a charged particle will move around a magnetic field line in a circular motion. The angular frequency of this motion is known as *cyclotron frequency*,

and the motion itself is called *cyclotron motion*:

$$\omega_c = \frac{q}{m} B \,. \tag{4.2}$$

In a Penning trap, this motion is a bit more complicated due to the additional electric field. The most intuitive approach is to describe the resulting motion by three independent oscillations that are superimposed:

1. The axial oscillation only depends on the electric fields, as there is no magnetic component perpendicular to the axial direction. This motion can be described as linear oscillation (which is often defined to be along the z-axis) with the frequency

   $$\omega_z = \sqrt{\frac{q}{m}\frac{U}{d^2}} \,. \tag{4.3}$$

   U is often called the "trap depth" because particles with an energy lower than qU will be stored in the trap, while particles with higher energies can escape. The parameter d strongly depends on the trap geometry. For the classical Penning trap with hyperbolic electrodes, it is defined as

   $$d^2 = \frac{1}{2}\left(z_0^2 + \frac{r_0^2}{2}\right) , \tag{4.4}$$

   where z_0 is the electrode distance and r_0 is the trap radius.

2. The cyclotron motion from (4.2) is modified by the electric field and therefore has a slightly different frequency, that depends on the trap depth and dimensions:

   $$\omega_+ = \frac{\omega_c}{2} + \sqrt{\frac{\omega_c^2}{4} - \frac{\omega_z^2}{2}} \approx \omega_c - \frac{U}{2d^2 B} \,. \tag{4.5}$$

3. The slowest motion is the magnetron drift around the center of the trap with the frequency

   $$\omega_- = \frac{\omega_c}{2} - \sqrt{\frac{\omega_c^2}{4} - \frac{\omega_z^2}{2}} \approx \frac{U}{2d^2 B} \,. \tag{4.6}$$

   The total magnetron drift can be split up into two independent components, the $E \times B$ drift and the ∇B drift:

   $$\boldsymbol{v}_{mag} = \boldsymbol{v}_{E\times B} + \boldsymbol{v}_{\nabla B} \,, \tag{4.7}$$

   with

   $$\boldsymbol{v}_{E\times B} = \frac{\boldsymbol{E} \times \boldsymbol{B}}{B} \,, \tag{4.8}$$

   $$\boldsymbol{v}_{\nabla B} = \frac{m_e}{eB^3}\left(\boldsymbol{v}_\parallel + \frac{1}{2}\boldsymbol{v}_\perp\right) \cdot (\boldsymbol{B} \times \nabla B) \tag{4.9}$$

   $$= \frac{1}{\gamma e B^3}\left(\boldsymbol{p}_\parallel + \frac{1}{2}\boldsymbol{p}_\perp\right) \cdot (\boldsymbol{B} \times \nabla B) \,. \tag{4.10}$$

$\boldsymbol{v}_\parallel$ and $\boldsymbol{v}_\perp$ denote the longitudinal and transversal components of the particle velocity relative to the magnetic field vector, respectively (the same applies to the momentum $\boldsymbol{p}$). γ is the Lorentz factor of the particle. In a homogeneous magnetic field, the magnetron drift will be oriented in azimuthal direction, and the particle's motion will be circular around the trap axis. However, any field inhomogeneities will result in additional drift components, e. g. in radial direction [Pic92] [San03].

With these definitions, the motion of a particle stored in the trap can be described by

$$\boldsymbol{r} = \begin{pmatrix} x_0 \cos(-\omega_- t + \varphi_-) + r_c \cos(-\omega_+ t + \varphi_+) \\ y_0 \cos(-\omega_- t + \varphi_-) + r_c \sin(-\omega_+ t + \varphi_+) \\ z_0 \cos(-\omega_z t + \varphi_z) \end{pmatrix} . \tag{4.11}$$

The parameters x_0, y_0, z_0 and $\varphi_-, \varphi_+, \varphi_z$ are free constants. r_c denotes the cyclotron radius. An example of a particle moving in a Penning trap is given in figure 4.1d.

4.2. Penning traps at the KATRIN experiment

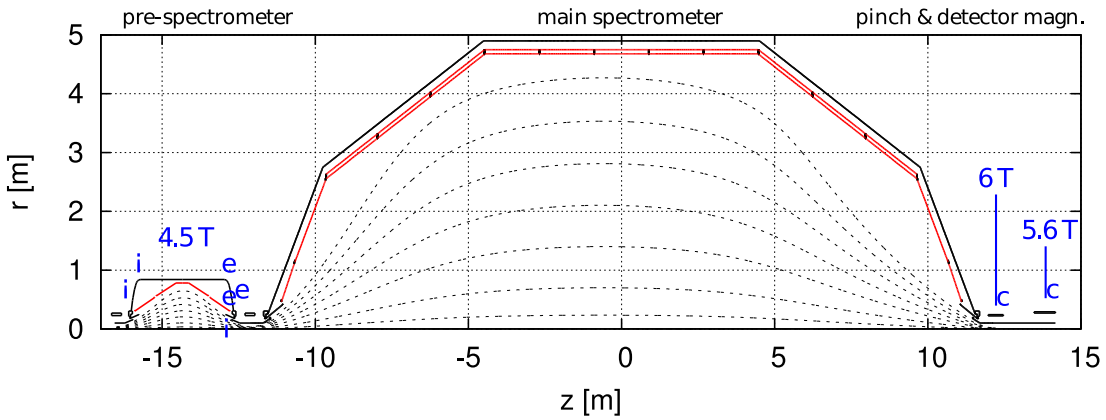


Figure 4.2.: Penning traps in the KATRIN SDS. Because of the high magnetic and electric fields, three major Penning traps occur in this section: one inside each of the spectrometers for positive particles, and one between the two for negative particles. Figure taken from [Val09].

In the KATRIN experiment there are a number of places where Penning traps occur (see figure 4.2). All these traps need to be investigated, as they can produce a fairly large amount of background electrons. Some of the traps can be avoided by implementing carefully designed electrodes at the spectrometer openings. In contrast, an intrinsic Penning trap exists between the spectrometers; it could only be avoided if the pre-spectrometer was not used at all (hence, no voltage would be applied) [Glu08]. This is of course not favorable, as the pre-spectrometer itself was designed to provide a large background suppression, by reducing the number of electrons that enter the main spectrometer by seven orders of magnitude and thus lowering the number of scattered electrons in the analyzing plane. Because the mentioned Penning trap can not be avoided, it is important to provide a mechanism that is able to efficiently remove electrons from the trap in regular intervals.

4.2.1. Contribution to unwanted background

The electrons in the Penning traps contribute to the background by three major mechanisms, which will be explained here in shortened form:

1. The *Townsend discharge* described first by J.S. TOWNSEND can occur by just applying a high electric field between two electrodes at a distance d, with a gas at pressure p between them. An electron between the electrodes will be accelerated towards the anode, and if its energy is sufficient, it can ionize atoms from the residual gas. The produced electrons are then accelerated as well, and can create secondary electrons themselves. This would lead to an avalanche process where more and more charged particles are created. Consequently, a current is exchanged between the electrodes, which could result in the breakdown of the voltage supply unit.

   To see if this process is relevant in the case of KATRIN, one has to look at the Townsend coefficient, which describes the total probability that an electron first ionizes an atom, and then scattering does occur:

   $$\alpha_n = A \cdot p \cdot e^{-\frac{B}{E/p}} . \tag{4.12}$$

   Here E is the magnitude of the electric field and p is the residual gas pressure. The values of the constants A and B depend on the gas filling used, and are given in the literature as $A = 1130\,\mathrm{mm^{-1}\,bar^{-1}}$ and $B = 27\,\mathrm{kV\,mm^{-1}\,bar^{-1}}$ in air. The pressure in the KATRIN experiment will be $p = 10^{-11}\,\mathrm{mbar}$, and typical fields are about $E = 1\,\mathrm{kV/m}$ between electrodes, but can be up to $1\,\mathrm{MV/m}$ in some places. Therefore the Townsend coefficient can be approximated to

   $$\alpha_n \approx A \cdot p , \tag{4.13}$$

   and with the given pressure the resulting coefficient is $\alpha_n \approx 10^{-10}\,\mathrm{mm^{-1}}$. Since this value is very small – only one electron in 10^{10} ionizes further atoms – the Townsend avalanche process itself is suppressed in the case of KATRIN.

2. Another process that could create additional background noise is *field emission*, where electrons can escape from metallic surfaces if large electric fields are present. In that case, the electrons can tunnel through the potential wall at the surface. These free electrons can then initiate a so-called *vacuum breakdown*, i. e. the residual gas becomes conducting and a discharge occurs. Although electric fields of about $E = 1\,\mathrm{MV/m}$ are necessary to enable field emission, these high field strengths can occur at microtips on the surfaces of the materials used.

   Two countermeasures are taken to prevent field emission at the electrode surfaces. On one hand, all parts inside the spectrometers are electro-polished to take away microscopic edges on the surfaces. Because the field strength at these edges depends on the surface curvature, increasing the curvature radii will decrease the probability of field emissions. The second countermeasure is the conditioning of the main spectrometer, which aims at the same result. By applying a voltage larger than the one used during measurement, possibly with inverted polarity, many of the microscopic edges are vaporized beforehand.

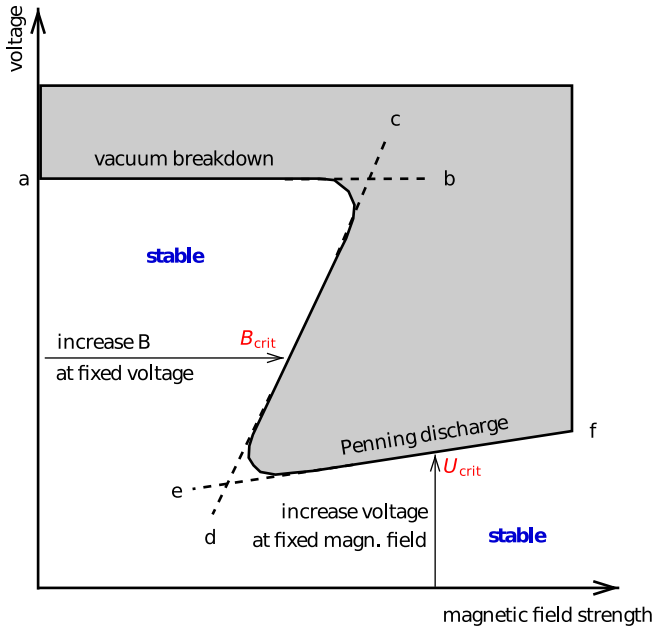


Figure 4.3.: Correlation between magnetic and electric field in a Penning discharge. The figure shows the dependency of the type of discharge on the abundant fields. The type of discharge also depends on the *order* in which the field strength are raised: If the electric field is raised fast, a vacuum breakdown can occur, while in the other case a Penning discharge is favored. At KATRIN, the magnets are only ramped very slowly due to the large amount of stored energy ($E_B = 1/2 LI^2$). Therefore, the magnetic field is increased before the electric field, and a Penning discharge is more likely to happen. There is also a stable domain shown in the figure, which corresponds to circumstances where the fields are not high enough to produce a self-sustaining discharge. Figure taken from [Val09].

3. The third important effect that leads to background is the *Penning discharge*. This type of discharge requires strong magnetic and electric fields, and is the most abundant discharge mechanism in the KATRIN experiment, if not suppressed successfully. Because of the strong magnetic fields, particles will follow the magnetic field lines in a spiral motion around the field line ("gyration"). This results in a drastically increased path length of the electrons that are stored in a Penning trap, compared to the length when following a field line directly. As the increased path length can be described as an effectively higher pressure, the scattering probability and thus the Townsend coefficient mentioned before is increased, too. The Penning discharge process strongly depends on the pressure of the residual gas.

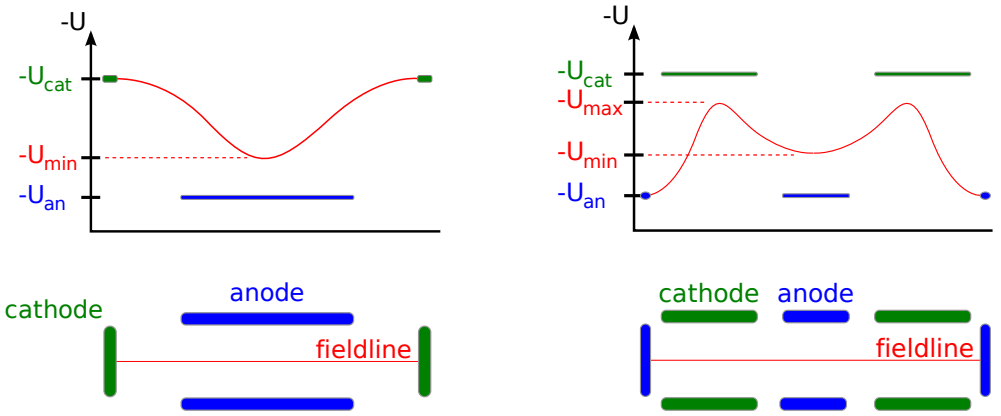
   Because of the effectively increased scattering probability, the Penning discharge can trigger Townsend discharges, which are normally suppressed at KATRIN as shown previously. It should be noted that if the storage time of the particles is limited in some way, the total path length is independent of the gyration (depending only on the velocity), and the total scattering probability stays the same.

   In addition, secondary electrons are trapped as well, and can create additional charged

particles. If the storage time is long enough this process will amplify itself avalanche-like, but only if the energy of the stored electrons is large enough to ionize residual atoms.

This minimal ionization energy is $E_{\text{ion}} = 15.4\,\text{eV}$ for H_2 molecules in the very good vacuum of the KATRIN experiment, with H_2 being the most abundant species in the residual gas (and $E_{ion}(H_2) = 15.4\,\text{eV}$). Therefore Penning traps with a depth below $U_{trap} = 15\,\text{eV}$ will not lead to Penning discharges.

4.2.2. Different types of Penning traps at KATRIN



(a) Cathode-to-cathode type **(b)** Vacuum-to-vacuum type

Figure 4.4.: Different types of Penning traps common in the KATRIN setup. Note that these are quite different than the "classic" Penning trap shown in figure 4.1, and have different properties. Both figures taken from [Zac09].

The classic Penning trap mentioned at the beginning of this section is an idealized case that is not met in the true experimental setup of KATRIN. Instead, there are mainly two different types of traps with different properties (see figure 4.4). The electron storage and discharge mechanisms of the trap will be discussed in this section for these two trap types.

Penning traps can develop in any region where high electric and magnetic field meet. As electrons are guided by the magnetic field lines, they mainly follow these lines in a cyclotron motion. If the electric potential exhibits a minimum along such a field line (as seen in the upper section of figure 4.4), a Penning trap occurs. A passing electron can get trapped in this region if it looses some energy along the way, e. g. due to synchrotron radiation or scattering processes. The trap depth describes the potential wall that a trapped electron has to overcome. All considerations in this section can also be applied to positive trapped particles; in this case the definitions of anode and cathode have to be exchanged.[1]

The two common trap types are shown in figure 4.4 schematically:

(a) **Cathode-to-cathode traps** have a potential that increases along the more positive anode in the middle. The trap depth in this case is $\Delta U = |U_{cat} - U_{min}|$. Traps of this type are

[1] It is however important to consider that ions are not bound to the magnetic field lines effectively, as their masses are much larger than for electrons.

fed mainly by electrons that enter directly from the cathode surfaces, as the magnetic field lines pass right through them. They can be stored in the trap by loosing energy inside the trap region.

(b) At **vacuum-to-vacuum traps** the potential walls are not located on an electrode surface, but inside the vacuum instead. The trap depth in this case is $\Delta U = |U_{max} - U_{min}|$. The magnetic field line does not hit any electrode surface, and thus electrons can only be trapped if they provide enough kinetic energy to first pass the potential wall and then loose energy inside the trap.

Of course, combinations of these simple trap types exist, e. g. vacuum-to-cathode traps. In such a case, the trap depth is defined by the lower potential wall, as an electron would leave the trap here if its energy is high enough. These combinations are especially likely to occur within regions with many different electrodes and/or strongly bent magnetic field lines.

The next section gives details about the Penning trap that exists between the two spectrometers of the KATRIN experiment. Apart from a discussion of its properties and how it contributes to unwanted background, previous investigations of this specific trap are shortly discussed together with the proposed countermeasures.

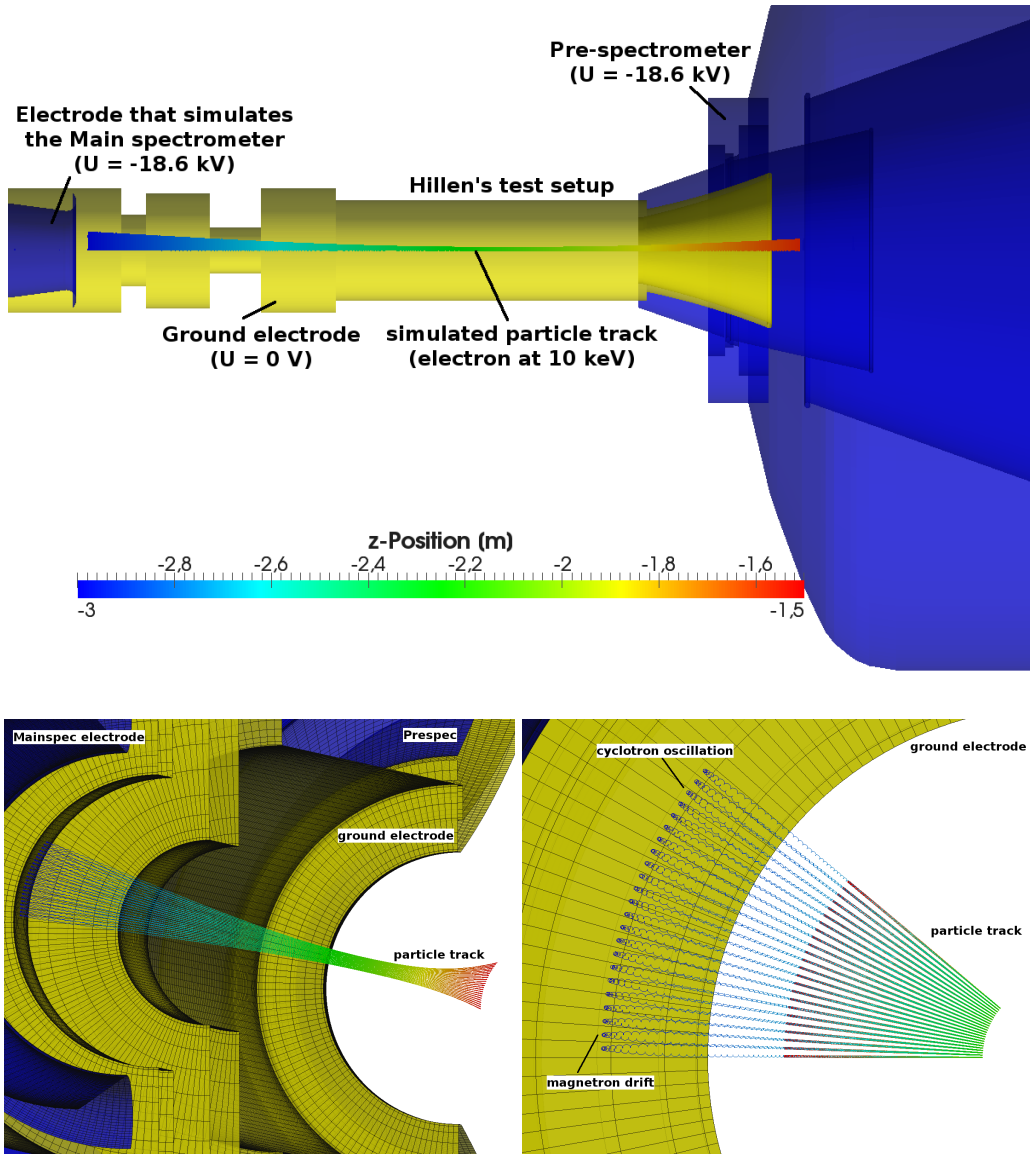


Figure 4.5.: Example trajectory of a stored electron. The three pictures show the flight path of an electron that is stored in the Penning trap at the test setup described in [Hil11]. The first plot shows the full test setup with the stored electron inside. The electrode geometries are visible here, too, and colored by electric potential. It becomes clear how the two cathodes with a potential of about $-18.6\,\mathrm{kV}$ each confine the electron to a region along the trap axis. The magnetic field created by the pre-spectrometer solenoid at $z = -2.15\,\mathrm{m}$ further confines to electron to the shown trajectory, i. e. the flux tube which the electron is following is shaped by the solenoid. The bottom two pictures show further details of the trajectory. Especially in the bottom right picture the magnetron drift can be seen, i. e. the electron slowly rotates around the trap axis. This and the axial oscillation is overlayer with the much faster cyclotron motion, which is also visible in the picture. Note: The visualization techniques used to create these graphics will be explained in the next chapter along with the simulation software.

4.3. The Penning trap between the spectrometers

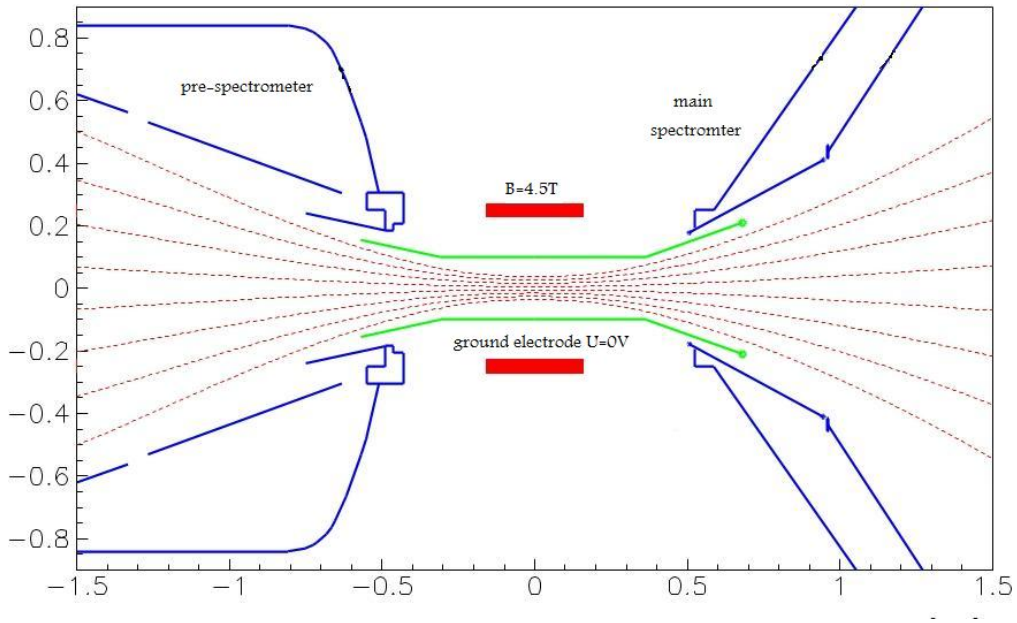


Figure 4.6.: The Penning trap between the spectrometers. The trap is located mainly inside the section covered by the ground-electrode, and results from the potential well created by the two spectrometers at $U = -18.6\,\mathrm{kV}$ and the magnetic field of the solenoid. Figure taken from [Hil11].

In this diploma thesis, the intrinsic Penning trap at the interface between pre-spectrometer and main spectrometer was investigated. This Penning trap is shown schematically in figure 4.6. It has a length of roughly 1.5 m and a depth of 18 kV. With the magnetic flux tube of $191\,\mathrm{T\,cm^2}$ at the center of the solenoid, the trap radius at this position is 3.7 cm. These values allow to estimate the three oscillation frequencies of stored electrons (see section 4.1): [Hil11]

$$\omega_+ \approx 1 \times 10^{11}\,\mathrm{s}^{-1} \tag{4.14}$$

$$\omega_- \approx 1 \times 10^{5}\,\mathrm{s}^{-1} \tag{4.15}$$

$$\omega_z \approx 5 \times 10^{7}\,\mathrm{s}^{-1}\,. \tag{4.16}$$

Electrons stored in the trap will loose energy due to synchrotron radiation and scattering processes. The energy loss by synchrotron radiation can be described by

$$\Delta E = E_0 \cdot e^{-\Gamma t}\,. \tag{4.17}$$

with an estimated value of $\Gamma = -8\,\mathrm{s}^{-1}$ at the center of the solenoid, i. e. at $B = 4.5\,\mathrm{T}$. Therefore, an electron with a kinetic energy as large as 18 keV will drop below the threshold of 15.4 eV for residual gas ionization after less than one second. The exact synchrotron energy loss depends of course on both the magnetic field (which is not constant throughout the trap volume) and the electron's polar angle with respect to the magnetic field lines. Furthermore this estimation only applies to single electrons, and other electrons can enter the trap in the meantime [Hil11].

For the case of scattering, two processes have to be distinguished: Elastic scattering can transform longitudinal kinetic energy into transversal energy and vice versa. It can therefore provide an additional cooling mechanism, since electrons with high transversal energies loose more energy through synchrotron radiation. This effectively reduces the overall kinetic energy of the electrons. Inelastic scattering can produce secondary particles, i. e. electrons and ions, which can then lead to the creation of background:

- Secondary electrons are guided by the magnetic field lines and confined to the trap, but they can produce more secondary particles due to inelastic scattering of their own, and thus lead to an avalanche effect that creates more and more charged particles. Additionally, a large number of stored electrons make up a space charge in the trap volume, which might then influence other electrons passing through. Since the trap volume is overlapping with the flux tube – i. e. electrons relevant for measurement will pass through on their way to the detector – this might indeed pose severe problems if such a space charge is allowed to build up.
- Ions are attracted by the negative potentials of the two spectrometers and are thus not bound by the Penning trap. They can therefore either hit an electrode, and create secondary electrons on impact, or travel into the spectrometers. The spectrometers themselves are strong Penning traps for positive ions, and thus the Penning trap between the spectrometers could serve as an efficient filling mechanism. This should of course be avoided.
- Apart from charged particles, photons can also be created by the de-excitation of residual gas molecules after reactions with the electrons. Although they are not guided by electric or magnetic fields, they can release electrons from the electrode surfaces by photo-emission, if their energy is high enough.

In conclusion, electrons stored in the Penning trap between the spectrometers can lead to various problems, and a mechanism to remove these electrons would prove very beneficial for the KATRIN experiment. A couple of different ways to do so were investigated in the past, and will be shortly reviewed in the next section.

4.3.1. Previous investigations of the Penning trap

Because of the severe problems the Penning trap between the spectrometers could cause, several studies on this topic were conducted in the past. These include efforts to understand the trap mechanisms, as well as examinations of proposed methods to remove trapped particles. Some of these previous studies will be shortly discussed in this section, followed by an overview of the proposed countermeasures.

Simulations

A first examination of the trap was done by KATHRIN ESSIG in her diploma thesis [Ess04]. In her work, she investigated different cooling processes of stored electrons that could prevent trap discharges, as these would only be possible for electrons with a kinetic energy above 15 eV, as stated in the previous section. It was shown that the cooling by synchrotron radiation does

not provide a way for the electrons to loose their kinetic energy efficiently, and thus would not prevent discharges. Another cooling process, the scattering on H_2 molecules from the residual gas, was then investigated by computer simulations. She found that even with this additional cooling, the electrons would achieve large storage times in the order of 1000 s with the used simulation settings.

A method to actively remove stored electrons was also investigated, namely the application of an electric dipole field on the pre-spectrometer electrodes. This dipole field would result in an $\boldsymbol{E} \times \boldsymbol{B}$ drift of the electrons that would disturb their trajectories and eventually force them against one of the electrode surfaces, thus removing these electrons from the trap. This method was proven to successfully remove electrons stored *inside* the pre-spectrometer, but It was also shown that low-energy electrons stored between the spectrometers would not be removed by a dipole potential even as high as $U_{dip} = 10\,\mathrm{kV}$. Even stronger dipole fields are, on the other hand, not possible without initiating strong discharges in the pre-spectrometer.

These studies made clear that the Penning trap at the interface of the two spectrometers would indeed pose a problem, as trapped electrons would generate a huge number of secondary particles due to their long storage times, and an active removal of the electrons would not be possible by applying a electric dipole field.

Experimental studies

In her dissertation in 2009, KATHRIN VALERIUS further investigated the background noise created by the Penning trap [Val09]. With the pre-spectrometer preliminary installed at the KATRIN site in Karlsruhe, first experimental studies were possible, which enabled a deeper understanding of trap mechanisms.

K. Valerius also proposed a selection of methods to counter-act the background created by the trap. Apart from general design optimizations and improved surface quality, which are either not applicable in this case or would not prevent the trap-induced background as such, the active removal of stored electrons was stressed as the only way to effectively reduce the background noise. The dipole drift that was investigated by K. Essig before would not remove all stored electrons. Therefore an alternative method – the interception of these electrons either by a stationary pin ("electron catcher") or by a flexible wire ("wire scanner") located in the trap region – was demonstrated by K. Valerius. She could prove that due to the fairly high magnetron oscillation frequency[2], such an interception would indeed provide an efficient method to remove stored electrons. While the stationary pin as the disadvantage of shadowing some part of the detector, the wire scanner can be kept outside the flux tube for most of the time, only sweeping trough the trap volume at specific intervals.

Another investigation of the Penning trap was done in the dissertation of BJÖRN HILLEN [Hil11]. Here the previous studies were continued on a setup slightly different from the one used before. Most notably, K. Valerius' studies were done with a backplate electrode in the experimental setup, which created a cathode-to-vacuum trap. Both studies employed the pre-spectrometer as used by KATRIN with a 4.5 T solenoid to create the magnetic field, and a vacuum pressure of 1×10^{-9} mbar. Hillen used a special electrode that electrically simulates the main spectrometer,

[2]The time for a full magnetron motion of stored electrons is in the order of tens to hundreds of microseconds.

and creates a vacuum-to-vacuum trap condition. A 64 px detector with an energy threshold of 10 keV was used for the measurements. B. Hillen's setup consisted of an extension to the pre-spectrometer as shown in figure 4.7.

In his studies, he also tested a third method to empty the trap: An external magnetic field, that changes the shape of the flux tube, would force some of the trapped electrons against the ground electrode, and thus remove them from the trap (see next section). This method was not intended to be used at the final interface between the two spectrometers due to very limited space and high field strengths of the main spectrometer entry magnet, but nonetheless provided interesting results and served as a test case for possible use at the main spectrometer. In the test setup, the air coil was located further away from the solenoid, where its field strength was sufficient to compensate the field of the pre-spectrometer magnet.

The present diploma thesis is a continuation of the studies done by B. Hillen. The goal was to further investigate and extend the experimental results by computer simulation. The simulation results will be discussed in detail in chapter 5.

4.3.2. Proposed methods to remove trapped particles

In the different studies mentioned above, mainly three methods on how to remove electrons from the trap were proposed. In this section, these methods will be discussed quickly, together with a short review of the results published in the dissertation of B. Hillen.

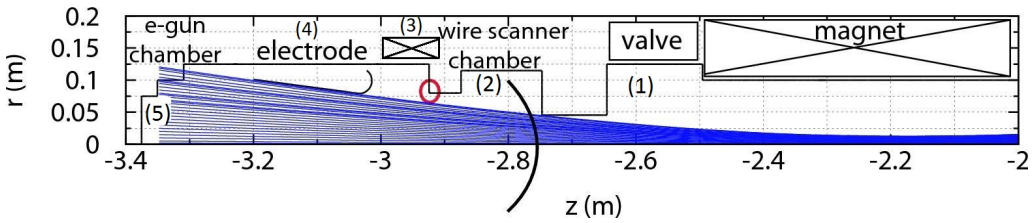


Figure 4.7.: The test setup used by Hillen. The diagram shows the position of the stationary pin (marked with a red dot), the wire scanner (2), and the pulsed coil (3); along with the pre-spectrometer solenoid. The flux tube is drawn in blue as series of magnetic field lines. The main spectrometer simulation electrode (4) is also shown here. The electron gun (5) shown at the left side was not used in the studies. The pre-spectrometer vessel was connected to the right side of the shown setup. Figure taken from [Hil11].

Electron catcher

The simplest approach to remove electrons from the Penning trap is a solid wire – or "electron catcher" – that is mounted on the ground electrode in a way that it reaches into the flux tube. Electrons that hit this wire will then be simply removed from the trap. This approach has the major advantage that it is very easy in design, compared to other methods, and is very effective. The one main disadvantage is that this stationary pin will also be employed during measurement, and therefore will shadow at least some of the 148 detector pixels. This is of course inconvenient, as it has a negative impact on the achievable statistics.

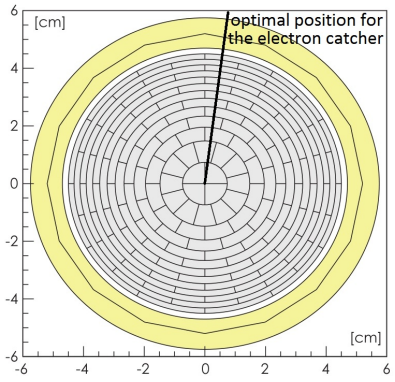
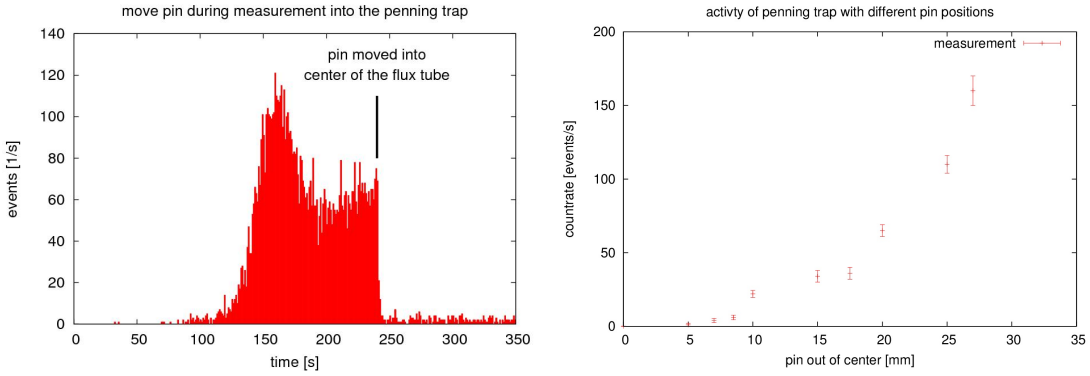


Figure 4.8: The proposed electron catcher. If employed in this form, it would shadow off at least 13 detector pixels, even at an optimal position like shown in the diagram. Figure taken from [Hil11].

The proposed electron catcher has a diameter of 2 mm. At the test setup, it could be moved about 10 cm into the center of the flux tube. Hillen's experimental results show that the pin works very efficiently at reducing the background rate (figure 4.9a), but only if it is moved completely into the flux tube (figure 4.9b). This would make sense if electrons stored at the center of the Penning trap were not able to reach the electron catcher, and therefore are not removed in case the pin is at an offset position. Indeed this is one of the aspects that was further investigated in the present diploma thesis. Of course, this leaves the conclusion that the pin will shadow 13 detector pixels, if fully moved into the flux tube to suppress the trap completely.



(a) Background with the electron catcher in place. **(b)** Background dependency on the wire offset.

Figure 4.9.: Background reduction with the electron catcher. Plot 4.9a shows the background rate measured at the detector in the range of a few seconds. Once the wire is moved into the center of the flux tube, the background rate drastically decreases. The other plot 4.9b shows the dependency of the background rate on the wire offset, i. e. the distance of its tip to the center of the flux tube. Obviously the electron catcher only works efficiently if it is fully moved into the center, and there seems to be a quadratic dependency on the offset. Both figures taken from [Hil11].

Wire scanner

The wire scanner works very similar to the electron catcher, but it avoids its main complication: The wire scanner consists of a flexible wire loop that is connected to a power supply, and is held in a fixed position outside of the flux tube during measurement. By applying a current, it will move into the flux tube by the Lorentz force. Electrons hitting the wire will be removed from the trap, like in the case of the electron catcher. By keeping the wire out of the flux tube, no detector pixels are shadowed by this method.

There are various modes by which the wire scanner can operate, e. g. rectangular or sinusoidal modulations, or a constant current through the wire. The experiments done by Valerius showed that the sinuous mode provided the best results, and this mode was therefore further investigated by Hillen. The main thing to consider here is the motion of stored electrons in the trap. When the wire scanner sweeps through the trap volume, its speed should be small enough for electrons to hit the wire loop eventually. Therefore, exact knowledge of the oscillation frequencies of stored electrons provide an advantage. The present diploma thesis contains a couple of studies on this topic, i. e. to determine the frequency dependencies on various parameters of the stored electrons.

Hillen's experiments could indeed show that the wire scanner is able to efficiently remove electrons from the trap, and thus can successfully prevent Penning discharges. It is therefore the preferred method that is likely to be implemented at the final KATRIN setup. However, the mechanical long-term stability needs to be confirmed before employing this method, as the wire scanner will likely have to work for several ten thousand cycles during the measurement phase, without an easy possibility to replace it.

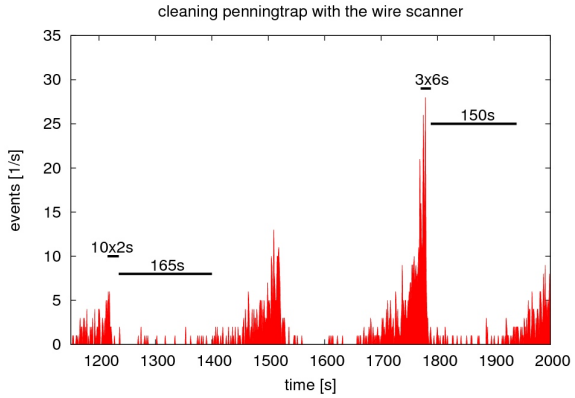


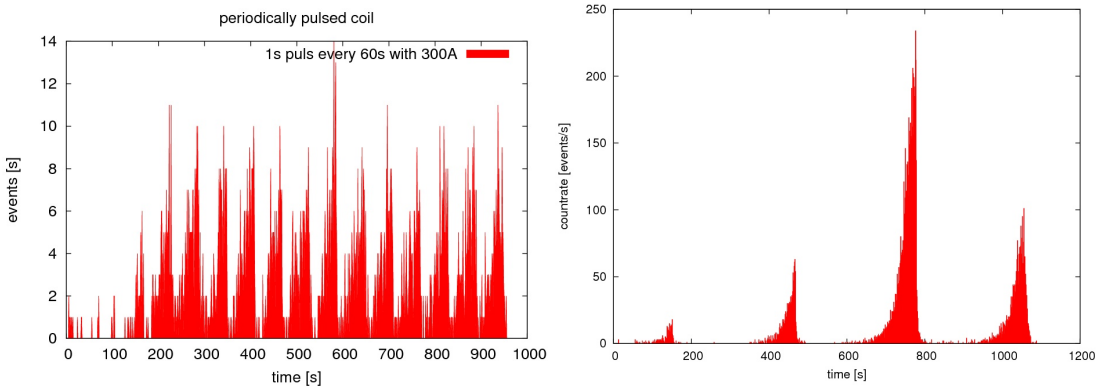
Figure 4.10.: Background reduction with the wire scanner. The plot shows the development of background rate while the wire scanner is employed. In this measurement, the wire loop swept through the flux tube at the given intervals. The cleaning efficiency does not depend on the exact intervals that are used, but on the effective time the wire loop was inside the flux tube. The background rate stays down after the application of the wire scanner for several minutes. Figure taken from [Hil11].

Pulsed coil

The last method to clean the Penning trap that was investigated by Hillen is a magnetic air coil around a part of the trap volume. This coil then can compensate a part of the magnetic field. By this method, a stored electron will loose its magnetic guiding if the coil is ramped up, and will eventually hit the ground electrode. With this consideration, electrons that are stored near the center might not be removed directly by decreasing the magnetic field with the coil.

However, Hillen's studies showed that with the pulsed coil it is indeed possible to remove most of the stored electrons, but an interesting effect was discovered: The length of the ramping phase and the strength of the compensating magnetic field only had a very low impact on the trap-cleaning efficiency. Pulsing the coil repeatedly lead only to a short reduction of background (see figure 4.11a). However, applying multiple pulses in rapid succession drastically increases the coil efficiency, and the background rate stays down for a fairly long time (figure 4.11b). This effect is currently not fully understood, and could be related to an additional $E \times B$ drift, created by the electric field that is induced from the changing magnetic field (see also section 6.4.1). A large part of this diploma thesis was therefore allocated to examine this possible drift and its net effects.

Another explanation for the increased effectiveness of multiple pulses could be that the pulsed coil only "cuts out" a limited region of the stored electrons' phase space (e. g. only electrons on outer trajectories). It could be possible that the multiple ramping of the magnetic field leads to a mixing of this phase-space, allowing electrons to move to other trajectories. These will then be removed by following pulses of the coil, which cuts out the same phase-space region again. A series of pulses could thus lead to very efficient cleaning of the trap with large background suppression, while a single pulse would give less good results.



(a) Background rate with short single pulses. **(b)** Background rate with multiple pulses.

Figure 4.11.: Background reduction with the pulsed coil. Plot 4.11a shows the background rate with a coil pulsed at regular intervals of 60 s, with pulse lengths of 1 s each. The background rate then goes up back to its normal rate shortly afterwards. If the coil is pulsed multiple times in a short time range, as shown in plot 4.11b, the background rate stays down for a much longer time after the pulse sequence. In the shown plot, ten pulses with a delay of 5 s in between were applied in each sequence. Compared to plot 4.11a, this mode works far more efficiently. Both figures taken from [Hil11].

5. Software

The main part of this diploma thesis deals with particle tracking simulations, and therefore various pieces of software were used throughout the course of this work. These shall be discussed in some detail in the following sections. Apart from the simulation software itself, the focus was also on improving the management of simulation geometries, and on providing a sophisticated visualization of both input geometries and output of the simulations. All of the tools were specifically designed for the KATRIN experiment, most notably the simulation software Kassiopeia itself. However, they could also be used in other contexts due to their general design. Kassiopeia and its core functions will be discussed in the following section.

The present diploma thesis presents a newly designed, script-based interface to Kassiopeia that can be used to efficiently handle simulation runs, and indeed proved helpful for the simulations in this work. The interface is aimed at helping users with the complex task of configuring Kassiopeia according to their needs, especially when a large number of similar simulations are to be conducted. The script interface is explained in section 5.1.4 on page 51.

In addition, a first "proof-of-concept" implementation of the geometry management tool Kreator was developed. Kreator is supposed to provide an interface between the geometry database and the simulation tool. As more complex simulation geometries will be needed to perform analysis of KATRIN measurements in future, the database will contain corresponding sets of geometry elements. Kreator is able to construct simulation geometries from these elements, according to the user's needs. This topic is further explained in section 5.2 on page 54.

Finally, the new visualization module of Kassiopeia is presented, including a number of output examples. The module provides fast and intuitive means to render simulation input and output, which makes it easier to design and verify simulation geometries, and to quickly analyze simulated particle tracks. This topic will be discussed in more detail in section 5.3 on page 58.

5.1. Particle simulations: Kassiopeia

Figure 5.1.: Proposed logo of the Kassiopeia software.

The main tool for particle simulations within the context of the KATRIN experiment is the software package Kassiopeia. It was developed in the KATRIN collaboration [Glu10] and contains a series of modules that simulate the various aspects of the KATRIN experiment. While some modules are designed for specific parts of the setup, the majority are versatile enough to be used in any simulation. Generally speaking, Kassiopeia is a tool that performs tracking of single charged particles (like electrons) through a static geometry, which is defined by a set of geometry input files. The tracking is done by splitting up the flight path of a particle into a discrete number of steps – possibly with adaptive step lengths – and solving the equation of motion at every step to compute the particle's momentum. The momentum is then used to compute the particle's position at the next step, and so on until the particle track ends.

Since Kassiopeia is a simulation software for static electromagnetic fields, the particle's motion is defined by the electric and magnetic fields in the geometry. These fields are calculated from the given geometric setup, or by other possible approaches (e. g. homogeneous or interpolated fields). Additionally, extensions to include dynamic fields are currently being implemented or extended. This is needed e. g. for the simulation of electron cyclotron resonance (ECR) effects [Lei12], or of pulsed magnetic coils. The latter is one of the topics in this diploma thesis, and is discussed in detail in chapter 6.

It is possible to execute multiple Kassiopeia runs in parallel and thus gain a huge benefit in computation time. This is especially important when a large number of independent simulations need to be performed, e. g. to analyze the impact of slight variations in the particle's initial properties on the particle's motion. It is also notable that Kassiopeia was not used at Institute for Nuclear Physics at WWU Münster beforehand; instead, a collection of script-based toolkits was employed. Kassiopeia provides a general, easy-to-use interface to all users, and its output files are better suited for analyzing the outcome of complex simulations. It is now being used for other simulation tasks as well, e. g. in the design of an electron gun [Zac13].

It is worth mentioning that the Kassiopeia package is currently undergoing changes with respect to the release of Kassiopeia 2.0. In this new release, Kassiopeia will be integrated into the KATRIN analysis and simulation package KASPER, making it easier to use together with other components of the package. Other major improvements include the full integration of visualization techniques (see also section 5.3), the integration of KEMField as main toolkit for field computation along with the possibility to use high-level geometry input instead of fully discretized geometries (see section 5.1.2 below), and a number of significant improvements to the tracking algorithms in general [Fur11] [Obl11] [Fur12].

5.1.1. Kassiopeia's configuration files

Kassiopeia uses a number of different configuration files for each run. Most files are used to configure one of the main modules of Kassiopeia ("toolboxes"), e. g. fields or stepping methods (compare fig. 5.2). The configuration files itself use a XML-like syntax in the version

of Kassiopeia that was used[1]. The most relevant aspects of the different files will be shortly discussed here.

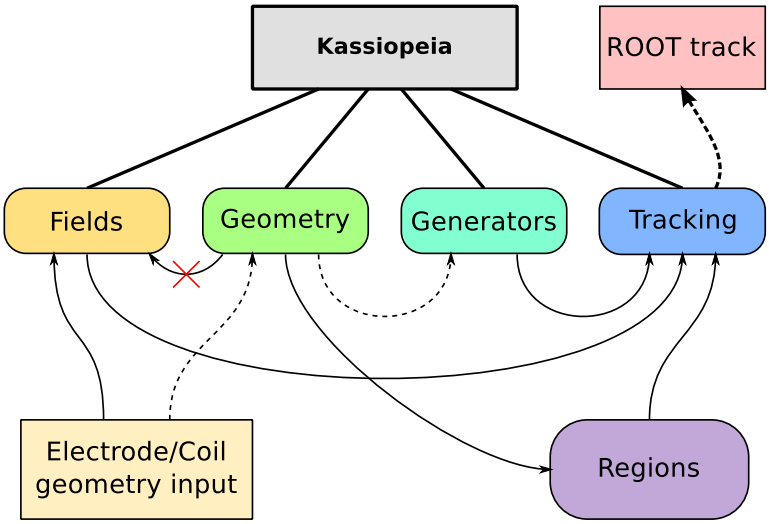


Figure 5.2.: Kassiopeia's internal structure. The software consists of a number of mostly independent modules, which together form the full particle simulation package. There are of course connections between the modules as shown in the diagram, e. g. particle generators are needed to create particles to be tracked, and also a number of fields are required to perform the tracking. In some cases it is possible to access the contents of a different module, but not required. For example, a particle generator can use a geometry (like a surface) to start particles, but also simply a fixed point. Also note that in the current version of Kassiopeia, field input and geometries are independent of each other.

GeometryConfiguration: This file defines a number of geometries which are used for the tracking process only, and are not connected to the geometries used for field computations. The geometries defined here are most useful if the simulation should employ a number of regions with different parameters, e. g. using a different stepping method inside the pre-spectrometer. Geometries can also be used for some of the particle generators or exit conditions discussed below.

GeneratorConfiguration: In the generator file, different particle generators can be defined, which are then used to start particles (create events) in the simulation. More than one generator can be used in parallel, using a weighted distribution if needed.

A particle generator consists of a four single sub-modules, which are responsible for the different parameters of a particle that are relevant at the beginning of the simulation: position, direction, energy, and time. For each sub-module a number of choices are available, e. g. using a fixed position, a disk with given radius or a geometric surface to start particles. It is also possible to use specialized particle creators, e. g. for the WGTS.

[1] Kassiopeia v1.50.00 – The configuration format will likely change in the next major release, but the general structure will be kept.

FieldConfiguration: The field configuration file contains a number of sections that configure the according field modules in Kassiopeia. The field computation in Kassiopeia is split up between electric and magnetic fields, and most modules use geometry input files to calculate static fields. It is also possible to define simple homogeneous fields or interpolate between pre-computed field values.

StepStrategyConfiguration: This file configures all the relevant parts of Kassiopeia's tracking algorithms. The main controls are the step computer and the ODE solver. For the ODE solver, a number of different Runge-Kutta implementations are available. For the step computer there are three modules available (see section 5.1.3).

The user also sets up different step size controls here; these manage the length of each step in the tracking process. Possible choices are fixed lengths, step sizes that depend on other parameters like cyclotron frequency, or dynamic step sizes that e. g. try to keep the energy error in a given range. All step size controls defined here can later be used concurrently inside a region; in case there is more than one step size effective for a step, the minimum of all computed lengths is used.

Finally, the exit conditions are also defined in this file. Exit conditions are responsible for ending a track if some specific requirement is met. Here, too, several modules are available; choices are e. g. minimum/maximum z-position, minimum kinetic energy or maximal path length of the tracked particle. If one of the effective exit conditions are triggered at a step, the track is ended at that point.

KassiopeiaConfiguration: The global configuration file is used to finally connect everything together. Here different regions can be defined, using the geometries defined in GeometryConfiguration, and the different field and stepping modules can be applied to them. If multiple electric/magnetic fields are applied in one region, their single contributions are superimposed, while multiple step strategies are used in parallel.

UserConfiguration: This file lets the user adjust global settings like the Config and Data paths (i. e. where Kassiopeia looks for other configuration files or geometry input), or the output format. Most important here is the possibility to restrict the output to tracks or events only, i. e. not writing each computed step into the output file. In cases where this is appropriate, it can help to reduce both computation time and size of the output file. It is also possible to not write every step, but chose an iteration value which only writes every n-th step to the file. This feature was in fact used for the majority of simulations done in the present diploma thesis, as the often large numbers of steps will otherwise result in file sizes in the GB range even for single tracks.

5.1.2. Field computation

Kassiopeia mainly uses fields created by static geometries for tracking particles, although it is also possible to provide additional field components, e. g. homogeneous or interpolated ones. Fields can be computed by several methods, which differ for the case of electric and magnetic fields. The currently used field code in Kassiopeia is based on a variety of programs that were developed earlier [Voe08] [Hug08]. Most of these methods have in common that they require a geometry input file from which the fields can be computed. More details are given in the following sections.

Electric fields

For electric fields, Kassiopeia uses the Elcd code developed by FERENC GLÜCK at KIT Karlsruhe [Glu04] [Glu06a]. Elcd was first implemented as stand-alone module that can compute fields from axially symmetric conical and wire electrodes (Elcd 3.2), and extended later to support arbitrary rectangular and wire electrodes (Elcd 3.3). Both variants are implemented in Kassiopeia and can be used accordingly.

The Elcd algorithm is based on a concept known as *boundary-element method* (BEM), a numeric method to compute electric potentials in a given volume. This method considers that electrodes can be seen as charged surfaces in this context, and assumes that the charge density on a given electrode surface is distributed homogeneously. This is allowed as the charge density is continuous, and the real charge density can be approximated with arbitrary accuracy. The total electric potential and field at a point in the volume can then be computed from the contributions of the single electrodes using the superposition principle. This will be explained in more detail below. An important aspect of BEM is that it needs discretized surfaces to produce accurate results. As said above, the charge density is supposed to be equal along an electrode surface. Obviously this is not true for large electrodes, where the surrounding fields lead to a inhomogeneous distribution in reality. To compensate for this, the surfaces have to be split up into smaller segments and be treated as independent electrodes (having the same electric potential, however).

In contrast to the widely used finite-elements and finite-differences methods (FEM/FDM), which divide the volume into a fine mesh, BEM is based on surfaces (or boundaries). For this reason, BEM is better suited for problems where a large volume with high discretization is considered. FEM/FDM approaches would only be possible by employing complex, adaptive meshes in this case. At the KATRIN experiment, the scale differences are very large, i. e. comparing the 10 m main spectrometer vessel to the 0.2 mm wires inside. A FEM approach would need about 10^{15} cells to produce the desired accuracy, which is not computationally feasible. On the other hand, a BEM simulation requires roughly several 10 000 elements. Therefore, BEM is the favored method for electric field calculations at KATRIN. It should be noted that the BEM approach is still limited by available memory and computing time. The memory consumption scales with $\mathcal{O}(N^2)$ ($\mathcal{O}(N \log N)$ in the best case), and computing time with even $\mathcal{O}(N^3)$ ($\mathcal{O}(N^2)$ in best the case). Therefore the number of elements can not be increased arbitrarily [Voe08][2].

[2] Another approach, the Robin Hood method, is currently being implemented into Kassiopeia. This method scales

With this concept in mind, the Elcd algorithm shall now be looked at in some detail. Each discretized electrode segment is assumed to have a constant charge density σ, and the potential contribution of a segment j at the center of another segment i is given by:

$$\Phi_j(\boldsymbol{r_i}) = \frac{1}{4\pi\varepsilon_0} \cdot \int\limits_{S_j} G(\boldsymbol{r_i},\boldsymbol{r}_S)\sigma_j \,\mathrm{d}^2\boldsymbol{r}_S \tag{5.1}$$

$$= \frac{1}{4\pi\varepsilon_0} \cdot \sigma_j \int\limits_{S_j} \mathrm{d}^2\boldsymbol{r}_S \frac{1}{|\boldsymbol{r_i} - \boldsymbol{r}_S|}\,, \tag{5.2}$$

i. e. by integrating Green's function $G(\boldsymbol{x},\boldsymbol{x}')$ over the electrode surface S_j. The charge density σ_j can be taken out of the integral because it is constant. Then the integral depends only on the electrode geometry, and (5.1) can be simplified to

$$\Phi_i j = \sigma_j C_{ij}\,. \tag{5.3}$$

Now the total potential at an element i from all other elements can be given as

$$U_i = \sum_{j=1}^{N} \Phi_{ij} = \sum_{j=1}^{N} C_{ij}\sigma_j\,. \tag{5.4}$$

As the potential U_i on each element is fixed, this equation corresponds to a set of linear equations. In the case of discretized electrodes, the potential at each segment corresponds to the potential of the "mother" electrode, as electrodes are equipotential surfaces.

After the set of equations has been solved – e. g. using the GAUSS-JORDAN algorithm – and the charge densities have been calculated, they can be used to compute the resulting potential at an arbitrary point in space. The contributions from all electrode segments are then added up to obtain the total potential and field. However, the results obtained from this method include the approximation of only taking the center of each electrode segment into account. This emphasizes why a sufficient discretization is needed to produce accurate results, as mentioned earlier. The algorithm used to compute the resulting potential depend on the type of electrode that is employed. The geometry used in the simulations contains only axially symmetric, conical electrodes. Therefore, the Elcd algorithm for this case is explained in some detail here.

Given a conical electrode segment with the endpoints (r_a,z_a) and (r_b,z_b), the potential at an arbitrary point can be computed from the previously computed charge density on the electrode by considering the full electrode to be made up from indefinitely thin ring electrodes. The potential of a such an electrode with the total charge Q is given as

$$\Phi = \frac{Q}{2\pi^2\varepsilon_0}\frac{K(k)}{S}\,. \tag{5.5}$$

with only $\mathcal{O}(N)$ in memory consumption. On the other hand, it allows for heavy use of multi-processing features, improving overall performance. See ref. [For11b] for details.

Here $K(k)$ denotes the first complete elliptic integral with

$$K(k) := \int_0^{\pi/2} \frac{\mathrm{d}\varphi}{\sqrt{1 - k^2 \sin^2 \varphi}}, \tag{5.6}$$

$$k = \frac{2\sqrt{Rr}}{S} \tag{5.7}$$

and

$$S = \sqrt{(R+r)^2 + (z-Z)^2}. \tag{5.8}$$

The potential contribution of a full conical electrode with charge density σ at a given point can then be computed by integration over an indefinite number of ring electrodes with thickness dp. With the shortcuts

$$R = r_a + (r_b - r_a) \cdot p/L, \tag{5.9}$$

$$Z = z_a + (z_b - z_a) \cdot p/L, \tag{5.10}$$

and the infinitesimal charge of a ring electrode $dQ = \sigma 2\pi R dp$ one gets

$$\Phi = \frac{\sigma}{\pi\varepsilon_0} \int_0^L dp \frac{RK(k)}{S} \tag{5.11}$$

from (5.1). L denotes the length of the line segment between the cone endpoints. The total potential from all electrodes is then calculated simply as the sum of the single contributions.

For cases where the input is symmetric, a number of simplifications can be used, e. g. only one charge density is computed for elements which are rotated around the z-axis. Because a symmetric arrangement of electrodes should lead to a uniform charge distribution on the electrodes, this doesn't affect the result, but speeds up the charge computation.

Magnetic fields

Magnetic fields are computed with the Magfield code, also initially developed by Glück [Glu06b]. Magfield uses a combination of two approaches: The first method uses elliptic integrals and has the advantage that it can compute magnetic fields even inside the coil wires. The disadvantage, on the other hand, is that these computations are very slow. Therefore, a second method using Legendre polynomials is used for points far away of the coils. In most cases the magnetic field inside a coil does not have to be known, and the latter method is used. This method is also much faster in comparison. It is also known as *zonal harmonic expansion* and will be explained shortly here. More details can be found in ref. [Glu11].

For the method using Legendre polynomials, a fixed point ($z = z_0, r = 0$) on the coil axis is chosen to compute the magnetic field. At this "source point", which lies on the symmetry axis of the coil, the magnetic field and its derivatives can be computed with high accuracy. These

coefficients are named B_n^{cen}; with B_1^{cen} equal to the magnetic field at the point, B_2^{cen} being the first derivative of the field, and so on. The equations (5.12) and (5.13) are exact; however, in computations the number of coefficients is limited, depending on the desired accuracy. These "central source coefficients" depend only on the coil geometry and current, and can be used to easily compute the magnetic field in an arbitrary point (z,r) not too far away from the source point, i. e. within a certain convergence radius. This convergence radius ρ_{cen} can be defined as the minimal distance of a source point from the coil windings (see fig. 5.3).

The components of the magnetic field at that point can then be given as

$$B_z = \sum_{n=0}^{\infty} B_n^{cen} \left(\frac{\rho}{\rho_{cen}}\right) P_n(u) \tag{5.12}$$

and

$$B_r = -s \sum_{n=0}^{\infty} \frac{B_n^{cen}}{n+1} \left(\frac{\rho}{\rho_{cen}}\right) P_n'(u) \tag{5.13}$$

with the shortcuts

$$u = \cos\vartheta = (z - z_0)/\rho\,, \tag{5.14}$$

$$s = \sin\vartheta = \sqrt{1-u^2} = 1/\rho\,. \tag{5.15}$$

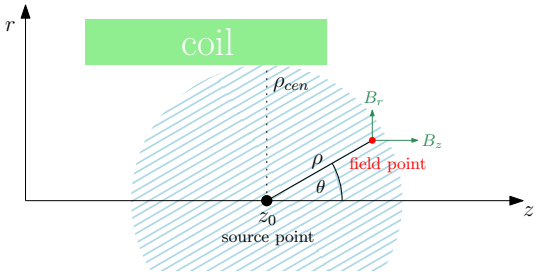


Figure 5.3.: Convergence radius of the central zonal harmonic expansion. Figure taken from [Kass].

Here $\rho = \sqrt{(z-z_0)^2 + r^2}$ is the distance of the field point (z,r) from the source point $(z_0, r_0 = 0)$, while $P_n(u)$ denotes the Legendre polynomial of order n and $P_n'(u)$ its first derivative. For multiple coils, the single contributions are summed up to give the total magnetic field.

Further developments

There has also been implemented another module for field calculations lately. This new module named *KEMField* was developed by Thomas J. Corona (University of North Carolina at Chapel Hill) and Joe Formaggio (Massachusetts Institute of Technology) and can use a different approach for computing charge densities, namely the *Robin Hood* method, in addition to the BEM concept mentioned previously. Robin Hood is a convergence algorithm that exchanges charges between electrodes, much like it is happening in reality when electrodes at different potentials are brought together. One main advantage of the Robin Hood algorithm is its memory consumption of $\mathcal{O}(N)$, where BEM scales with $\mathcal{O}(N^2)$ or $\mathcal{O}(N \log N)$. A disadvantage is that the time needed to compute charge densities is longer, and in fact depends strongly on geometric complexity. For a high number of elements, RobinHood's computation time scales with $\mathcal{O}(N^2)$. This is comparable to optimized algorithms like Gaussian elimination or Gauss-Seidel that could be employed by classic BEM methods, so the main advantage of Robin Hood is its lower memory footprint.

Furthermore, both charge density and field calculation can make use of multi-threaded CPU and GPU computing, i. e. running parts of its code in parallel on a (graphics) processor. This feature obviously provides a huge benefit in computation speed while maintaining accuracy (speedup factors of $10 to 50$ have been measured using OpenCL), and makes most sense if it is used with very highly discretized electrodes or large geometries [For11b] [Cor11] [For11a].

In addition, KEMField supports triangular electrodes, which is important especially for complex geometries, as round surfaces can't be adequately described using rectangular segments only. Furthermore, methods were implemented to automatically discretize electrodes into smaller segments, making the geometry input files much easier to handle. In the next version of Kassiopeia, it will also be possible to use even complex geometries like intersections between conical electrodes (like shown in figure 5.4).

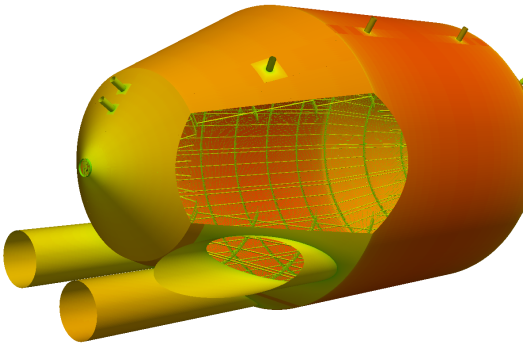


Figure 5.4.: Example of a "complex" geometry. KEMField allows to use geometries like this main spectrometer electrode structure for field computations. The shown geometry is colored by the logarithmic area of the electrode sub-elements. Figure created with ParaViewfrom data files provided by [Cor12].

However, KEMField was not used for the simulations in this diploma thesis, as only axisymmetric geometries were needed and the features provided by KEMField would not have been a huge advantage here. However, these features are likely to become more relevant for large, complex geometries in the near future, and KEMField is in fact going to replace Elcd and Magfield as standard field modules in Kassiopeia.

Geometry input files

The electrode and coil geometries are defined by input files (as shown for example on page C7), where each line corresponds to one element (or a group of elements). Elcd supports a number of different elementary electrode types, these are: conical electrodes, wires, and rectangular electrodes. The latter have the main benefit that they do not imply axial symmetry, and therefore can be used with non-symmetric geometric setups. The geometries used in this diploma thesis only used conical electrodes (and, to a lesser extent, wires), as the geometry is strictly axisymmetric.

While rectangular electrodes are already defined with full discretization – i. e. as single electrode segments, even if they correspond to a larger electrode in reality –, conical electrodes and wires allow an extra discretization parameter. This parameter is used to split up the full electrode into single segments along the z-axis. As noted above, discretization is very important when working with BEM to achieve a high accuracy. For coil geometries, however, such discretization is not necessary, and the corresponding input files define one single coil per line (see page C7 for an example).

The electrode input file used for the present diploma thesis was created using a Python[3] builder script developed by S. VÖCKING in his diploma thesis [Voe08], in combination with an already existing geometry for the pre-spectrometer that is included in Kassiopeia. The coil input files for the magnetic fields computations were created manually for the case of the pulse coil, or taken over from standard a geometry file provided by Kassiopeia for the case of the pre-spectrometer solenoid.

[3]Python is an interpreted programming language with object-oriented features. See [Py] for details.

5.1.3. Particle tracking

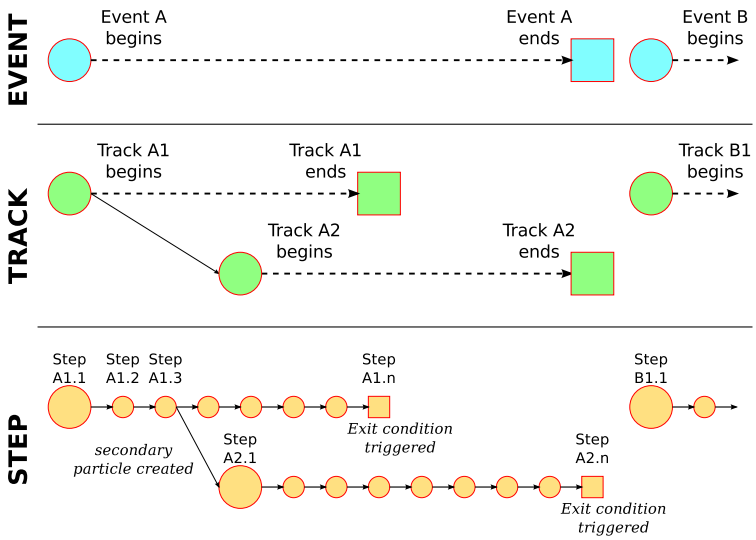


Figure 5.5.: Kassiopeia's particle tracking. The diagram shows the three "levels" of Kassiopeia's particle tracking. An **event** is the most abstract view, and contains one or more particle **tracks**. Each track is made up of a series of single **steps**, that define the particle's flight path.

The simulation software Kassiopeia allows to track charged particles through a given geometry. At least for now it does not allow to track multiple particles in parallel, therefore particles are simulated one after the other. Kassiopeia divides the simulation into three domains (compare figure 5.5):

- An **event** is the top-most domain and corresponds to a particle being created by one of the various particle generators that can be used in Kassiopeia.
- A **track** corresponds to a single particle being simulated. This also includes secondary particles that can be created by scattering processes; therefore each event can consist of one or more tracks, and each track is connected to a parent event.
- Finally, a **step** is the shortest interval along a track and corresponds to one iteration of the simulation. Each track contains a number of steps, possibly with varying time intervals, that make up a path from the beginning to the end of the track.

At each step, the motion of the tracked particle is computed. This is done by solving the chosen equation of motion by an ODE solver, e. g. the *Runge-Kutta 7/8* method[4] that is used by default. The equation takes into account the electric and magnetic fields at the current particle position, therefore these have to be computed at each step, too. It is also possible to include additional effects like synchrotron radiation or scattering processes, where the latter may result in secondary particles to be created. From the data computed at each step, the

[4]The Runge-Kutta methods were developed around 1900 by C. RUNGE and M.W. KUTTA, and are an important family to compute approximated solutions for ordinary differential equations (ODEs).

next step length is then chosen by evaluating a number of user-defined conditions. Before the simulation proceeds to the next step, the given exit conditions are checked, and the track is ended if one of them is triggered. It is also possible to use regions with different setting; in this case, it is checked at each step if a region boundary was crossed. This feature is useful if one wants to use finer step sizes in a critical region, while using larger steps in another to improve performance.

The core of each simulation is the equation of motion that is solved at each step to compute the particle's flight path. In Kassiopeia, there are currently two methods implemented: [Kass]

- The **ExactStepComputer** (ESC) works by solving the complete Lorentz equation (4.1) at every step. The equation of motion that is solved in this case is

$$\dot{\boldsymbol{x}} = \boldsymbol{v} \tag{5.16}$$

$$\ddot{\boldsymbol{x}} = q \cdot (\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \,. \tag{5.17}$$

- The **AdiabaticStepComputer** and the **AdiabaticDriftStepComputer** (ASC/ADSC) use an adiabatic approximation instead. These only compute the motion of the guiding center of the particle trajectory, i. e. skipping the cyclotron motion, and thus achieve a much higher computation performance at the cost of a loss of accuracy. Also, these step methods won't give correct results for non-adiabatic simulations. The equation of motion used in the ASC is

$$\dot{\boldsymbol{x}} = \hat{\boldsymbol{B}} \cdot v_{\parallel} \tag{5.18}$$

$$\ddot{\boldsymbol{x}}_{\parallel} = -\frac{\mu}{\gamma} \cdot (\nabla B)\hat{\boldsymbol{B}} + q\boldsymbol{E} \cdot \hat{\boldsymbol{B}} \,, \tag{5.19}$$

  where μ is the magnetic momentum (3.8) and γ is the Lorentz factor (3.1) of the electron. $\hat{\boldsymbol{B}} = \boldsymbol{B}/B$ denotes the unit vector of the magnetic field. The guiding center approximation can be derived from invariance of the magnetic moment, i. e. the energy exchange between cyclotron motion and motion of the guiding center, as shown in reference [Pic92].

The step computers can also be extended by additional modules that describe other physical aspects not covered in the step method itself, e. g. adding energy loss by synchrotron radiation or including various scattering processes.

5.1.4. Running simulations with Kassiopeia

When Kassiopeia is executed, it first tries to read in all necessary files from the current directory. In addition, Kassiopeia accesses the Data and the Config directories as fall-back option. While the config directory holds a default versions of each of the configuration files, the Data directory contains standard geometry input files or parameters for the scattering modules. It is also used as a cache directory for files created by the field modules, thus requiring read/write access for the user. By passing a specific UserConfiguration file to Kassiopeia, these directories can also be changed.

Kassiopeia's simulation output is written to a ROOT file[5], whose size can quickly increase while Kassiopeia is running. As noted above, the file size can be significantly reduced by using a step iteration value, or by excluding the single steps from the output.

The ROOT output file contains independent trees for the event, track and step level, and can be directly processed with ROOT or ROOT-based applications. The standard ROOT `TBrowser` provides an easy way to have a first look at the data and perform simple analysis tasks. A number of parameters is available at each of the tree entries. The exact contents depend on the output configuration and on the active modules that were employed. For example, the step tree normally contains basic data like particle position and time, but can also be configured to include the values of electric and magnetic fields, or the particle energies [Kass].

Run management

Handling the different configuration files of Kassiopeia can be quite difficult, especially if several simulations are to be run with slight variations of the input parameters. Additionally, managing all the configurations by hand is prone to errors.

For these reasons, a new, fully scriptable interface was written in Ruby[6] during this diploma thesis. The *Kassiopeia Run Manager* (KARMA) uses a number of wrapper classes that resemble the modules available in Kassiopeia together with several template files that correspond to the configuration files of Kassiopeia (see section 5.1.1). With KARMA it is possible to manage simulations efficiently since the whole configuration is done in the main script file, which is then executed to start simulation sub-runs with the given settings. KARMA supports starting processes in the background, thus allowing several sub-runs to be executed in parallel in an easy way[7]. The script can also submit jobs to a batch system like the one used at the MAF computing grid of the Institute for Nuclear Physics at WWU Münster. Especially the latter provides a good way to speed up simulations, as sub-runs of one simulation (or even completely different simulations) can run in parallel on up to about 200 CPU cores, which would be impossible on a single machine, even taking today's efforts in implementing multi-core CPUs into account[8].

Because the KARMA interface is fully scriptable and each sub-run is identified by an incremented number (the run id), it is very easy for the user to variate simulation parameters in the single sub-runs. For example, the user can set up geometries, fields and step strategies at the beginning of the KARMA script, and then variate the starting energy of tracked particles in each sub-run, using a simple loop-construct. This will lead to a number of sub-runs, possibly started in parallel, where each sub-run uses a different particle starting energy but otherwise shares the same settings. Because ROOT is used for the track output, it is fairly easy to later combine the sub-run results.

[5] ROOT is a very powerful, object-oriented data processing and analysis framework developed at CERN. See [ROOT] for details.

[6] Ruby is an interpreted programming language with object-oriented features, in many ways similar to Python. See [Ruby] for details.

[7] This feature is fairly useful on today's multi-processor machines, which often consist of four or more CPU cores. Therefore the computation speed can be more than double without any disadvantages in most cases.

[8] For example the *nubase* simulation workstation of the AG Weinheimer in Münster has "only" 8 CPU cores. However, it was used in addition to the computing grid for Kassiopeia simulations.

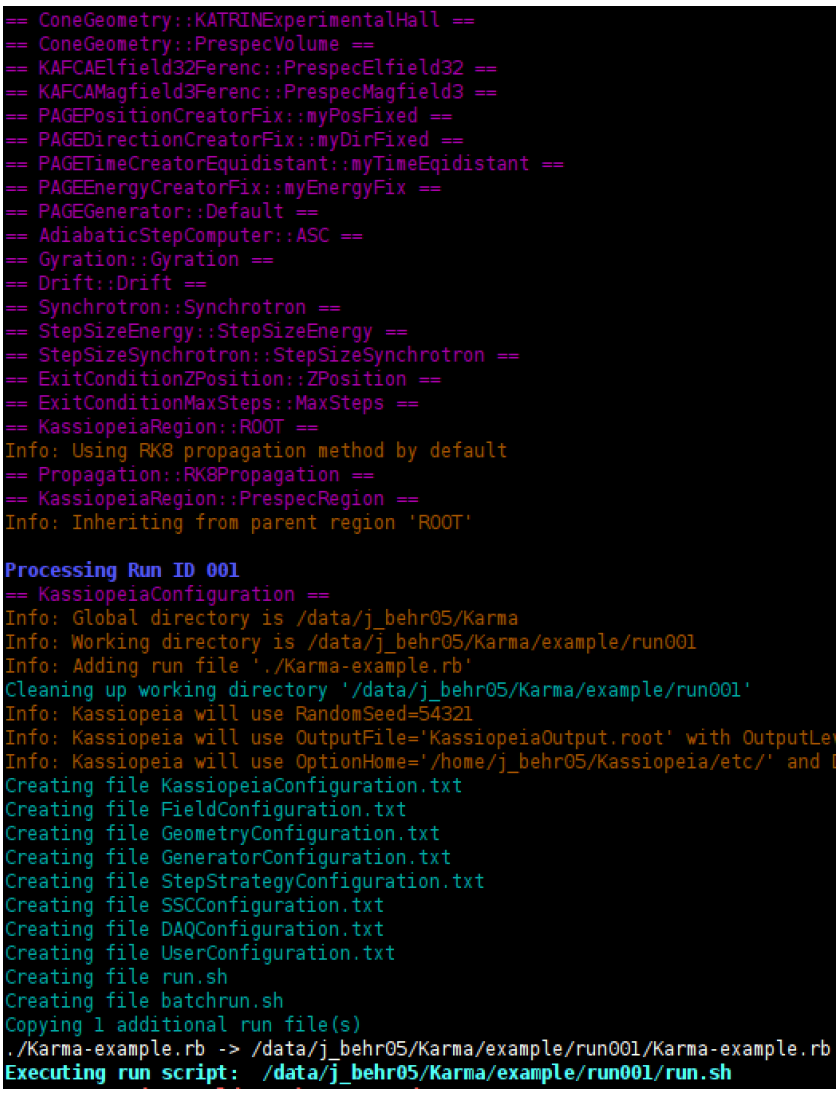


Figure 5.6.: An example of a KARMA run.

An example KARMA script file is included in appendix A. The full source code is accessible at the KATRIN Git repository[9].

[9] https://nuserv.uni-muenster.de/cgit/cgit.cgi/Karma.git/

5.2. Geometry management: Kreator

Figure 5.7.: Proposed logo of the Kreator software.

With more complex geometries like the ones used in the analysis of KATRIN measurements, it is important to have a mechanism to efficiently manage the input data used in simulation. It would not make sense to use a geometry for the whole KATRIN experiment in full detail, if one is only interested in the simulation of particles in front of the focal plane detector, for example. Therefore a general geometry database was proposed, which would contain geometrical descriptions of all parts of the experiment, but in different levels of detail. A management software could then be used to access this database, providing an interface to the end-user who is then able to choose the geometric representation according to his needs. Finally, this management tool would create a geometry input file that can be used in simulations. It would also be possible to implement the retrieval of sensor data from the ORCA database and apply this data to the geometry, too, if required[10] [Zac10] [Fur10].

To utilize this concept, a first version of a geometry management software named *Kreator* was developed in this diploma thesis. The main goals of this software are as follows:

- Provide an interface to the KATRIN geometry database for the end-user.
- Allow users to construct geometry input files for use in simulations.
- Make use of nested templates in the database (for details see below).
- Allow to save and restore files in Kreator, e. g. to make modifications to a previously configured geometry setup.
- Include elements from other parts of the database in the resulting geometry file, most notably sensor data like electric potentials.
- Employ a “timestamp” mechanism, which allows to set up simulation geometries that represent the exact state of the experiment at a given time.

The access to the database is provided by the *KaLi* library developed by SEBASTIAN VÖCKING, which allows the application to retrieve elements from the database in an easy-to-implement way.

To keep the need for duplicated database entries down to a minimum, a sophisticated method of cross-referencing and nesting templates was constructed. It consists of three kinds of database elements:

- Branches,
- Nodes,
- Templates.

[10] This concept would allow users to e. g. include current readings from the power supply into their simulations.

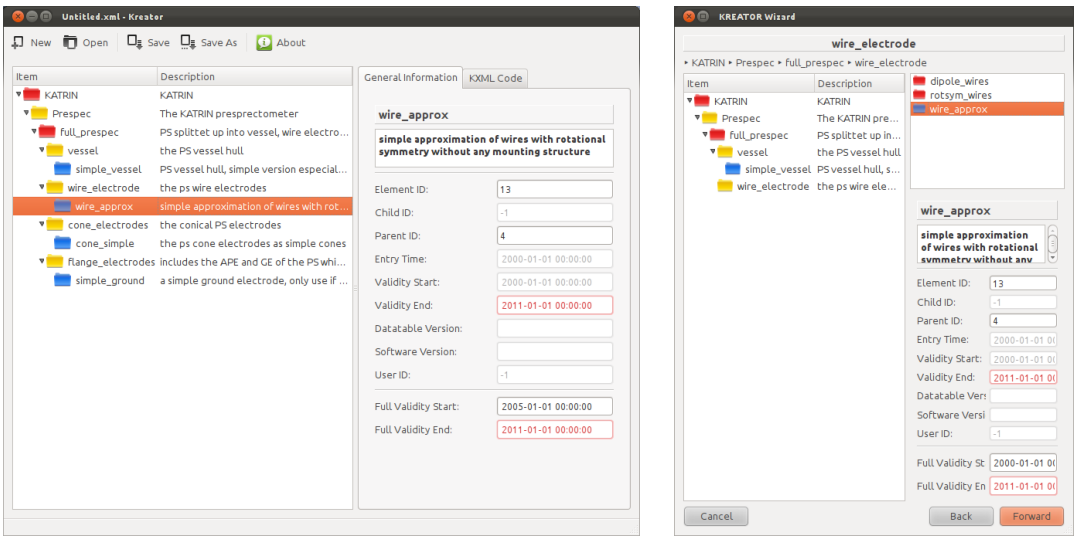
While the templates contain the final representation of a geometry to be used in a simulation, possibly together with meta-data like CAD drawings or QA reports, the other two element types are necessary to allow the referencing of database entries that is needed for KATRIN. The main difference between the two is that branches imply a selection on one of the underlying sub-elements by the user, while a node requires *all* of its sub-trees to be configured.

This concept is shown in figure 5.9. For example, a user creating a new geometry setup starts at the KATRIN branch on top (yellow), and chooses between a template with simple representation of the whole experiment (blue), or one of the two available nodes that contain a more sophisticated collection of geometries (red). These two possible collections in the given example correspond to the SDS and STS sections, respectively, and both contain a representation of the spectrometer. Since the spectrometer geometry should not be duplicated in the database, it makes sense to be able to include the same elements in different collections, like it is shown here.

As the multiple nested templates in the database can be confusing, the end-user will need a powerful interface that helps him in constructing the final geometric setup from the database. This feature is provided by the Kreator wizard (see figure 5.8b), where the user starts at the top-most element in the database – the KATRIN branch in the example from above – to create a new geometry setup, and then traverses the database tree structure downwards by choosing from available elements at each branch. If only one element is available, it is chosen automatically for convenience. At nodes, the wizard will guide the user through each sub-tree, until all elements at the node are configured. The final geometric setup will then consist of a tree with a number of branch and node elements, and with a template at each leaf, i. e. each sub-tree ends with a part of geometry that can be used for simulations. This tree can then be viewed and edited in the Kreator software, or be exported to a simulation geometry file (see figure 5.8a).

It is important to understand that Kassiopeia itself (and other, related tools) are not supposed to handle the various realizations of geometric setups in the database. These tools work on a complete, fixed setup which is constructed for the specific task. Instead, the Kreator tool will manage the conversion from the single templates in the geometry database to a simulation geometry that is usable in Kassiopeia. Note that with Kreator's own file format it is possible to reconstruct the full tree including the user's selections at each point, since the complete tree structure is saved (in addition to the single tree elements). Using XML as a data format for Kreator makes this task a lot easier, as XML inherently supports nested elements, and thus the tree structure can be embedded directly into the file. Also, XML has the advantage of being a human-readable format, which means that in principle users can edit or verify the file manually. This also applies to the geometry format that will be used in Kassiopeia 2.0, which is also based on XML. But as noted, the Kassiopeia geometry files do not contain any meta-information – i. e. from which templates the geometry was constructed – but only the pure geometric elements.

Currently the Kassiopeia software is undergoing severe changes as it is merged into the KATRIN simulation and analysis framework *Kasper*. In this process, the handling of geometry input is also changed from the old pre-discretized Elcd/Magfield format to an easier-to-handle XML format that only describes high-level geometric objects like cones or cylinders. The discretization is then supposed to happen in Kassiopeia itself, providing the ability to use different discretizations, while maintaining only one input file while also improving the handling of regions in Kassiopeia. Since a lot of work has already been done on this topic,



(a) The Kreator main program. **(b)** The Kreator wizard.

Figure 5.8.: Screenshots of the Kreator software. On the left, the interface of the main program is shown. The user can either create a new geometry structure using the Kreator wizard or open an already existing file. The main interface will then allow to reconfigure the tree from some starting point, or export a geometry file from the tree which can then be used in Kassiopeia. As currently major changes are made to Kassiopeia, including the geometry system, this export feature is not fully usable in the current version of Kreator. The right side shows the Kreator wizard, where a user can retrieve geometry elements from the KATRIN geometry database and compose a new set of geometry. The wizard takes care of properly traversing the tree structure from the database, i. e. showing branches, nodes and templates and making sure that a complete tree is constructed by the user. Note that the database-specific features are currently not fully implemented, and therefore a local, file-based database is used for testing purposes.

and the current version provides a “proof-of-concept” as it is able to construct geometries from a number of different implementations that are read from a database, it is planned to appropriately extend the software in the near future.

Another useful idea that is worth mentioning, is to include a visualization of the single geometry elements or even the full setup directly in the software, allowing users to quickly check their setup for obvious mistakes. The visualization mechanisms are already implemented in Kassiopeia, as will be seen in the next section, and therefore can be easily included in the Kreator software later on.

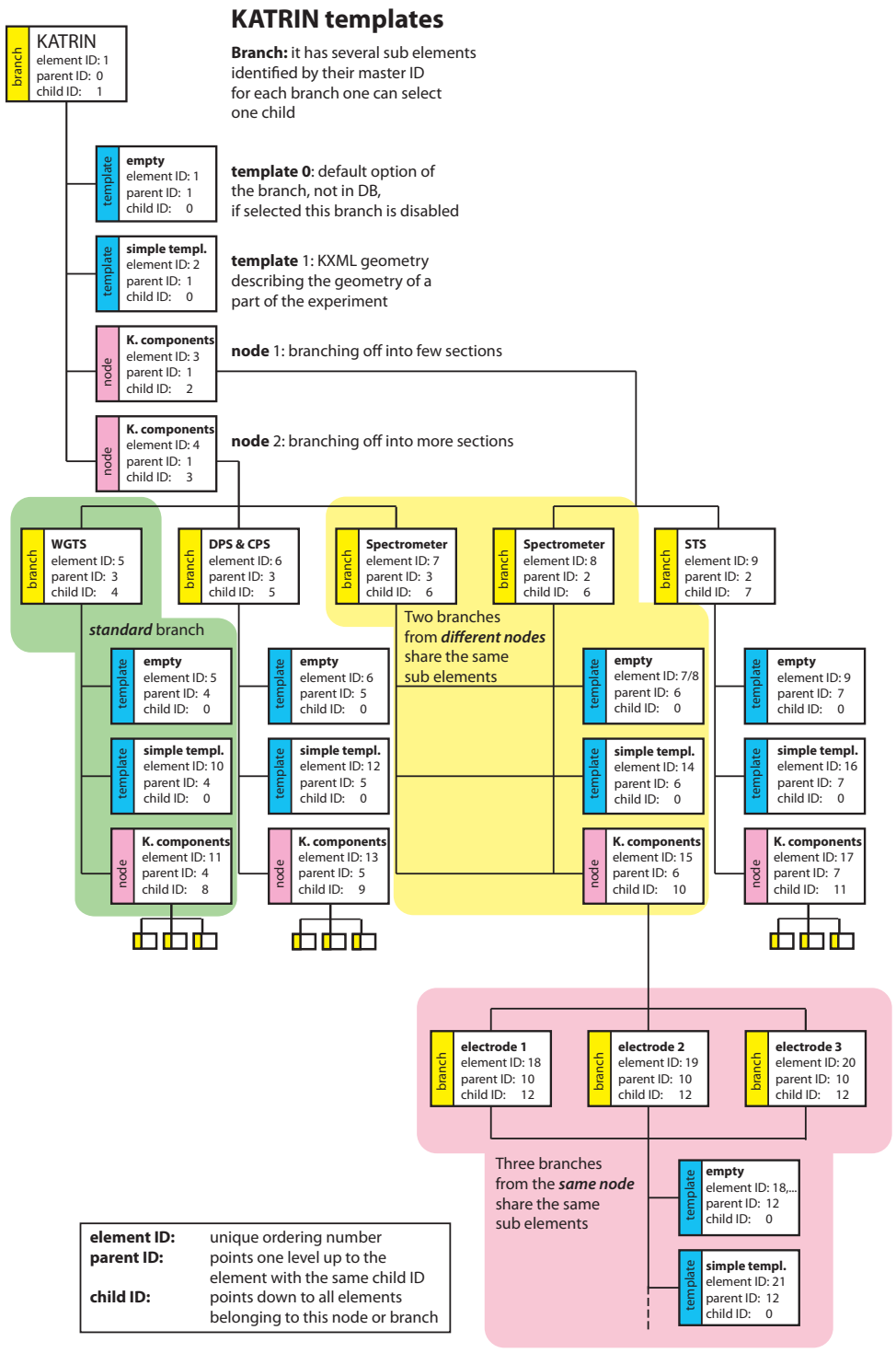


Figure 5.9.: The geometry template system used in the KATRIN database. The main elements – branches, nodes, and templates – are connected by cross-references. They can be retrieved from the database by the Kreator application to construct a geometry for Kassiopeia simulations. Figure taken from [Wol11].

5.3. Visualizations: VTK

Visualization of both input and output of simulations is getting more important as the simulations are increasing in complexity. In the case of KATRIN, even simulations of a small part of the whole experiment require a large number of input parameters. Especially the geometry used may contain several thousands of elements. A way to quickly check the geometry before starting a simulation will surely be beneficial. Also the output – one or more particle tracks with large amounts of related data – contains a large amount of information which can better be digested by using a decent visualization technique.

There are a number of software libraries available that can perform the task of visualizing all kinds of scientific data. One of these is the *Visualization Toolkit* (VTK). VTK is currently developed and distributed by the company Kitware, and is freely available due to its open-source license.[11] VTK is a good choice to visualize simulation data especially in the Kassiopeia context for the following reasons:

Figure 5.10.: Logo of the VTK toolkit.

- VTK is written in C++, as is Kassiopeia. This allows for an easy and straight-forward integration.
- Usage in Kassiopeia is allowed without restrictions because VTK is open-source.
- A large number of visualization mechanisms are supported. As Kassiopeia uses a number of very different types of data, being able to handle this data in an easy way is very convenient. Data types which can be visualized include polygonal data from the geometry, track data (points and/or polygonal lines), scalar and vector fields. VTK has different methods to create visualizations for these, e. g. scalar coloring or contour lines.
- Additonally, the independently developed front-end software *ParaView* is available. It also uses VTK, and is able to read in VTK-compatible files[12] and create visualizations from these (see below). Thus it is not necessary to write an own sophisticated software which is able to perform all the actions for working with the visualization data. Instead, ParaView provides an interface to almost all of the required functions.

For these reasons, a new visualization module was implemented in Kassiopeia. The main goal was to provide visualization methods that can be used directly within Kassiopeia, e. g. to show the simulation geometry or the user-defined regions before starting a simulation, or to even show the tracking process while the simulation is running. However, due to the ongoing changes in Kassiopeia with respect to the new major release, two stand-alone applications were implemented first. These applications can be executed independently of Kassiopeia, and are meant to provide the necessary visualization tasks, and serve as “proof-of-concept” implementation. The applications will be explained in further detail in the next section.

[11] See [VTK] for further information.

[12] It is also possible to import other widely-used formats like STL or SDML, which are supported by many CAD applications.

5.3.1. VTK and Kassiopeia

The Kassiopeia visualization module was intended to be used directly inside Kassiopeia, and thus an own user interface needed to be implemented. This simple interface allows to show geometries and tracks, and to use color maps on the elements to visualize data like electric potential or kinetic energy (in the case of track output). As ParaView already provides a very good interface that is easy to use, this simple interface was not extended very much, but instead methods were implemented to create corresponding VTK data files, which can then be read directly into ParaView. The internal user interface is still kept, however, as it provides a nice way to have a quick look at the results, without needing to create and import files to ParaView.

As noted above, the current implementation of the visualization module contains two stand-alone applications, *TestGeometryVTK* and *TestFieldmapVTK*. These can be used either before or after a simulation run to visualize geometries and tracks, or compute three-dimensional field maps, respectively. These features are useful to verify both geometry input and track output of Kassiopeia, as well as to provide electric and magnetic field maps that can be analyzed further. Both applications allow to create VTK data files that can be imported in ParaView later on to provide additional possibilities regarding visualization. Some output examples are shown in figure 5.11.

The first program, TestGeometryVTK, is able to read in different files used or created by Kassiopeia and visualize their contents directly in an output window. This applies to both Elcd and Magfield input files, as well as Kassiopeia's ROOT output tracks. With the integrated user interface, the user can move around in the scene using the mouse and the keyboard for control. This provides a very simple and efficient way to check the geometry files and track output. In addition, all relevant data from the input files can be visualized in the application, using a color map on the objects. This is also shown in figure 5.11, where the electrodes and coils are colored by electric potential and current density, respectively[13]. With the addition of KEMField to Kassiopeia, the TestGeometryVTK program can now also compute charge densities on electrode surfaces directly, and visualize these together with the geometry itself. This provides a good way to check the discretization level of the input geometry, as a too coarse discretization would result in an unequal distribution of computed charge densities. Therefore, large deviations in charge density along a discretized electrode would imply an insufficient level of discretization.

The application can also read an output file created by Kassiopeia and visualize the corresponding particle trajectories, possibly with color-coded output of the related parameters like particle energies. Obviously this only works when the output of single steps was enabled in Kassiopeia's configuration (compare section 5.1.4). For large particle tracks, it is possible to specify an iteration value to only show every n-th step of the track. All track data can also be exported into a VTK file, which can then be shown in ParaView. With ParaView it is even possible to create animations of a simulated particle.

The second program, TestFieldmapVTK, uses VTK's data types and KEMField to compute the values of electric and magnetic fields in a rectangular, three-dimensional grid volume. Since the KEMField implementation from Kassiopeia is used, it can also make use of KEMField's new

[13] As both of the pre-spectrometer solenoid have the same current density, both coils have the same color applied.

features regarding multi-processor support, or use the RobinHood algorithm instead of BEM (compare section 5.1.2). The application has no user interface; it just creates VTK files for ParaView, as it would be difficult to provide an own user interface for all the VTK features.

A number of input parameters can be passed to the application to define the volume where the fields are to be computed. These parameters are given as the center coordinates, the three-dimensional dimensions of the grid, and the grid discretization level (i. e. the number of grid points in each direction). Additionally, a "power" factor can be given, which results in a non-uniform grid spacing. This feature is useful in situations where a higher resolution is required either at the center or at the boundaries of the grid.

Usage examples

Both programs follow the GNU standard for passing command-line arguments to the application, and a command overview can be displayed by passing the `-help` option. Some of the most important features will be explained here shortly.

```
TestGeometryVTK --elcd32 aufbau+prespec.el32 --mag3 ps_magnet.mag3 --mirror
```

This command will reproduce the output shown in figure 5.11a. It shows both the electrode and coil geometry given by the two input files, using their respective formats (Elcd 3.2 and Magfield 3). As the electrode geometry is symmetric to the $z = 0$ plane, only half of it is defined in the file; however, the field algorithm will apply this symmetry when computing the electric field. To produce a correct visualization, the additional `-mirror` parameter is passed to the program.

```
TestGeometryVTK --track KassiopeiaOutput.root --polyline
```

This is an example of the Kassiopeia output visualization. The file `KassiopeiaOutput.root` is a ROOT file created by Kassiopeia, which contains a particle track. The track will be visualized as a series of line segments by passing the `-polyline` parameter (otherwise, the single steps would be shown as points).

```
TestGeometryVTK --elcd32 aufbau+prespec.el32 --kemfield --kmethod rh
```

Since the new KEMField code is included in Kassiopeia and therefore in the test programs, it is possible to pass the geometry input to KEMField and then visualize the used geometry directly (i. e. in the exact form which KEMField uses for its calculations) . This will also enable the display of charge densities and total charges on the electrode surfaces, which could be useful to check the discretization settings (compare section 5.1.2). The `-kmethod` parameter is used here to select the RobinHood algorithm instead of the Gauss-Jordan algorithm that is used by default to compute charge densities.

```
TestGeometryVTK --elcd32 aufbau+prespec.el32 --mag3 ps_magnet.mag3
        --track KassiopeiaOutput.root --interval 100 --novtp
```

This example shows a combination of the different features. An electrode and coil geometry is visualized together with a Kassiopeia track. With the given `-interval` parameter only every 100th step is read from the track, and memory consumption is reduced significantly. Additionally, the `-novtp` parameter skips the writing of any VTK output files, which is most useful if one wants to only take a quick look at the simulation results, without the possibility to further analyze the data later in ParaView.

```
TestFieldmapVTK --mag3 ps_magnet.mag3 --size 1.0,3.0 --disc 100
```

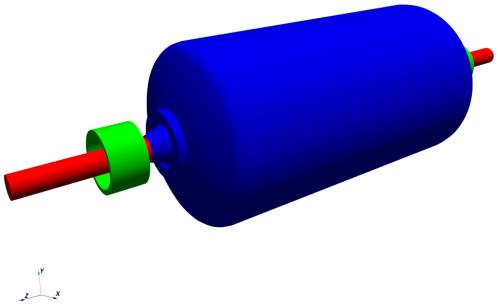
This command shows a simple run of the TestFieldmapVTK program. It will compute the magnetic field as created by the pre-spectrometer solenoid defined in the given Magfield 3 file in a grid of 100x100x100 points. The `-size` parameter is used to set the grid dimensions to $1\,\mathrm{m} \times 1\,\mathrm{m} \times 3\,\mathrm{m}$, centered at (0,0,0).

```
TestFieldmapVTK --mag3 ps_magnet.mag3 --center 0,-2.15 --size 1.0 --disc 100
```

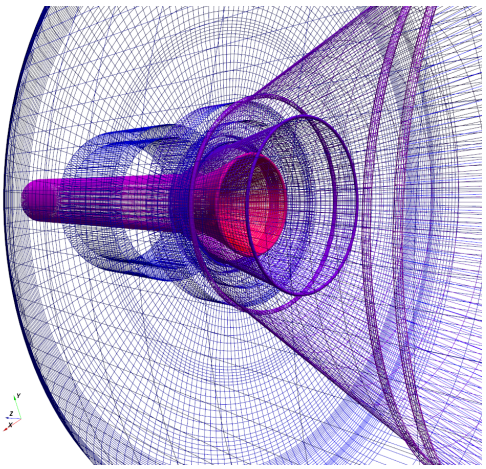
This example is similar to the last one, but uses a grid of $1\,\mathrm{m} \times 1\,\mathrm{m} \times 1\,\mathrm{m}$ that is centered at $(0,0,-2.15)$. Note that `-size`, `-center` and `-disc` allow to define one, two or three numbers. When only two numbers are given, the first one refers to both x and y coordinates, while the second correspond to the z coordinate.

```
TestFieldmapVTK --elcd32 aufbau+prespec.el32 --mag3 ps_magnet.mag3
        --center 0,-2.15 --size 0.5,1.5 --disc 100,300 --kmethod rh
```

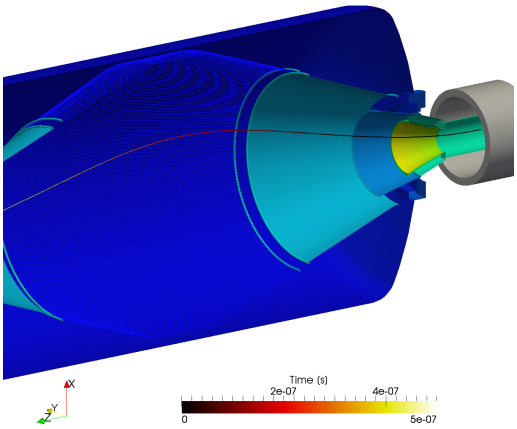
Here a combined calculation of electric and magnetic fields is shown. In this case, the application will write additional data to the VTK output file (e. g. the $E \times B$ drift), which is disabled when computing only either electric or magnetic fields. This example also uses a asymmetric grid of $100 \times 100 \times 300$ points, and uses the Robin Hood methods to compute charge densities prior to the calculation of the electric field at each grid point. When imported into ParaView (see below), output like the one shown in figure 5.11d is possible.



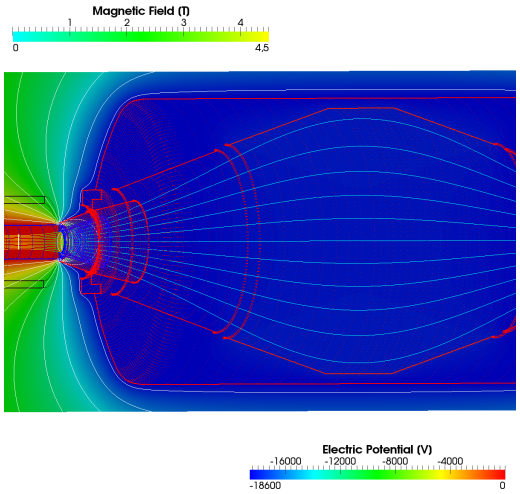
(a) Elcd and Magfield geometries shown in TestGeometryVTK.



(b) Detailed view of Elcd ann Magfield geometries in TestGeometryVTK.



(c) Simulated Kassiopeia particle track processed with TestGeometryVTK.



(d) Electric and magnetic field computed with TestFieldmapVTK.

Figure 5.11.: Example of visualization output. Shown here is a simulation geometry containing both electrodes and magnetic coils (left), together with a field map that was computed using this geometry with the Kassiopeia field algorithms (right). The field map contains both magnetic field lines and contour lines of the electric potential. The upper two TestGeometryVTK screenshots were generated directly from the application, while the lower screenshots of TestGeometryVTK and TestFieldmapVTK output are shown in ParaView. Some of the advanced features of ParaView can also be seen: The left image shows a cut through the geometry to make the track visible. The right image shows magnetic field lines from the $\boldsymbol{B}$ vector field and electric contour lines from the U scalar field, which were both computed with TestFieldmapVTK previously.

5.3.2. ParaView

Figure 5.12.: Logo of the ParaView software.

Although the VTK library provides an easy-to-use programming interface with the ability to create own end-user programs, it is far more easy to just use existing software for this purpose. The software *ParaView*, which is also developed by Kitware in addition to VTK, provides a sophisticated user interface with many features[14] and is ideal for the visualization tasks in the KATRIN experiment. Because it is based on VTK itself, it is very easy to use it together with files created by TestGeometryVTK/TestFieldmapVTK, or with other VTK applications yet to be developed. The user interface of the TestGeometryVTK application was therefore not extended very much after its first implementation; it is only supposed to provide a quick look on the output.

Further analysis of simulation results can then be done by importing the VTK files (geometry, tracks, fields) into ParaView. Some of the possibilities available here are to make cross-cuts through the objects or cut away parts (e. g. creating a “window” in an electrode surface to provide a view on a particle track), quickly compute field lines from a previously computed field map, or draw two- or three-dimensional contour lines. A threshold filter can be used to cut away parts of an object based on the values it contains, which is useful especially with particle tracks as this feature allows to show only a limited range of steps in a track. An animation feature is included, too, and allows to e. g. animate a particle track, and then export the animation as a movie. Additionally, ParaView can be extended fairly easily by providing user-defined filters written in Python or C++.

[14] See [PV] for a list of features and an introduction to the software.

6. Simulations of the Penning trap

The core of this diploma thesis are particle simulations which are a continuation of the experimental studies of B. HILLEN. The simulations were conducted to better understand the Penning trap between the two KATRIN spectrometers, and to investigate different methods to empty the trap. These methods were an electron catcher, a flexible wire loop, and a magnetic compensation coil; all of which would force stored electrons out of the flux tube and thus remove them from the trap [Hil11]. These and other previous studies were shortly reviewed in chapter 4.

The conducted simulations use a geometry which resembles the setup employed by B. Hillen: It consists of the KATRIN pre-spectrometer together with a special construction that mimics the interface to the main spectrometer. One important part of the setup is a special electrode that electrically simulates the main spectrometer; allowing to create trap conditions that are comparable with the real conditions at the final setup of the KATRIN experiment (see also section 4.3.1).

Kassiopeia simulations of stored electrons were conducted to further investigate the characteristics of the trap, and to provide a better understanding of the proposed trap-emptying methods. The first section in this chapter discusses the general behavior of stored electrons in the trap, and includes investigations of the trap dimensions and the frequencies of the electrons' motion. Thereafter, the simulations regarding two of the emptying methods – the electron catcher and the pulsed compensation coil – are discussed. The wire loop was not further investigated, as its effect is comparable to that of the electron catcher.

As mentioned earlier, Kassiopeia is able to track single particles sequentially[1]. Most of the simulations discussed here used only single electron tracks with variations in the starting

[1] However, the computing grid at the Institute for Nuclear Physics at WWU Münster allowed to execute more than one simulation in parallel, thus gaining a huge benefit in computation time.

parameters[2], e. g. for the investigation of the trap dimensions. For the simulations of the electron catcher, Monte-Carlo simulations with several thousands of randomly-started electrons were used. However, the tracking of single electrons makes it impossible to include interactions between the stored electrons in the simulations. On the other hand, tracking thousands of electrons simultaneously with possible interactions would require both a large amount of memory and computation time. Simulations of this kind could serve as an enhancement to the results that are presented here. Therefore, we are not able to study the Penning discharge itself, but the ejection mechanism of stored electrons.

It should be noted that the simulations make it possible to examine the exact trajectories of single stored particles, depending on a variety of parameters. This would not be possible by experiment alone, therefore the simulations should be seen as an enhancement to the experimental studies presented by B. Hillen to understand the functional details of the ejection methods applied there.

6.1. Simulation geometry

As mentioned in section 5.1.2, Kassiopeia uses a set of geometry input files to compute the electric and magnetic fields, which are needed for particle tracking. Hence it was necessary to create a set of suitable geometry files including the pre-spectrometer itself, B. Hillen's setup with the electrode to simulated the main spectrometer and the magnets. The electrode and magnet geometries are shown in the appendix page C1.

A validation of the electric fields in the trap region showed that the discretization was indeed sufficient[3], and the combinations of electrodes create a vacuum-to-vacuum Penning trap similar to the one shown in figure 4.4.

In figure 6.2, the electric potential and magnetic field in the trap region is shown. The $-18\,\mathrm{kV}$ potentials of the main spectrometer electrode and the pre-spectrometer vessel create a potential well, and with the strong magnetic field of the pre-spectrometer solenoid this leads to a very deep Penning trap that is located in the inner flux tube region. As explained previously, the stored electrons can induce background by various means, e. g. by the creation of secondary particles. The figure also shows a sequence of magnetic field lines with different radii. The stored electrons will basically follow the magnetic field lines, and their motion can be described as a superposition of three oscillations as explained in section 4.1.

The electric potential, magnetic field and resulting $E \times B$ drift (which is important for the magnetron motion of stored particles, see section 4.1) is shown as two-dimensional colormaps in figure 6.3. The first image shows that the electric potential is at ground level within large parts of the test setup, and the potential quickly drops down to $-18\,\mathrm{kV}$ near the pre-spectrometer vessel and the main spectrometer electrode. The potential gradients (and thus the electric

[2] The most relevant starting parameters are the kinetic energy E_0, the polar angle to the magnetic field line ϑ_0, and the axial and radial position (z_0, r_0); this was concluded from the first simulations that were performed prior to the investigations discussed here.

[3] An insufficient discretization could be detected e. g. by looking for oscillations of the electric potential along the main axis. Another test for sufficient discretization is to increase the discretization level, and see if there are changes to the electric potential. With the used setup, the deviations are in the mV range and therefore much smaller than the trap depth and the investigated particle energies.

fields) are very high in these regions, especially near the main spectrometer electrode where the potential drops by several kV over a length of a few mm. The second image shows the magnetic field of the pre-spectrometer solenoid, which is distributed very inhomogeneously along the test setup. The third image shows the (logarithmic) magnitude of the $E \times B$ drift speed. It is apparent that the drift speed becomes very high near the boundaries between the electrode segments, which is especially the case near the main spectrometer electrode where high electric fields are present. There is also a radial dependency of the drift visible, i. e. the drift speed depends on the distance to the z-axis. This will become relevant later, when discussing the magnetron drift of stored electrons.

Figure 6.1 shows a visualization of the test setup together with a simulated electron track. It can be seen how the electron stays confined to a specific region within the ground electrode (red), as it is reflected by the large negative potentials of the pre-spectrometer vessel and main spectrometer electrode (blue). The figure depicts the general motion of electrons which are trapped between the spectrometers.

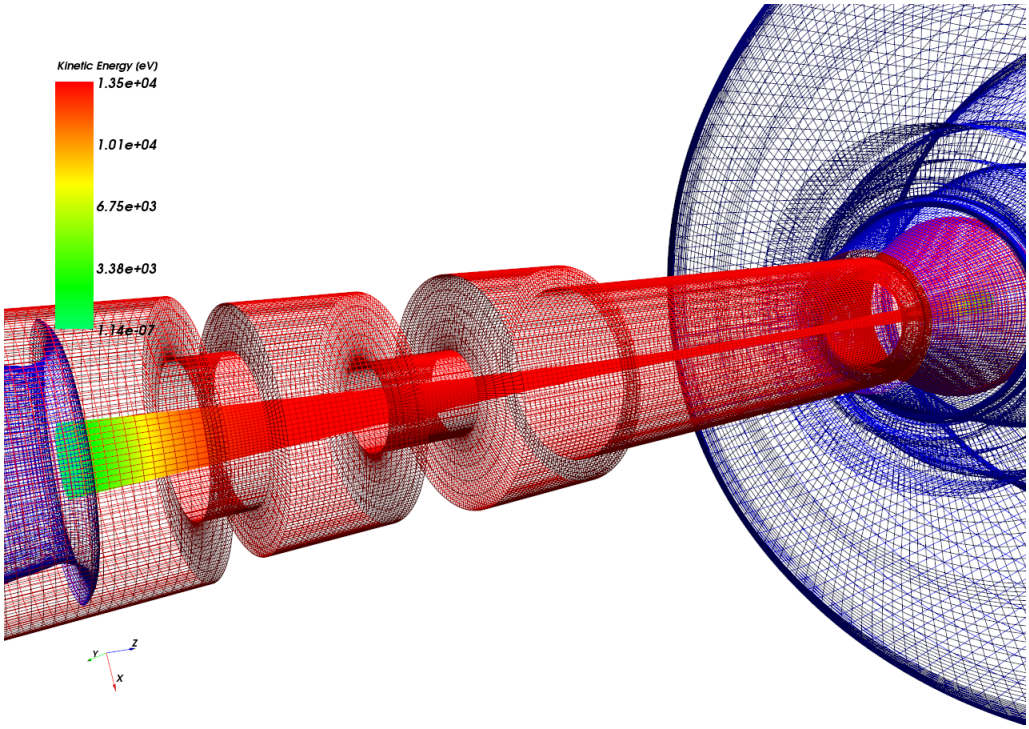


Figure 6.1.: Example simulation output. The image was generated with the *TestGeometryVTK* program (see section 5.3.1) and shows the track of a stored electron with 13.5 keV initial kinetic energy. The particle track is colored by kinetic energy, and it can be seen how the particle is slowed down at the trap endpoints and its kinetic energy is reduced. Also shown is the electrode geometry which was used in the simulation, colored by electric potential (red: 0 V, blue: −18.6 kV).

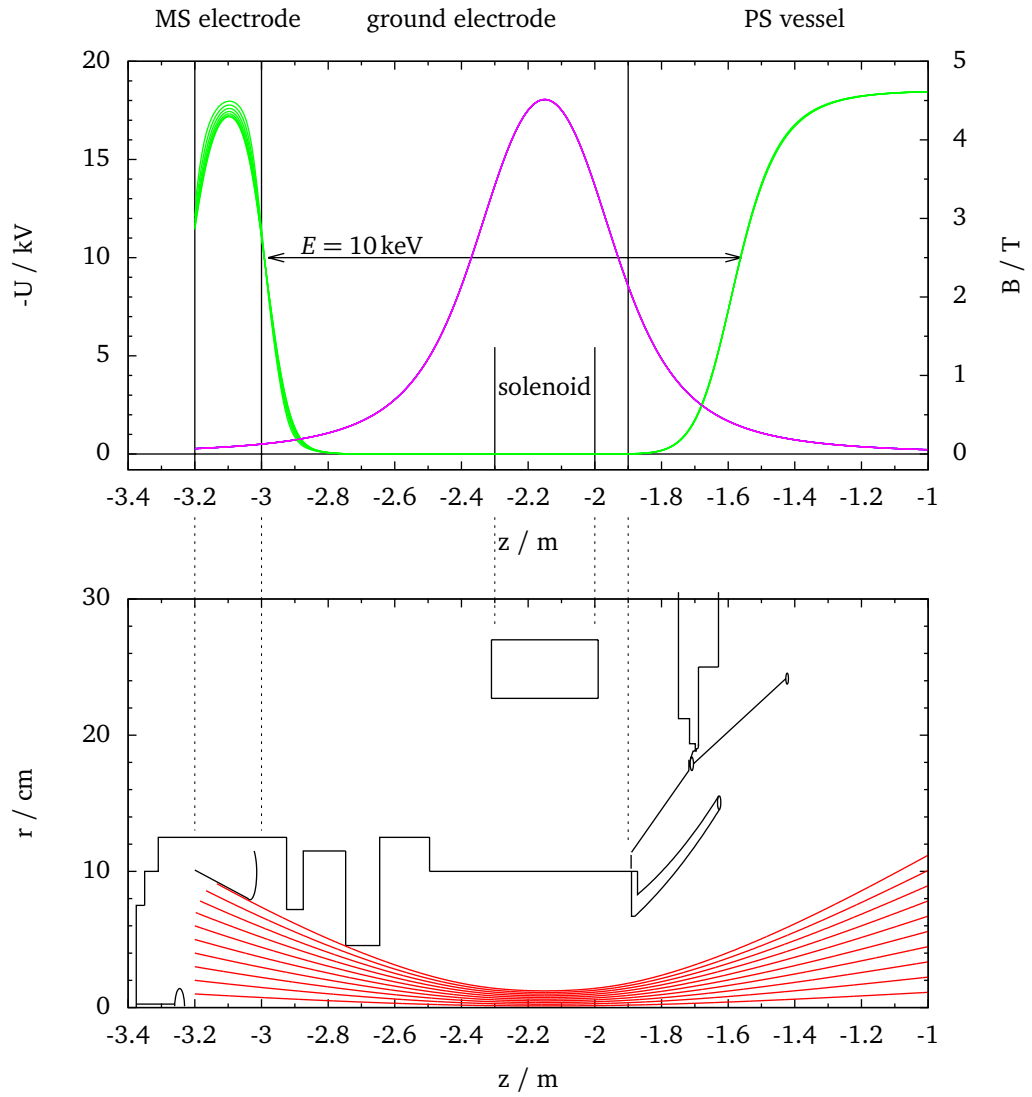


Figure 6.2.: Electric potential and magnetic field in the trap region. In the top figure, the electric potential and magnetic field along the magnetic field lines in the trap region are shown. The potential well that creates the trap can be seen clearly; it is generated by the pre-spectrometer vessel and main spectrometer electrode, which are both at negative potential. The pre-spectrometer solenoid is located in between. A radial dependency of the electric potential can be seen near the main spectrometer electrode, where the potential is slightly different for the five computed field lines. The bottom figure shows a set of magnetic field lines in the trap region. Note that this does not corresponds to the full 191 T cm^2 flux tube of the KATRIN experiment, since a part of the flux tube is obstructed by the electrodes in this setup. The magnetic field of the pre-spectrometer solenoid leads to a compression of the flux tube towards its center at -2.15 m. The stored electrons will approximately follow the shown field lines, depending on their initial radius. Also shown in this plot is the electrode and magnet geometry.

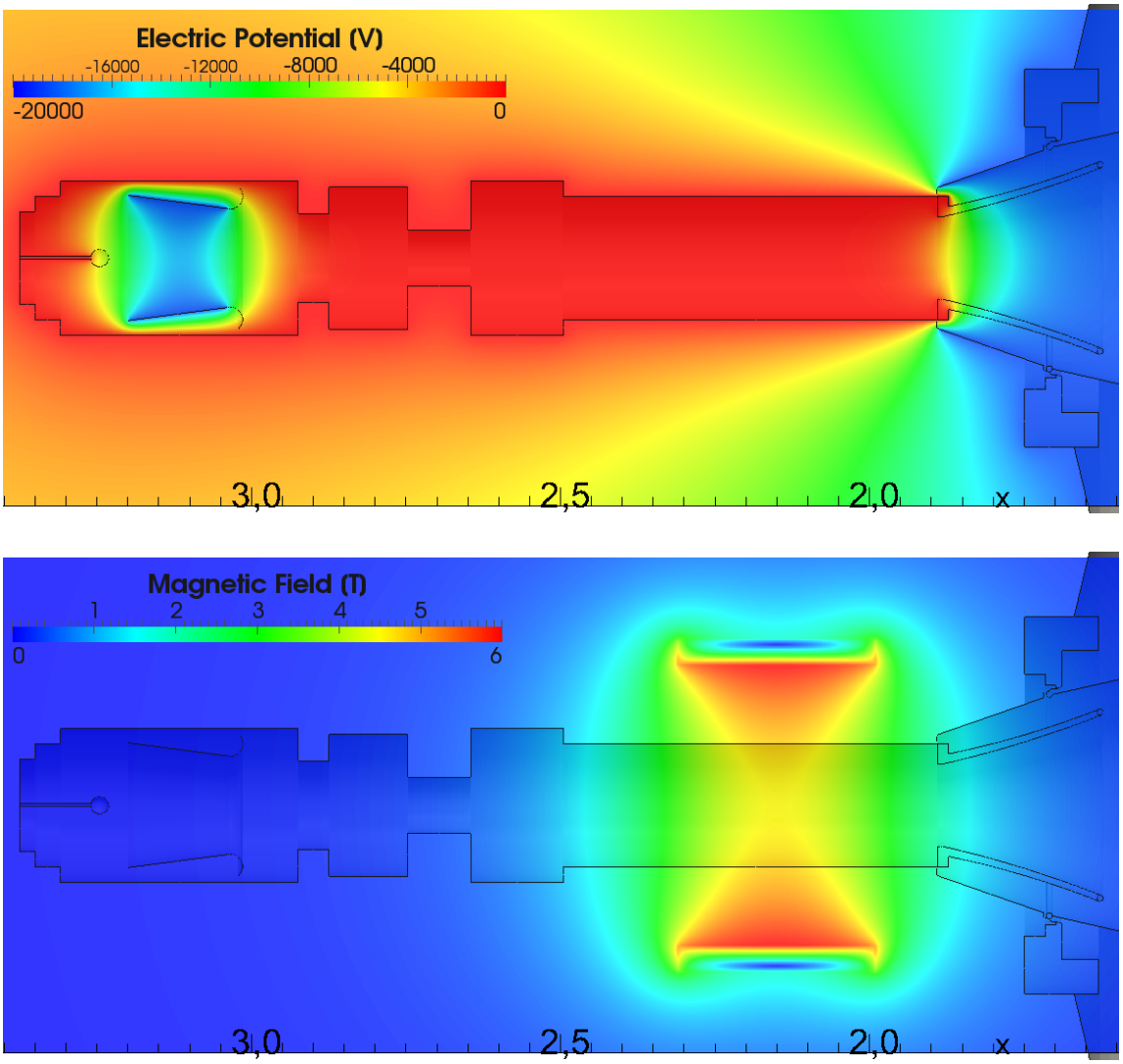


Figure 6.3.: Electric potential and magnetic field in the trap region. The images show two-dimensional maps of the two quantities, using a total of one million field points each. The maps were generated with the *TestFieldmapVTK* program (section 5.3.1). The first figure shows the electric potentials resulting from the pre-spectrometer vessel, the ground electrode and the main spectrometer electrode; the second shows the magnetic field created by the pre-spectrometer solenoid.

6.2. Trap properties

A better understanding of the general trap properties and the motion of electrons stored within is important for the investigations of the trap-emptying methods. Therefore several simulations were conducted to study the behavior of stored electrons. The examined properties include the form and dimensions of the trap, especially in axial direction, and the motion of trapped electrons. As will be shown in this section, the inhomogeneous magnetic field and the complexity of the electrodes result in trap conditions that can no longer be described by the simple relations given in section 4.1.

6.2.1. Trap dimensions

It is both very interesting and helpful for future investigations to know the exact dimensions of the trap region. This also allows to determine in which parts of the interface between the spectrometers electrons will be trapped. Following the explanations given in chapter 4, the trap dimensions should have a dependency on the electrons' parameters, most notably the kinetic energy, because the electrons will be electrically reflected by the potential walls, and the point of reflection depends on the available energy. Simulations show that this is indeed the case, and the trap length changes drastically with the electron's kinetic energy. Additionally, the effect of magnetic mirroring has a strong impact on the trap length for high starting angles with respect to the direction of the magnetic field. These effects will be further discussed below.

On the other hand, the effective radial dimension of the trap (i. e. its diameter) is mainly given by the size of the flux tube in the adiabatic case. This follows from the fact that stored electrons stay confined to the magnetic field lines[4] and therefore the initial radial position of an electron defines its trajectory. The general form of the electrons' trajectories correspond to the magnetic field lines shown in figure 6.2, i. e. an electron started at a low radius will stay on an inner trajectory, and electrons started at higher radii will stay on an outer trajectory. As the magnetic shielding prevents electrons from radioactive decays in the electrodes from entering the trap, it can be said that the flux tube defines the effective trap volume in radial direction. The trap itself can be filled by electrons from the Tritium beta-decay, which e. g. are reflected at the main spectrometer's potential and then loose energy by scattering (compare also chapter 4).

The flux tube at KATRIN has a size of $191\,\mathrm{T\,cm^2}$, and due to conservation of magnetic flux its diameter changes with the magnetic field. At the center of the pre-spectrometer solenoid, the flux tube diameter is at its minimum value of 7.35 cm, and the flux tube widens towards the trap endpoints. In the case of B. Hillen's setup, the trap diameter is also limited by the diameter of surrounding electrodes, as stored electrons on outer trajectories will hit one of the electrodes and will thus be removed.

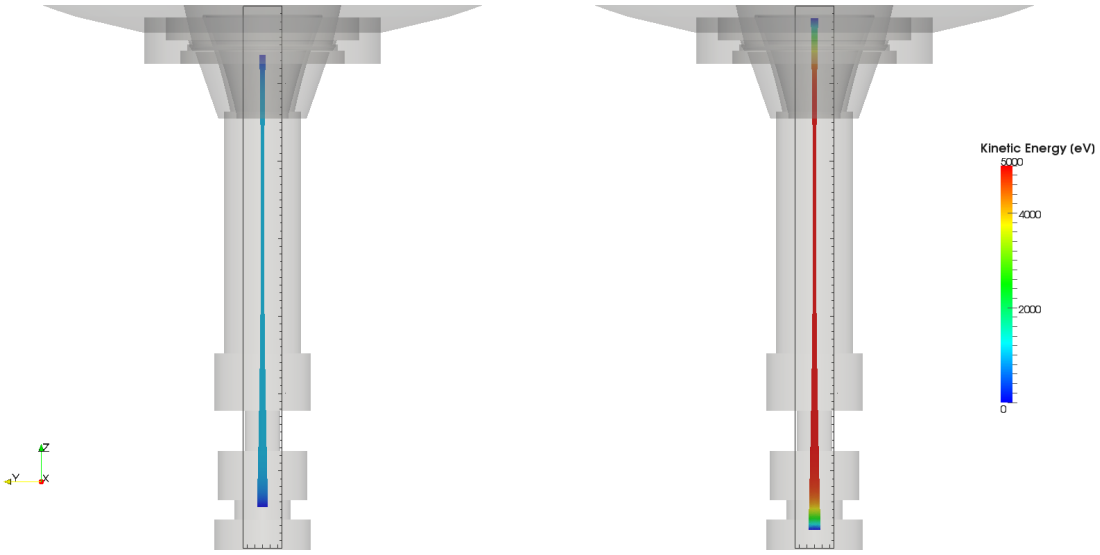
It should be noted that the simulations in this section were done without taking any energy losses into account (e. g. synchrotron radiation or scattering processes), which allows a closer investigation of the dependencies on the starting parameters. Furthermore, the performed simulations lasted only fractions of milli-seconds flight-time per tracked electron, and therefore the energy losses are negligible anyway.

[4] The magnetron drift leads to an azimuthal motion of the electron around the trap center, but does not change the radius of the electron's trajectory.

Dependency on kinetic energy

The kinetic energy of stored electrons heavily influence the trap length. This is shown graphically in figure 6.4 for two kinetic energies of 1 keV and 5 keV. The simulation results are given in figure 6.5, and show that the trap has a length of about $1\,\mathrm{m} to 2\,\mathrm{m}$, and it strongly increases with higher kinetic energies of the electrons. This is of course to be expected, as the electrons are confined mainly by the electric potential well (figure 6.2). The region of confinement is therefore determined by the points where the electrons' kinetic energy "crosses" the potential walls. As the potential does not have a steep edge, but drops from $-18.6\,\mathrm{kV}$ to $0\,\mathrm{V}$ along a certain interval on the z-axis, the trap length varies with kinetic energy. The curvature of the potential walls also vary along the z-axis; this explains the non-linear increase in trap length that is shown in figure 6.5.

Of course, in reality an electron will loose its kinetic energy while it is stored in the trap, e. g. by scattering. This implies that the trap length is not fixed for an arbitrary electron, but slowly decreases with time as kinetic energy is lost.



(a) Trap dimensions at $E_{kin,0} = 1\,\mathrm{keV}$.

(b) Trap dimensions at $E_{kin,0} = 5\,\mathrm{keV}$.

Figure 6.4.: Trap dimensions with different initial kinetic energies. The two figures each show a visualization of a large number of simulated electron tracks. It can clearly be seen that the trap length increases significantly with higher kinetic energies. However, the general form of the trap stays the same, and is basically defined by the flux tube. The shown tracks correspond to a number of electrons started at the center of the pre-spectrometer solenoid, and with varying radial positions corresponding to the available flux tube in the setup. The initial polar angles cover a range of $0° to 60°$.

Dependency on starting angle

Apart from the kinetic energy, the initial polar angle of an electron also has an impact on the trap length. This is shown in the lower half of figure 6.5, also for different kinetic energies. This influence is much smaller than the one of the kinetic energy itself, but measurable especially for higher energies. This dependency can be explained easily by taking into consideration that a high starting angle implies that a certain part of kinetic energy is stored in the transversal energy of the electron (compare relation (3.2)). At the trap endpoints, only the longitudinal kinetic energy is relevant to work against the electric potential. Thus a high starting angle results in a part of kinetic energy "missing" to overcome the potential walls, and therefore leads to an effectively reduced trap length.

The fraction of kinetic energy stored in the transversal component is given by the relation (see section 3.2)

$$E_\perp = E_0 \cdot \sin^2 \vartheta \,, \tag{6.1}$$

where ϑ denotes the angle of the electron's momentum against the magnetic field line ("pitch angle", $\vartheta = \measuredangle(\boldsymbol{p},\boldsymbol{B})$).

From this equation, one would assume a much larger dependency on the starting angle; e. g. with an angle of 60°, only 25 % of kinetic energy corresponds to the longitudinal component and the trap length should decrease by a factor of 4. However, one must consider the influence of the inhomogeneous magnetic field on the electron's momentum vector. Similar to the effect of the MAC-E filter explained in section 3.2, the conservation of magnetic moment results in a transfer from transversal to longitudinal energy towards the trap endpoints. Therefore the longitudinal energy available at the turning points is higher than what would be expected from the above equation, and the trap length is not decreased as much.

Dependency on starting radius

Simulations with a variation of the radial starting position show no strong influence on the trap length. This also holds true for higher kinetic energies; however for very large kinetic energies above 10 keV, the trap length is a bit smaller for higher radii. These deviations in trap length are of the order of mm, and compared to the overall trap length in the range of meters seem to be negligible if one examines only the trap dimensions. The dependency on starting radius is therefore not further investigated here.

Magnetic mirror effect

A rather interesting effect which could be seen in the simulation results is the magnetic mirror effect. The effect is equivalent to the one discussed in section 3.2.2, where the magnetic mirroring prevents electrons with high starting angles from entering the spectrometers. The same applies to the electrons which are stored in the Penning trap: The pre-spectrometer solenoid provides a strong magnetic field of 4.5 T at the mid-region of the trap, and the field strength drops to much less than 1 T at the edges of the potential well, depending on the axial

position. Therefore, an electron starting at some distance away from the solenoid's center with a high polar angle will experience magnetic mirroring, i. e. it will be reflected in the vicinity of the magnet. The exact angle at which electrons will be reflected depends on the ratio between the magnetic field at the starting position and the maximum magnetic field in the trap, which is given by the strength of the solenoid ($B_{max} = 4.5\,\mathrm{T}$):

$$\vartheta_{max} = \arcsin\sqrt{\frac{B_{start}}{B_{max}}}\,. \tag{6.2}$$

For example, at $z = -2.6\,\mathrm{m}$ the magnetic field is about 0.7 T (compare figure 6.2), and therefore the starting angle will be limited to $\vartheta_{max} = 23°$ according to (6.2). Electrons with a higher starting angle are reflected near the magnet, and thus are effectively confined to a smaller region of the trap than other, unreflected electrons. A graphical representation of this behavior is given in the top half of figure 6.6: For starting angles lower than ϑ_{max}, no mirroring is taking place (indicated by the arrows); higher starting angles result in a reflection at some specific axial position that is given by the value of the "mirroring magnetic field" B_{mir}. This value defines the magnetic field at which the electron will be reflected, and increases with the starting angle. This also implies that electrons with starting angles above ϑ_{max} are reflected earlier, i. e. at some distance from the magnet's center that depends on the starting angle.

The effect was further investigated by simulations, and the results are shown in the bottom half of figure 6.6. The plot shows the trap length for electrons started with various starting angles at four different axial positions on both sides of the pre-spectrometer solenoid. The results show that at first the trap length is effectively independent of the polar angle[5], until the value ϑ_{max} is reached. Then the trap length is drastically reduced, and even higher starting angles lead to an additional decrease in trap length. This last effect follows directly from the upper plot in figure 6.6: With higher polar angles, the electrons are reflected at some distance away from the magnet's center (at a lower magnetic field), and this distance increases with higher angles. The blue line in the plot shows electrons started at $-2\,\mathrm{m}$, where ϑ_{max} is higher than 60°. As the simulations used only starting angles below this value, no mirroring is seen in this case and the sharp "drop" in trap length is missing. It should also be noted that the exact mirroring angle changes if the electron is started in a region with an electric potential $U \neq 0\,\mathrm{V}$. In this case, the electron gains longitudinal kinetic energy on its path towards the magnet, resulting in an effectively changed initial polar angle.

The magnetic mirror effect applies to any electrons with a high-enough starting angle, where the exact starting angle that leads to mirroring depends strongly on the axial starting position. These can be electrons from the source, which become stored in the trap due to energy loss, but more probably these will be secondary electrons created by scattering inside the trap. While the angle of source electrons is already limited due to the magnetic mirror effect that is inherent to the KATRIN setup[6], the secondary electrons can have arbitrarily high polar angles.

Magnetic mirroring is only effective as long as no transversal energy is lost. But since stored electrons loose this transversal energy by the emission of synchrotron radiation very quickly

[5] The slight decrease in trap length is a result of the "missing longitudinal energy" with higher starting angles, as discussed in the previous section.

[6] Here the polar angles are limited to 63.4° by the magnetic field gradient from 3.6 T at the WGTS to 4.5 T at the pre-spectrometer; see also chapter 3.

(in the order of 1 s, see 4.3.1), the electrons are not affected by the magnetic mirroring for very long. This is important to consider with respect to the proposed electron-removal methods: electrons that are reflected on the *other* side of the solenoid – where none of these methods are employed – would be confined to that part of the trap, and could not be removed, if the mirroring was kept up for a long time. However, due to the electron's loss of transversal energy, they can pass through the center of the magnet within seconds, and can then be removed from the trap by either method. In ref. [Hil11], the energy loss due to synchrotron radiation at $B = 4.5\,\mathrm{T}$ was estimated to $\Gamma \approx 8\,\mathrm{s}^{-1}$. Therefore, electrons will loose their transversal energy within roughly $1\,s to 10\,\mathrm{s}$ at the available magnetic fields (see also section 4.3).

It is also worth mentioning that the magnetic mirroring could significantly change the trap properties: Without mirroring, the electrons are confined by the two walls of the electric potential. In the case of mirroring, one of the trap endpoints is instead created by the *magnetic* field, and the electrons are reflected in a region where no high electric fields are present. It is clear that this could result in different trap characteristics in some cases; however, this possibility was not further investigated in this thesis, and the existing results show no entirely different behavior of electrons that are affected by magnetic mirroring.

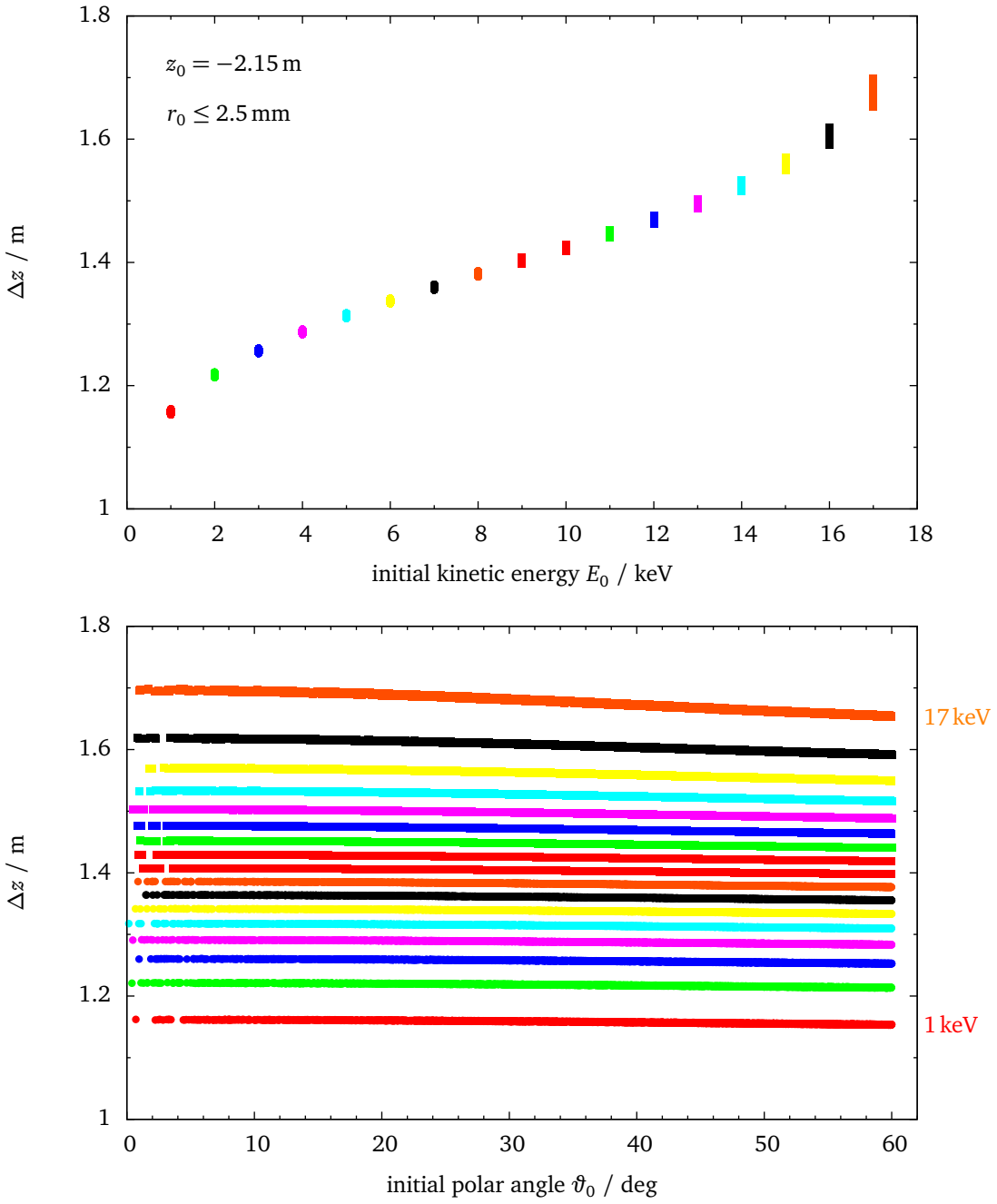


Figure 6.5.: Axial dimensions of the trap. The upper plot shows the dependency of the trap length on the initial kinetic energy of stored electrons. Here 2500 electrons where tracked for each fixed starting energy, with varying starting angle up to 60° and varying radial starting position. The lower plot shows the same simulation data, but plotted against the starting polar angle. Both plots show that the kinetic energy has by far the biggest influence on the trap size, followed by the starting angle. Other parameters such as radial starting position have only negligible impact. Both plots use the same color scheme for identifying the initial kinetic energy.

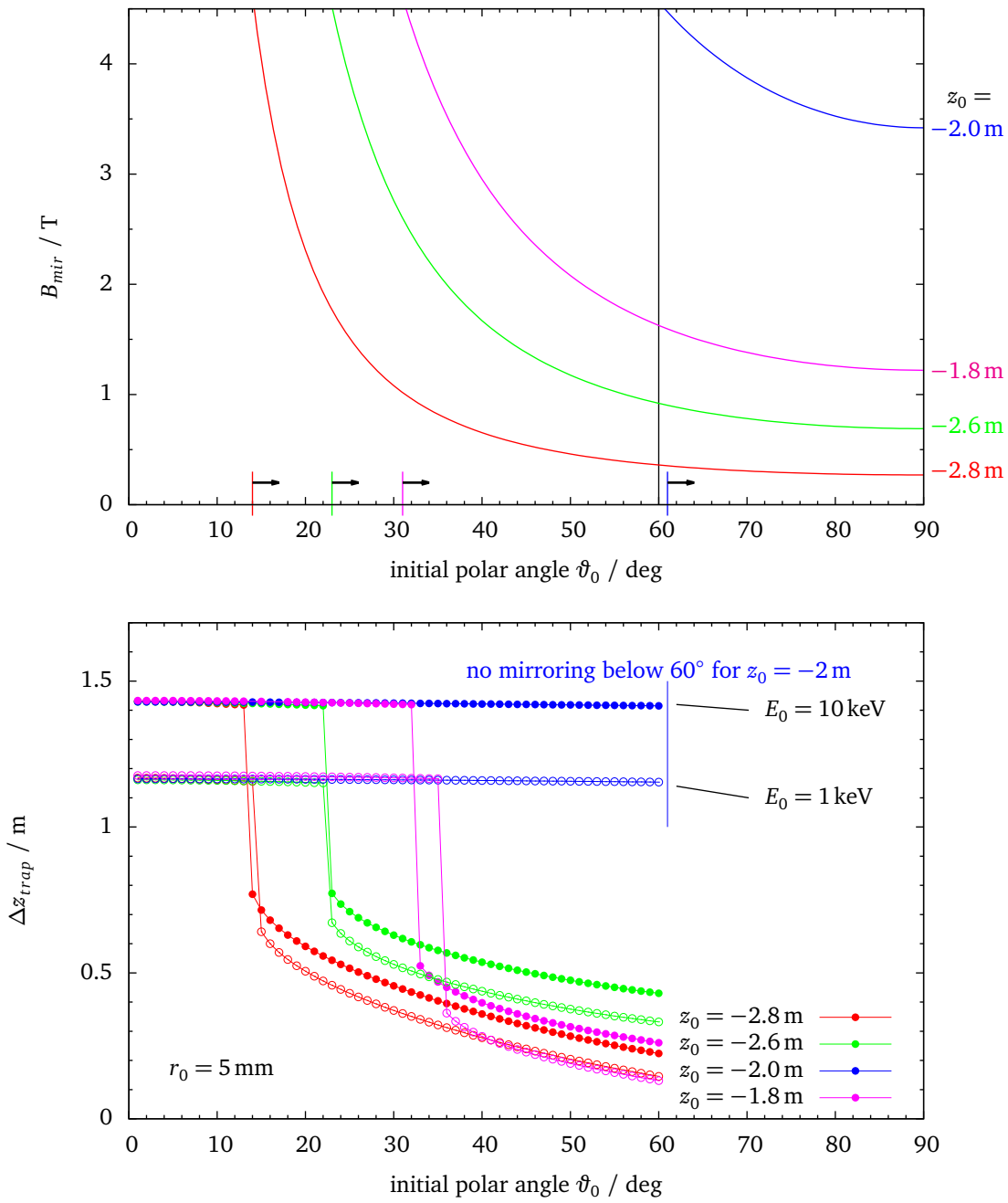


Figure 6.6.: Impact of the magnetic mirror effect on the trap length. In the first plot it can be seen how the magnetic mirror effect leads to a reflection of electrons. The plot shows the magnetic field at which an electron will theoretically be reflected, when it is started with given polar angle at a certain axial position. The mirroring angle depends only on the ratio of initial and maximal magnetic field. The second figure shows simulation results by plotting the trap length against the starting angle, which was investigated in a range of $\vartheta_0 = 0° to 60°$. Depending on the axial starting position, the trap length is drastically reduced once the starting angle passes a certain value. This value marks the angle where magnetic mirroring starts to occur and corresponds to the mirroring angle from the first plot. Due to the inhomogeneity of the electric fields, the exact mirroring angle depends on the electron's kinetic energy. Both plots use the same color scheme for identifying the axial starting position.

6.2.2. Movement of stored particles

In section 4.1 it was explained that the motion of a particle stored inside a Penning trap can be described by three superimposed components: The slow magnetron drift around the trap axis, a faster oscillation along the axis and the very fast cyclotron motion around the magnetic field lines with comparably low amplitude. The cyclotron frequency depends solely on the surrounding magnetic field, while the other two frequencies also depend on a number of parameters like the trap depth and its geometry. However, in the complex setup of the considered Penning trap, the approximations may no longer hold true. Especially the inhomogeneity of the electric and magnetic fields should have an influence on the stored electrons' motion, as they experience different fields along their trajectories.

A main goal of the simulations was to investigate the dependencies of the three oscillation frequencies on different parameters, like initial kinetic energy or initial polar angle of the electrons. In the following, the three frequencies – cyclotron, axial and magnetron oscillation – will be referred as ω_c, ω_z and ω_m, respectively. The results presented here were obtained by tracking a single stored electron with Kassiopeia for each set of starting parameters. Using the KARMA interface discussed in section 5.1.4, the large number of simulations could be managed efficiently, and the simulations were executed in parallel on the available computing cluster.

The simulations where done without energy losses, as only very short time scales were considered here (within the order of μs), and both scattering and synchrotron energy losses would be negligible. In addition, skipping energy losses allows to examine the frequency dependencies more accurately.

Cyclotron frequency

In a homogeneous magnetic field, the cyclotron frequency is constant as it depends only on the field's magnitude, as defined in equation (4.2). The magnetic field in the trap, however, has both axial and radial dependencies, which can be seen in figure 6.3. This implies that the cyclotron frequency will change along an electron's flight path, but nonetheless should still be proportional to the magnetic field at each fixed position, if one assumes that relation (4.2) is still correct.

The results shown in figure 6.9 show that this is indeed case. The data was retrieved directly from simulation output, i. e. from the single-electron tracks. The cyclotron frequency is computed internally by Kassiopeia, and written to the output file if enabled by the user. The first plot in the figure shows the cyclotron frequency plotted against axial position, and it can be seen that the cyclotron frequency follows the magnetic field, i. e. it reaches its maximum at the center of the pre-spectrometer solenoid at $z = -2.15\,\mathrm{m}$ and drops down towards the endpoints of the trap. A yet better proof that (4.2) holds true is given in the second part of the figure, where the cyclotron frequency is plotted against the magnetic field. The plot shows a linear dependency so that $\omega_c \sim B$.

The last plot shows the cyclotron radius ("gyroradius") of an electron in the trap region. The cyclotron radius can be computed by

$$r_c = \frac{m_e v_\perp}{e|\boldsymbol{B}|} = \frac{p_\perp}{eB}\,. \tag{6.3}$$

Because of the transversal momentum term in this equation, the cyclotron radius depends on the total kinetic energy and the polar angle, and changes along the electron's trajectory due to the inhomogeneous magnetic field. The cyclotron radius has its minimum at the center of the pre-spectrometer solenoid, and reaches local maxima at the trap endpoints. This behavior is what one would expect from (6.3). Although the radius depends on the transversal momentum and the magnetic field – which both decrease towards the trap endpoints –, the decrease of the magnetic field is larger than the decrease of transversal momentum. This leads to an increase of the gyroradius, which can also be seen considering the following relations:

$$r_c = \frac{m_e v_\perp}{eB} = \frac{2E_\perp}{v_\perp eB} \tag{6.4}$$

$$\sim \frac{1}{v_\perp} \qquad \text{with} \quad \frac{E_\perp}{B} = \mu = \text{const.} \tag{6.5}$$

$$\sim \frac{1}{\sqrt{B}} \qquad \text{with} \quad B \sim E_\perp = \frac{1}{2} m_e v_\perp^2 \,. \tag{6.6}$$

Consequently, the gyroradius scales with the inverse square-root of the magnetic field in the adiabatic case.

In section 4.3, it was estimated from the trap properties that the cyclotron frequency should be in the order of $10^{11}\,\mathrm{s}^{-1}$. This is indeed the case, as the simulation results in figure 6.9 show. However, the cyclotron frequency varies by almost two orders of magnitude due to the large variations in the magnetic field. It has a maximum of about $1.25 \times 10^{11}\,\mathrm{s}^{-1}$.

For the removal of trapped electrons, the other two characteristic frequencies are far more important than the cyclotron frequency, as the cyclotron oscillation is very fast, but weak in amplitude. Although the cyclotron radius depends on the kinetic energy, it stays in the order of $10^{-6}\,\mathrm{m} to 10^{-5}\,\mathrm{m}$, and thus does not have a large effect on the electrons' exact positions in the case of static magnetic fields.

Axial oscillation frequency

The axial oscillation is the second-fastest component of the electrons' motion, and is caused by the reflection of electrons on the sides of the potential well (or by magnetic mirroring, see above). In section 4.3 the axial oscillation frequency was estimated to be in the order of $5 \times 10^7\,\mathrm{s}^{-1}$.

The simulations were performed using the same methods as described above. The axial oscillation frequency was retrieved from simulation output by first defining the turning points of the electron track. This can be done by continually checking the direction of momentum at each step of the track. When the direction changes, a turning point has been found, and the time between two turning points defines a half oscillation period. To get meaningful results, five full oscillations were simulated instead of one in each track, and the mean value of the determined periods was used to compute the oscillation frequency[7].

[7] To reduce the amount of output Kassiopeia produces, only every 100th step was written into the output file, which results in some spread when calculating the exact turning-points of stored electrons. Due to the small step lengths, this spread was found to be in range of $\Delta T/T < 10^{-3}$, and averaging over a few oscillation periods will provide accurate results.

According to the results shown in figure 6.10, the axial oscillation frequency is in the range of $5 \times 10^7\,\mathrm{s}^{-1} to 5 \times 10^8\,\mathrm{s}^{-1}$, and therefore loosely fulfills the estimation. However, it depends strongly on kinetic energy and polar angle. The simulation results will be explained in the following, corresponding to the four plots in the figure:

a) The strong dependency on the kinetic energy is a result of the higher speed of the electrons. For the axial oscillation, the longitudinal speed (and thus, kinetic energy) is the relevant factor. If the frequency is proportional to the longitudinal velocity, it follows that

$$\omega_z \sim \nu_\parallel \sim \sqrt{E_\parallel} \qquad \text{with} \quad E = \frac{1}{2} m v^2 \,. \tag{6.7}$$

The Lorentz factor of the moving electrons is $\gamma \approx 1$, therefore the non-relativistic energy can be used here.

Additionally, there is a dependency on the initial kinetic energy E_0 due to the increased trap length. From the results of the previous sections, it is fair to assume that this dependency is approximately linear. The total dependency on E_0 should therefore basically follow $\omega_z(E_0) = a\sqrt{E_0} - bE_0$. This is confirmed by the fit that is shown in the plot for one of the curves. As seen below, the effective kinetic energy is also modified by the polar angle.

b) There is also a strong dependency on the initial polar angle of the electrons. At starting angles below 23°, a part of the total kinetic energy is stored in the transversal component, and therefore the effective longitudinal energy is decreased at higher angles. As the oscillation frequency depends on the (longitudinal) kinetic energy as shown above, this explains the decrease in frequency at angles below 23°. Then, at a starting angle of 23°, the magnetic mirroring comes into effect: The trap length is drastically reduced, and so is the flight path along an oscillation period. This then leads to a sudden increase in oscillation frequency. At even higher angles, the trap length is further reduced because the electrons are reflected at a larger distance away from the pre-spectrometer solenoid, and therefore the oscillation frequency increases slightly with the starting angle.

Of course, the angle of 23° in this case depends on the axial starting position of the electrons. Here the electrons were started at $z = -2.6\,\mathrm{m}$, and the mirroring angle ϑ_{max} is 23°. For other starting positions, the behavior would be similar, but the exact dependencies may differ.

c,d) There is no dependency on the initial radial or axial position (shown in the last two plots). This is to be expected, since the electrons are simply reflected on the sides of the potential well. If one considers an "effective" initial kinetic energy by adding the electric potential, i. e. $E_0' = E_{0,kin} - U(\boldsymbol{x}_0)$, the dependency on starting position can be explained solely by the changes in electric potential.

The axial oscillation is fairly important for the electron removal methods, as its frequency defines the axial velocity of an electron, and thus how much time a stored electron spends at a given axial position within a certain time interval. For the "electron catching" methods (the electron catcher and the wire scanner) it is additionally important as it defines the size of the "gaps" along the circumference of the magnetron motion: At a given point on the z-axis, the magnetron motion makes up a circle with its normal equal to the trap's main axis. Because the

axial oscillation is superimposed on the slower magnetron motion (figure 6.7), the electron will be only at a discrete set of points along the circumference. This is shown schematically in figure 6.7.

The gap length can be estimated by using the following considerations. The number of axial oscillations along a full magnetron motion can be given as

$$n_{osc} = \frac{\omega_z}{\omega_-} . \tag{6.8}$$

With a magnetron radius ρ – which can be calculated for each axial position from the conservation of magnetic flux –, the circumference of the magnetron motion is given by

$$l = 2\pi\rho . \tag{6.9}$$

Then the length of the gaps along the circumference is

$$\Delta l = \frac{l}{n_{osc}} = 2\pi\rho \cdot \frac{\omega_m}{\omega_z} . \tag{6.10}$$

For a magnetron radius of 0.01 m and a frequency ratio ω_z/ω_m of 10^3 (which are both reasonable values according to the simulation results), the gap length results to 6.28×10^{-5} m. This is much smaller than the diameter of the electron catcher and in roughly the same order as the cyclotron radius. Therefore, an electron will certainly hit the electron catcher along a full magnetron motion, without "skipping by" due to the mentioned gaps. This will become relevant in the next section of this chapter, where the electron catcher is investigated more closely.

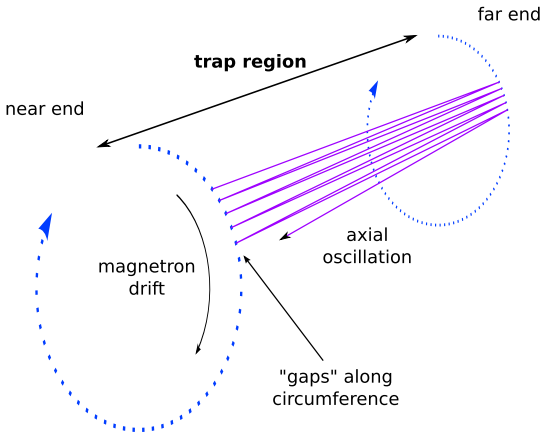


Figure 6.7.: Magnetron and axial motion of stored electrons. The combination of the two results in "gaps" along the circumference of the magnetron motion. The distance between these gaps depends on the ratio of the magnetron and axial oscillation frequencies.

Magnetron frequency

The slowest component of the stored particles' motion is the magnetron oscillation around the trap axis. It is caused by a combination of azimuthal drifts, as explained in section 4.1. The high electric fields in the trap region make the $E \times B$ drift the main component of the magnetron motion[8]. Figure 6.8 shows that the $E \times B$ drift reaches very large values especially near the main spectrometer electrode, and has a strong radial dependency. This will become important when discussing the simulation results further below.

[8]The ∇B drift is comparably small because of the small gradient of the magnetic field in the trap region.

The simulations were done in a way similar to the simulations discussed in the previous section, but in this case a special feature of Kassiopeia was used: The "Magnetron" exit condition will end a track once a full magnetron cycle is complete. Then the magnetron frequency can be retrieved directly from the start and end times of the track.

In section 4.3 it was estimated that the magnetron frequency should be in the order of $10^5\,\mathrm{s}^{-1}$. The results shown in figure 6.11 confirm the estimation: The frequency is in the range of $1\times10^5\,\mathrm{s}^{-1} to 1\times10^6\,\mathrm{s}^{-1}$, with strong dependencies on the kinetic energy and the polar angle that are somewhat similar to the dependencies of the axial oscillation frequency. However, the results show a decrease in frequency for higher kinetic energies, and there is also a dependency on the radial starting position. These results will be explained in the following, again corresponding to the four plots in the figure:

a) In the case of the axial oscillation, the dependency on kinetic energy was found to be a result of the increase velocity of the electrons. In the case of the magnetron motion, however, the effect is different: Since the magnetron motion is a result of the $E\times B$ drift (see section 4.1), its frequency should be independent of the kinetic energy.

   However, higher kinetic energies result in an increased trap length, and the trap endpoints are moved outwards along the z-axis. The magnetron drift depends strongly on the electron's position, which is especially important near the main spectrometer electrode where the electric field gradients are large and the field direction changes. Furthermore, the fields at the trap endpoints are much more important than the fields along the track, because the particle is slowed down and spends more time near the endpoints.

   The observed frequency as seen in the last plot in figure 6.11 for high kinetic energies can therefore be explained by considering the total $E\times B$ drift that the electrons experience along a full magnetron turn. With higher kinetic energies, the endpoints are shifted outwards, and especially near the main spectrometer electrode the electric field are changing rapidly even for small variations of axial position. If the field now would change in a way that the total $E\times B$ drift along a full magnetron cycle is reduced for higher E_0, this would explain the frequency decrease.

   Figure 6.8 shows the $E\times B$ drift speed along an electron track for three different kinetic energies. At an energy of 10 keV, the drift is maximal near the main spectrometer electrode, and is always in positive direction. For larger energies however, the drift changes its direction near the trap endpoints, resulting of the distorted electric field (see lower part of the figure). As noted above, the drift speed near the endpoints is more effective because the electron spends most of its time in this region. Therefore the increases trap length leads to a compensation of the positive and negative $E\times B$ drifts along the electron's track, so that the total drift along a magnetron cycle and thus the magnetron frequency is decreased for higher energies.

   Furthermore, figure 6.11 shows that the magnetron frequency increases for energies above some specific value around 14 keV. The simulation data also shows that the direction of the magnetron drift changes its direction (i. e. from clockwise to counterclockwise when looking into positive z-direction). Considering the explanations above, the reason for this behavior is that now the section with negative azimuthal drift is larger than the section of positive drift (compare figure 6.8), resulting in a negative net-drift in azimuthal direction. Because the negative component increases with higher energies as

the turning points are shifted outwards, the total drift also gets larger, resulting in an increasing magnetron frequency.

The magnetron frequency is additionally reduced for high starting angles that are below the mirroring angle ϑ_{max}, because then a part of the kinetic energy is stored in the transversal component. This leads to an effectively reduced longitudinal kinetic energy and trap length, which in turn results in a decrease in magnetron frequency.

For angles higher than ϑ_{max}, the magnetic mirroring causes a different dependency on the kinetic energy. This can be seen even better in the next plot, and will be explained below.

b) The dependency on the starting angle is a result of the same effects discussed for the axial oscillation. Starting angles below ϑ_{max} result in a "missing" longitudinal energy, which changes the total $E \times B$ drift experienced by the stored electrons.

On the other hand, angles above ϑ_{max} lead to magnetic mirroring with a reduced trap length, and the frequency increases again towards higher angles. A possible explanation is that the $E \times B$ drift is now happening at only one of the turning points, as the other one is in a region with low electric and high magnetic fields (i. e. near the pre-spectrometer solenoid). This is shown in the upper part of figure 6.8, where the resulting $E \times B$ drift in the mid-region near the solenoid is small. The lower part of the figure also shows a map of the azimuthal $E \times B$ drift. It can be seen that the drift velocity reaches large values near the main spectrometer electrode.

c) The slight dependency on the radial starting position is connected to the electrons' motion along the magnetic field lines: Stored electrons will follow these field lines, and a high starting radius will result in a larger distance to the trap axis at the endpoints. Especially near the main spectrometer electrode the electric field has a strong radial dependency; therefore an electron that gets close to the electrode will experience a different $E \times B$ drift than for the case of an inner trajectory. The exact trajectory depends on both axial and radial starting positions, since the flux tube widens with further distance from the pre-spectrometer solenoid. Therefore this r_0-dependency can be seen more clearly in the plot for electrons started at $-2.6\,\mathrm{m}$ than for electrons started at $-2.4\,\mathrm{m}$.

d) There is almost no dependency on the axial starting position visible. For the same reasons mentioned for the axial oscillation, this is to be expected: The starting position defines the field line the electron is following. As explained above, some dependency would be visible for even higher starting radii, since the electrons would experience different $E \times B$ drifts if their distance to the electrodes is small (compare figure 6.8).

The magnetron frequency is very important especially for understanding the removal of electrons by the electron catcher. A stored electron will slowly proceed in azimuthal direction due to the magnetron motion. Therefore, the time for an electron to reach the electron catcher depends on its azimuthal starting position and on the speed of the magnetron motion. These effects will be further analyzed in the next section, where the electron catcher is investigated in detail.

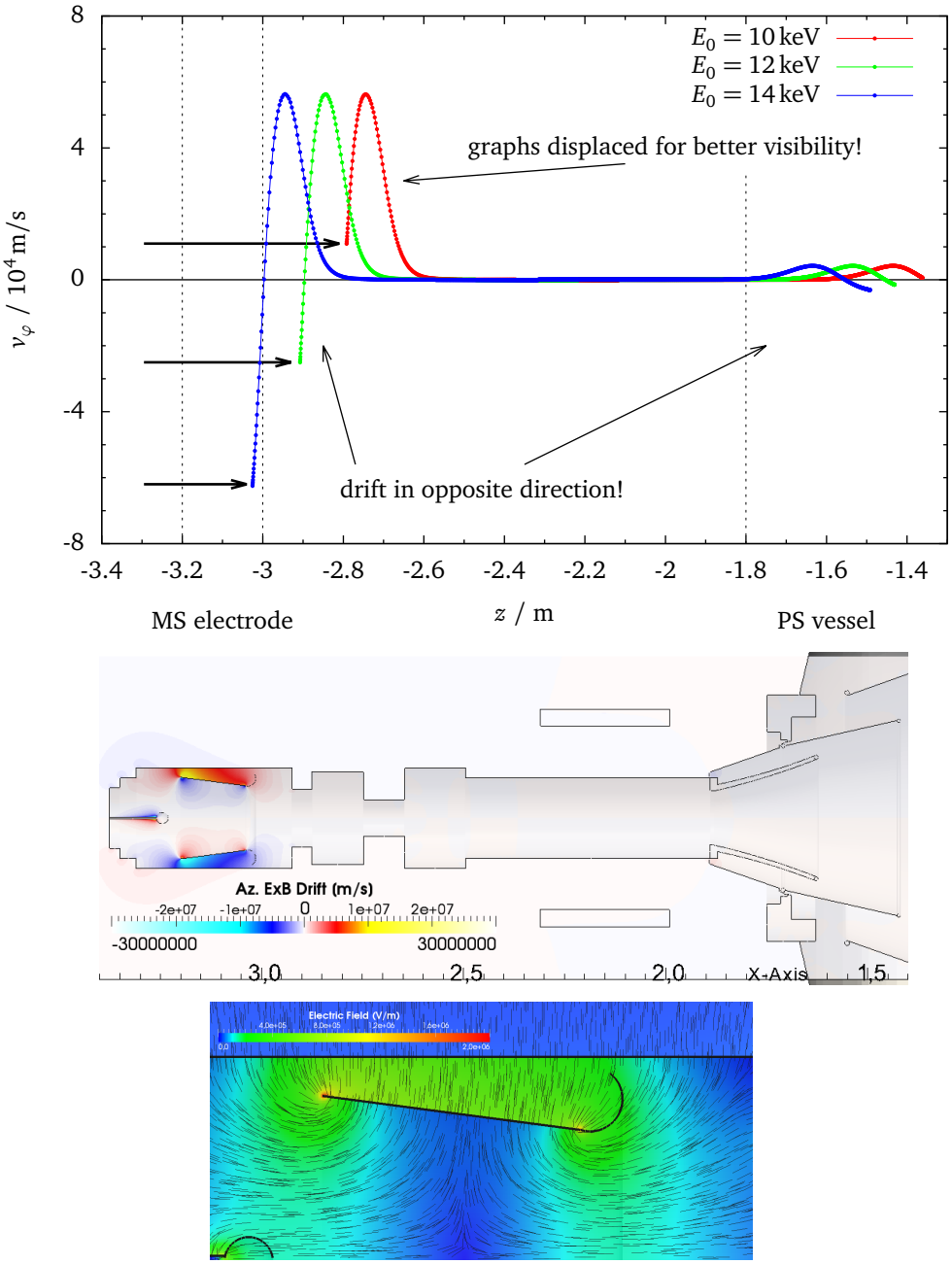


Figure 6.8.: Azimuthal $E \times B$ drift of stored electrons. The upper plot shows the azimuthal $E \times B$ drift velocity experienced by a stored electron. It can be seen that the drift speed is higher where large electric fields are present, especially near the main spectrometer electrode. The kinetic energy of the electrons changes the total $E \times B$ drift due to the increased trap length. For energies above 12 keV, the drift speed changes its direction near the trap endpoints (marked by the arrows). For better visibility, the curves for 10 keV and 12 keV were moved by 0.2 and 0.1 meters on the z-axis, respectively. The second graphic shows a two-dimensional map of the azimuthal $E \times B$ drift velocity within the simulation geometry (using a logarithmic scaling). It can be seen that the drift velocity reaches large values near the main spectrometer electrode, and its direction changes inside the electrode region. The last graphic shows a detailed view of the electric field in the region near the electrode as an overlay of electric field vectors and the field magnitude.

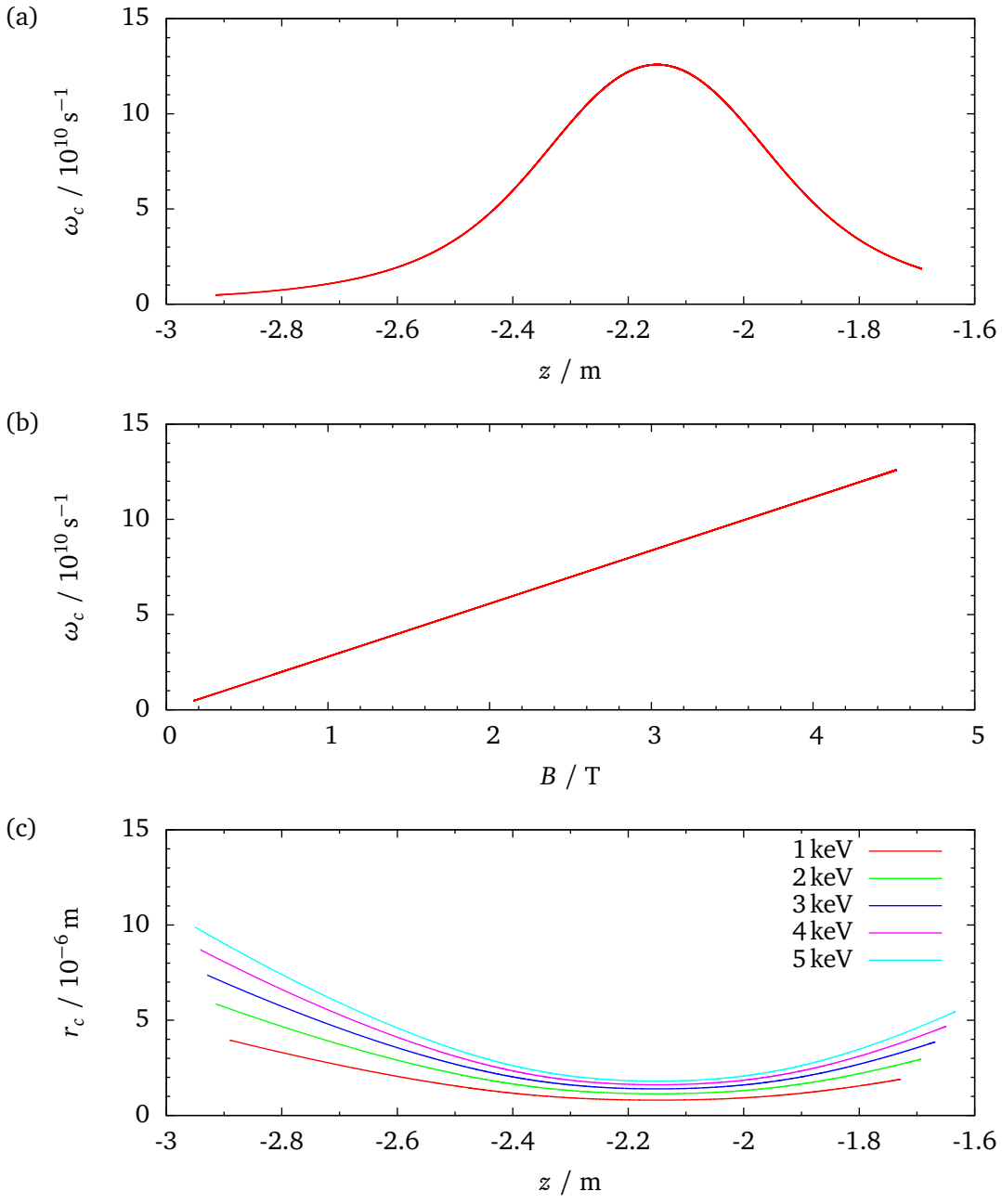


Figure 6.9.: Cyclotron frequency of stored electrons. The first two plots shows the cyclotron frequency of stored electrons. The frequency is proportional to the magnetic field, as can be seen in the second plot. The third plot shows the according cyclotron radius ("gyroradius") of the electrons, i. e. the radius of the circular motion of an electron around a magnetic field line.

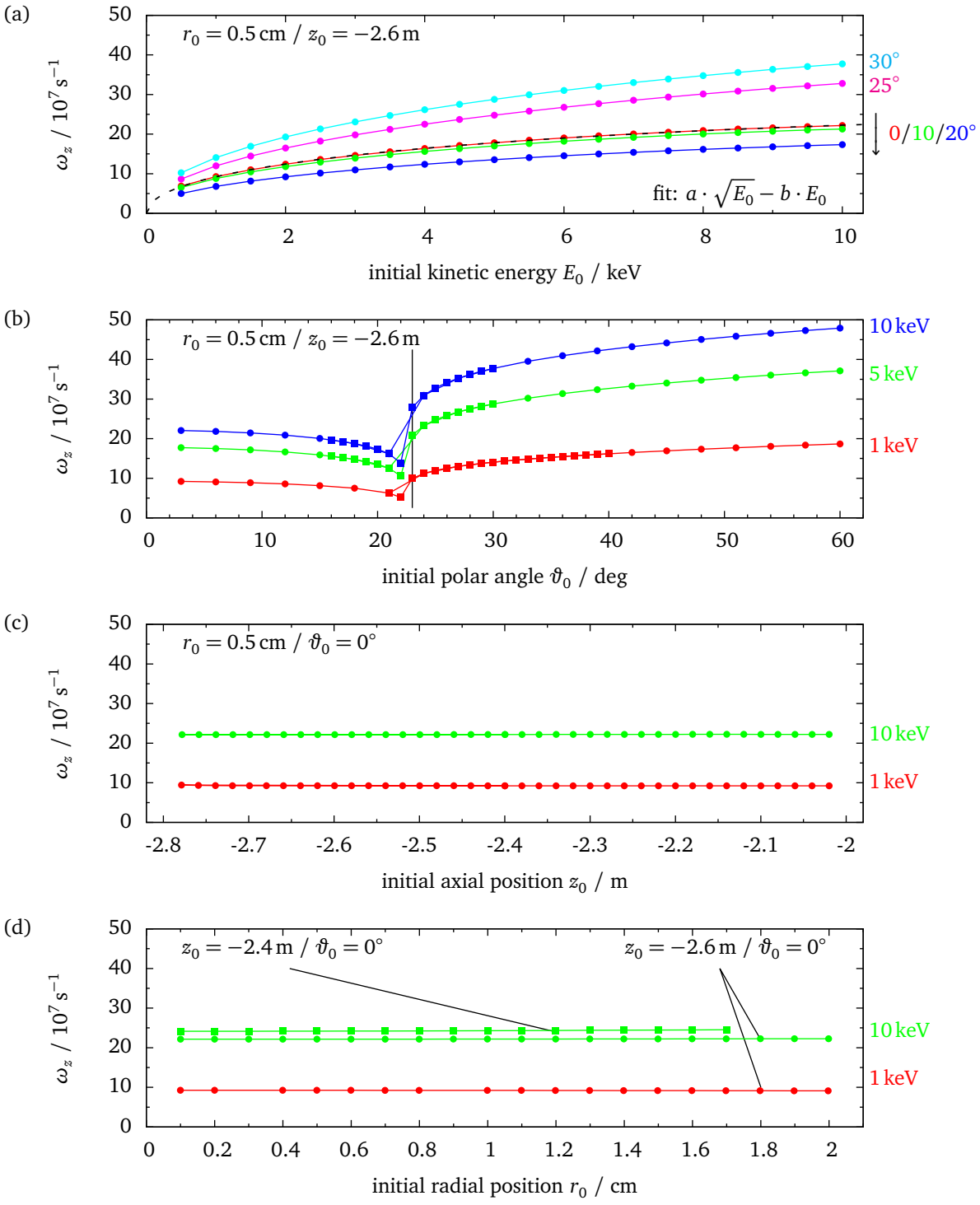


Figure 6.10.: Axial frequency of stored electrons. The four plots show the dependency of axial frequency on different initial parameters: kinetic energy, polar angle, and radial and axial position. The plots were generated from a series of single-electron simulations, each with an appropriate set of starting parameters. The points for $z_0 = -2.4\,\mathrm{m}$ in the last plot are moved up by two units for better visibility. In the ϑ_0 plot, the angle where magnetic mirroring occurs is marked with a line (23° in this case).

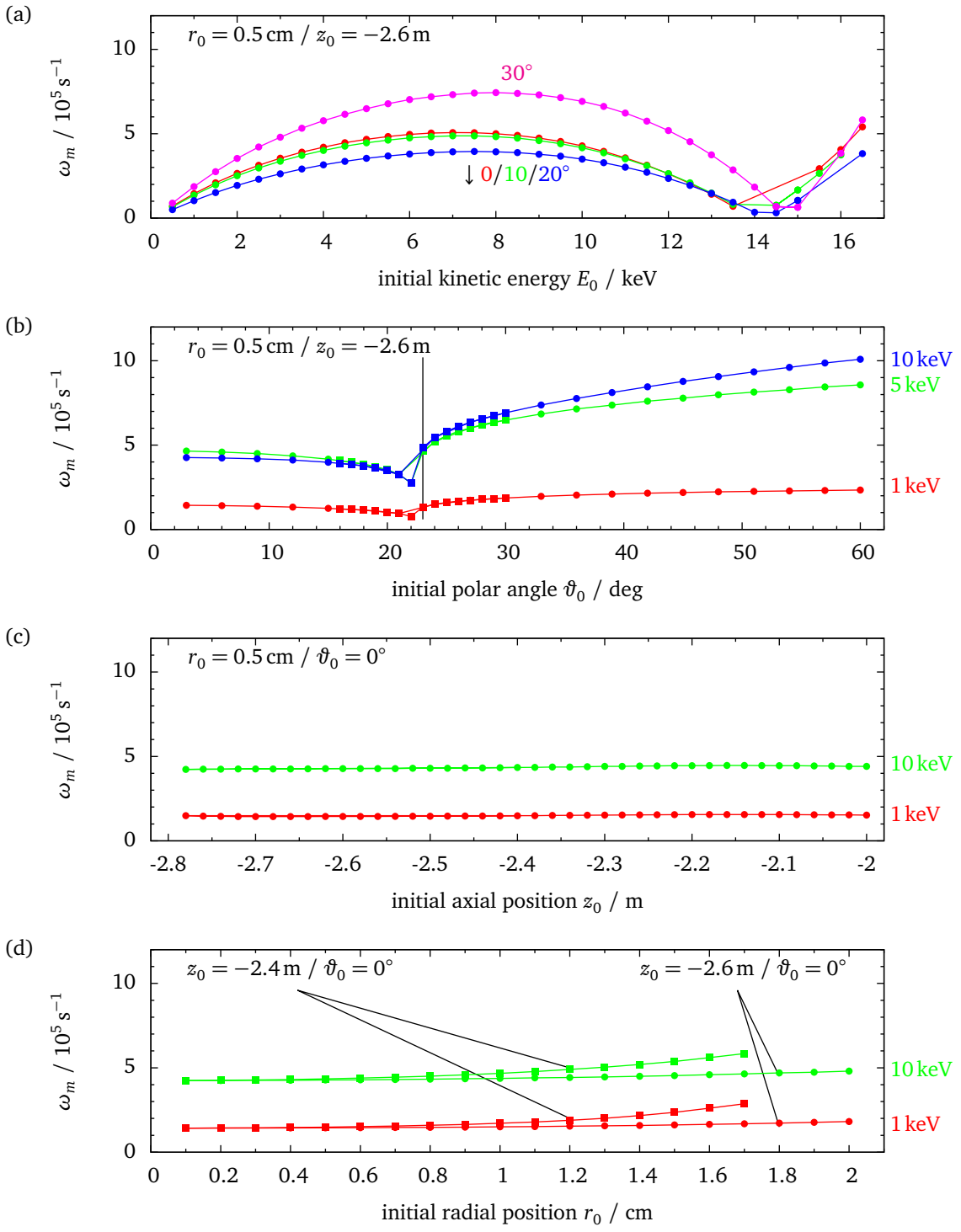


Figure 6.11.: Magnetron frequency of stored electrons. The four plots show the dependency of magnetron frequency on different initial parameters: kinetic energy, polar angle, and radial and axial position. As with the axial oscillation frequency, the plots were generated from single-electron simulations for each parameter setting. The angle where magnetic mirroring occurs is marked with a line in the ϑ_0 plot. At energies above 14 keV in the E_0-plot, the magnetron drift is in opposite direction due to the changed $E \times B$ drift (see continuous text).

6.3. Electron catcher

In this diploma thesis, two of the proposed methods to remove electrons from the trap were investigated. One of these is the "electron catcher", a solid wire that intercepts electrons on their trajectories inside the trap. The experimental studies that were done by B. Hillen showed that the electron catcher works very efficiently and removes the majority of electrons from the trap volume, but only if it is fully moved into the flux tube. The results of these studies were shortly discussed in section 4.3.2.

The simulations presented here were done to further analyze the effect of the electron catcher on stored electrons, and to provide additional information regarding its efficiency. It could be proven that the electron catcher is able to remove all stored electrons if it is fully moved into the center of the flux tube, and that the storage times of electrons with the electron catcher employed are mainly defined by the magnetron frequency.

6.3.1. Geometric setup

The simulations regarding the electron catcher used a fairly simple setup. B. Hillen's experimental setup used a stationary, solid wire with a diameter of 2 mm that could be retracted from the center of the flux tube up to a distance of 10 cm. This wire was positioned at $z = -2.8\,\mathrm{m}$ in the setup (see also figure 4.7 on page 37).

Since the electrode setup was already available from previous simulations, only the electron catcher itself had to be implemented in a way for Kassiopeia to include its effect on stored electrons: The electron catcher will basically collect and remove any electrons that make contact with the wire surface. Therefore it was possible to use an appropriate exit condition in Kassiopeia to simulate this main effect of the electron catcher. It should be noted that an electron hitting the wire surface would be able to create free electrons with low kinetic energies, as the kinetic energies of stored electrons are likely to be larger than the work function of the wire material. However, the freed electrons would have only small energies. They are also not accelerated by electric fields, since no large electric fields are present in the region of the electron catcher. Furthermore, these low-energy electrons would also be removed by the electron catcher eventually.

The already existing `ExitConditionGeometryHit` module can employ and arbitrary geometry, and is triggered once the tracked electron comes into the vicinity of the geometry surface, i. e. when the distance between the surface and the electron falls below a given threshold. For the purpose of the electron catcher simulations, a simple cylindrical geometry with appropriate dimensions was used, and the minimal distance was set to 0.05 mm, which is much smaller than the wire diameter. Note that the electrode geometry itself, which is used to compute the electric fields, is not changed by this method. This is allowed because the electron catcher and the surrounding ground electrode are at a potential of 0 V, and there is no large electric potential surrounding the region of the electron catcher. This can also be seen in figure 6.2 on page 68 if one looks at the electron catcher's position ($z = -2.8\,\mathrm{m}$): The deviation of the electric fields in the trap region due to the electron catcher is very small.

The main advantage of this approach is that it does not require additional computing resources due to its simplicity, as would be the case if the electrode setup was extended to include the electron catcher. In particular, the electron catcher would break the rotational symmetry of the setup, and computation times would drastically increase for this reason. With the chosen approach it was possible to perform a series of Monte-Carlo simulations where a large number of electrons is tracked with random starting parameters. The simulations therefore provide a way to closely investigate the cleaning efficiency of the electron catcher with a large fraction of possible starting parameters covered.

Storage times

It was already mentioned that electrons which are stored in the trap follow the magnetic field lines and undergo a slow magnetron drift around the trap main axis. This motion is superimposed with an axial oscillation and a much faster cyclotron motion. As the amplitude of the cyclotron oscillation is only in the range of 10^{-6} m (see section 6.2.2) and therefore much smaller than the solid wire's diameter, it is negligible in this context. In the previous section it was also explained that the axial oscillation does not prevent electrons from reaching the electron catcher on their trajectory by "skipping by" along the magnetron cycle. The "gaps" along the magnetron cycle were calculated to be in the same order as the amplitude of the cyclotron motion.

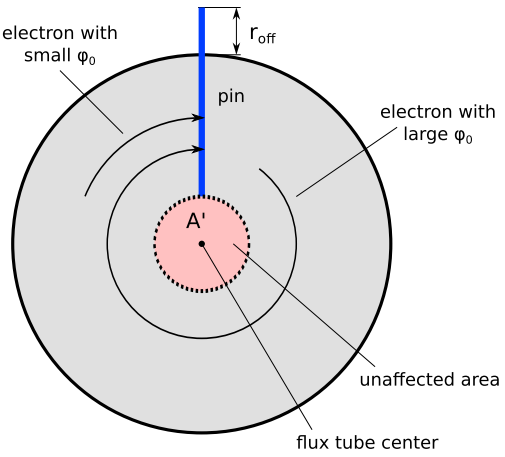


Figure 6.12.: Schematic view of electron removal by the electron catcher. The flight path (and lifetime) of stored electrons depend strongly on their azimuthal starting position φ_0, as shown in the diagram. The offset parameter r_{off} defines the area of the flux tube that is not affected by the electron catcher, i. e. where electrons would not be removed.

This leads to the conclusion that a stored electron should certainly hit the wire surface – and thus be removed from the trap volume – within one full magnetron cycle. This effect can be confirmed by simulations, as shown in figure 6.13: It shows a selection of electron tracks that were generated with the simulations described above, using different starting parameters. It can be seen that the electrons slowly follow the magnetron drift around the trap axis, and hit the electron catcher within one magnetron cycle.

Additionally, there is a strong dependency on the azimuthal starting position: Electrons which are created "in front" of the electron catcher[9] (e. g. by scattering processes) will have a short flight path and reach the electron catcher quickly, while electrons starting at the other side will need almost a full magnetron cycle to get removed (figure 6.12). This effect can clearly be seen in the simulation results, and is shown in the lower half of figure 6.13 on page 90 for two

[9] In azimuthal direction, i. e. in the direction of the magnetron drift.

single electron tracks. The left electron starts with a high azimuthal distance and therefore has a long flight path, while the right electron which started at a low azimuthal distance reaches the electron catcher quickly.

In conclusion, the lifetime of an electron should be limited by the magnetron frequency if the electron catcher is employed, and a strong dependency on the azimuthal starting position of electrons is to be expected. These considerations are confirmed by Monte-Carlo simulations, with the simulation results shown in figure 6.14. The electrons were started on a disk at $z = -2.6\,\mathrm{m}$ with a radius of 2.5 cm. This maximum radius corresponds to the effective width of the flux tube at the given position, thus avoiding that particles hit the ground electrode and being lost for the simulations. The initial kinetic energy was chosen randomly from the interval $1\,\mathrm{keV} to 16\,\mathrm{keV}$. The initial polar angle was fixed in each of the four simulation runs; this is shown in the plots by the differently colored data points. The simulations were done with energy losses due to synchrotron radiation and scattering on H_2 molecules at a residual gas pressure of 10^{-9} mbar. In contrast to the simulations presented in the previous sections, the additional effects allow a more realistic investigation of the electron catcher's effect on stored electrons: It is possible that the energy losses have an effect on the storage times and thus on the electron catcher efficiency. However, it will be shown later that the very short storage times result in only small energy losses. Secondary electrons created in the simulations by scattering were not investigated further to keep the required computing time down to a minimum. Additionally, with the low storage times with the electron catcher in place, scattering events which actually create secondary particles are rare.

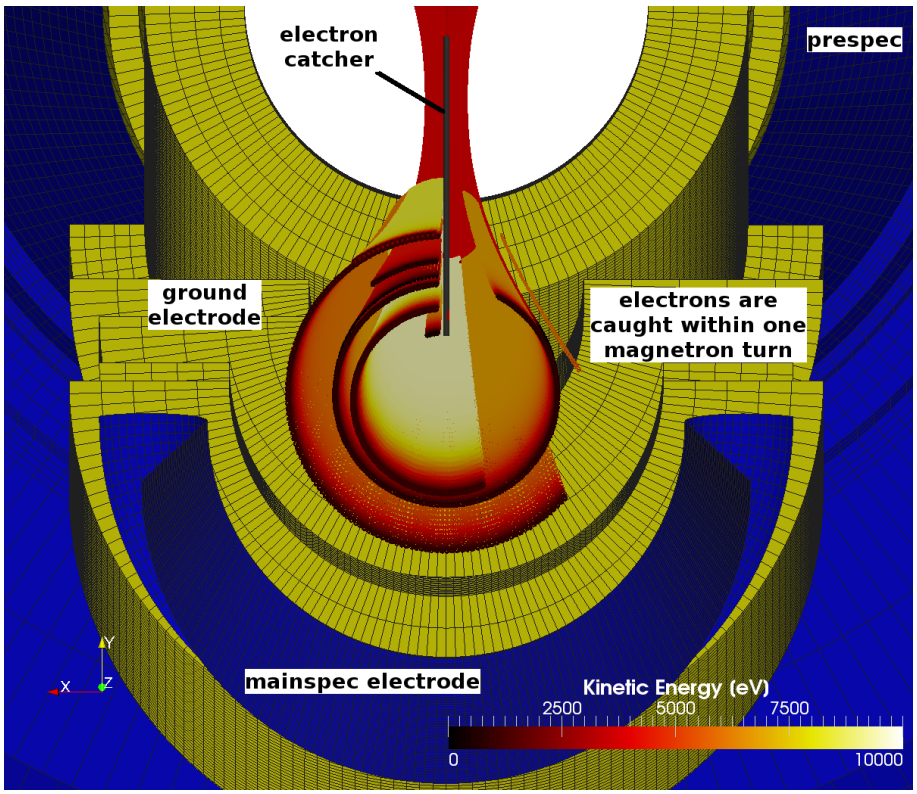
The first three plots in the figure show the storage times of all simulated electrons, i. e. the time until they are removed by the electron catcher, in dependency of their initial kinetic energy, and their initial radial and azimuthal positions. There is an obvious dependency on the kinetic energy because the kinetic energy has a strong impact on the magnetron frequency, as simulations discussed in the previous section have shown. Since the maximum storage time of an electron is mainly given by the time it takes to undergo a full magnetron turn, the storage time should directly depend on the magnetron frequency:

$$\Delta t_{max} = \frac{1}{f_-} = \frac{2\pi}{\omega_-}. \tag{6.11}$$

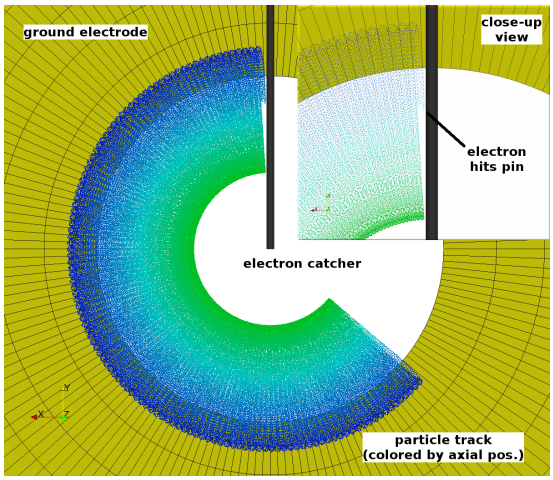
The storage times are limited to a range of $20\,\mu s to 50\,\mu s$, which nicely matches the typical timescales of a full magnetron cycle according to section 6.2.2. Of course there are storage times that are much smaller than these maximum times; this is the case when not a full magnetron cycle is needed for the electron to be removed.

The dependency on kinetic energy also matches the inverse dependency of the magnetron frequency from figure 6.11. However, it must be considered that the additional energy losses have an impact on the storage times: Stored electrons loose some of their energy, which effectively changes the magnetron frequency (see section 6.2.2) and therefore also influences the resulting maximum storage times.

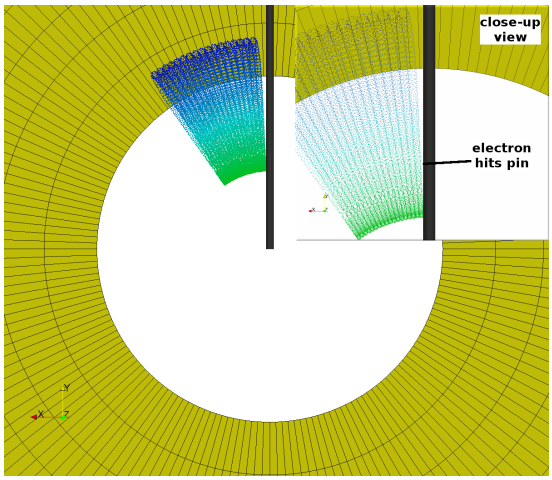
The second plot shows no dependency on the radial starting positions of the electrons. This also matches the results from section 6.2, where the magnetron frequency was found to be almost independent of the starting radius. In the shown plot, the density of data points increases for larger radii. This is just an effect of Kassiopeia's particle generator that was used for the



(a) Simulated electron tracks with the pin in place.



(b) Electron started at large φ_0.



(c) Electron started at small φ_0.

Figure 6.13.: Simulated particle track with the electron catcher employed. The top figure shows tracks of ten electrons that were simulated with Kassiopeia. It can be seen that all tracks end at the electron catcher (except one on the right; here the electron hit the ground electrode before it could reach the electron catcher). The tracks where electrons started at a high distance away from the electron catcher have significantly longer paths than others where electrons started in front of the electron catcher. It is also apparent that the magnetron motion forms a circle at each fixed z-position. The bottom two figures each show a detailed view of a single electron track, looking in positive z-direction (i. e. towards the pre-spectrometer). The dependency of the electron lifetimes on the initial azimuthal position is obvious.

simulations[10]: The interval for the starting radius is normalized by the area of the starting disk, and therefore the starting positions follow an $\sqrt{r}$-distribution.

More interesting results are provided by the third plot, which shows the electrons' storage times against their azimuthal starting angle φ_0. As explained above, the azimuthal starting position has a direct impact on the path length and thus the lifetime of an electron. In the shown plot, the electron catcher is located at an azimuth of 0°. It can clearly be seen that electrons that start directly "in front" of the electron catcher have very short lifetimes, while electrons that start at the maximum distance have comparably long lifetimes. In this plot the effect of the initial polar angle can also be seen: For an starting angle above $\vartheta_{max} = 23°$, the storage times are smaller than those of lower starting angles. This is an effect of the magnetic mirroring (see previous section), which reduces the trap length and therefore the electrons' lifetimes. Starting angles below ϑ_{max} also have an impact on the lifetime because of the "missing longitudinal energy" that was also discussed earlier.

It should be noted that the simulations used an additional exit condition that limits the maximum number of computation steps. This was done to keep the required simulation times in a practicable range. With the used simulation settings the step length is shorter for higher starting angles, therefore the maximum path length that was allowed in the simulations also depends on the starting angle. This is the reason why there are no data points for $\vartheta_0 = 20°$ and $\varphi_0 \lesssim 60°$. For the case of $\vartheta_0 = 30°$, the magnetic mirroring reduces the trap length and thus the total path length of trapped electrons.

The last plot shows the path length against the azimuthal starting angle. It can be seen how the path length is limited to roughly 450 m by the maximum number of simulations steps. The impact of the polar angle on the total flight path is also visible here: A high polar angle leads to a cyclotron motion with higher amplitude. This results in a longer effective path for an electron even if the path of its guiding center[11] stays the same. For polar angles above ϑ_{max} the path length is reduced due to magnetic mirroring.

Electron catcher efficiency

The above results confirm that the electron catcher is able to efficiently remove electrons from the trap, as it was demonstrated in the previous experimental studies. The storage times with the electron catcher in place were found to be in the millisecond-range. The simulations showed that scattering events in this short time are rare, and no high-energetic secondary particles were produced in the investigated electron tracks. As will be shown in the next section, the typical time-scale for scattering is in the range of seconds. The high electron-removing efficiency of the electron catcher – electrons are removed within one magnetron turn – and the short storage times match the experimental results nicely, and explain the electron catcher's effectiveness in cleaning the trap.

In B. Hillen's experimental setup, the electron catcher was employed as a retractable device, i. e. it was possible to move the solid wire up to 10 cm out of the center of the flux tube (called "offset" in the following). The experimental results showed a background rate that increased

[10] The particle generator `KPAGEDiskPositionCreator` uses a disk with fixed radius with its normal in z-direction.

[11] The electron's motion can be split up into the motion of its guiding center, which follows the magnetic field lines, and an additional "gyration" that corresponds to the cyclotron motion around the field line.

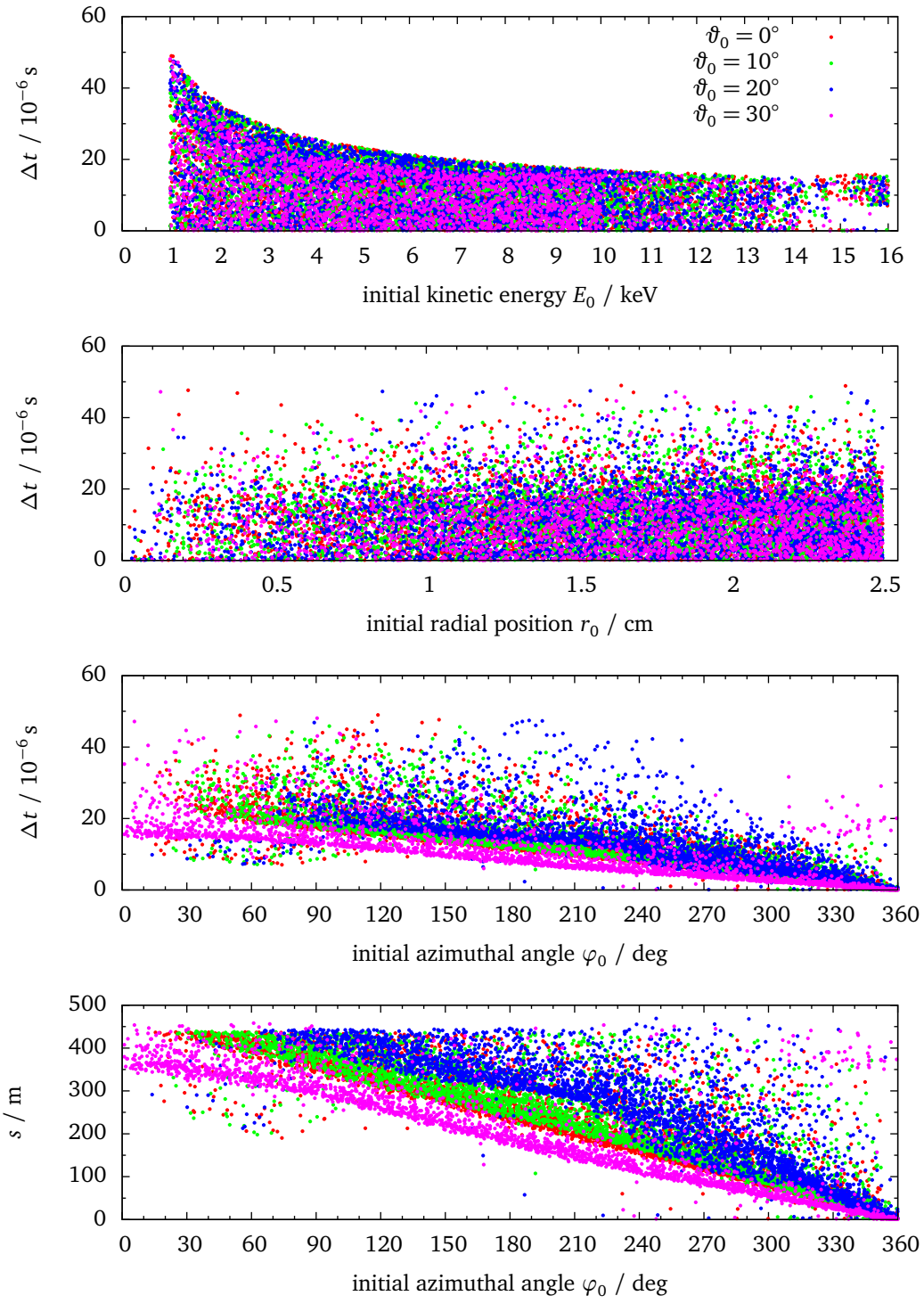


Figure 6.14.: Particle lifetimes with the electron catcher employed. The four plots show the results of a Monte-Carlo simulation with roughly 15 000 electrons being tracked until caught by the electron catcher. Each electron was started at a fixed axial position of $z_0 = -2.6\,\mathrm{m}$, but with varying kinetic energy (E_0), radial position (r_0), azimuthal angle (φ_0), and polar angle (ϑ_0). The first three plots show the particle lifetimes against different starting parameters. The most obvious effect comes from the azimuthal starting position: electrons with higher azimuthal angles take much longer to reach the electron catcher. This can also be seen in the last plot, where the path length is plotted against the azimuthal angle φ_0.

with the electron catcher offset, and seemed to approximately have a quadratic dependency. Using the knowledge of the electrons' motion gained by computer simulations, it is possible to explain this behavior.

The electron catcher covers an cross-sectional area of the flux tube where all electrons are eventually removed when they hit the wire surface. This shadowed area obviously depends on the wire offset, since a part of the flux tube will not be affected by the electron catcher when it is not moved fully into the center (see figure 6.12). This unaffected area is given by

$$A' = \pi r_{off}^2 \,. \tag{6.12}$$

From the simulation results it can be determined that the radius of an electron's trajectory is almost constant for a fixed axial position: The magnetron drift forms a circular path, with small deviations due to the cyclotron motion and "gaps" along this circle due to the axial oscillation. However, both are small in amplitude and do not have a large effect on the electron's circular motion.

If a homogeneous filling of the trap is assumed, i. e. with a constant electron "column density" $\rho_{e^-}(r) = \text{const.}$, the produced background rate should be proportional to this area. In section 6.2.1 it was shown that the trap length depends only slightly on the initial radius of the electrons. The variations in trap length due to changes in the radial starting position are of the order of cm even for high kinetic energies of 10 keV and more, and even less for smaller energies. The trap length itself is about 1.5 m, and increases significantly with higher energies. If the trap is filled by electrons with a homogeneous radial distribution, the column density should therefore be constant.

In addition, scattering processes of electrons should have an uniform distribution within the trap volume, because scattering itself is a random process. With a homogeneous electron density in the trap, a uniform distribution can be assumed. As these scattering processes are one of the major filling mechanism of the trap, it is fair to estimate that the trap is filled homogeneously. With the previous considerations this leads to the conclusion that the background rate will be proportional to the area that is not affected by the electron catcher:

$$\dot{N} \sim A' = \pi r_{off}^2 \,. \tag{6.13}$$

This quadratic dependency approximately matches the results of B. Hillen's studies (figure 4.9 on page 38). If additional measurements at the test setup were possible, they could even provide a way to determine the electron density in the trap region, because a deviation from (6.13) would imply a deviation from an electron density which is constant in radial direction (i. e. $\rho_{e^-}(r) \neq \text{const.}$). The measurements could also be improved by taking advantage of the segmentation of the detector, since a possible ring structure visible on the detector would imply a non-uniform generation of background electrons, and therefore a inhomogeneous filling of the trap.

6.4. Pulsed coil

Another electron removal method that was investigated in this diploma thesis is the pulsed coil. In the experimental studies by B. Hillen, an aircoil was placed at a axial position of $z \approx -3\,\mathrm{m}$. By applying a current of 300 A, it was possible to create a magnetic field that is directed against the one created by the pre-spectrometer solenoid and nullifies the global magnetic field in a limited area, as shown in figure 6.15. The experimental results showed that the coil removes electron from the trap efficiently if the current is applied in a series of short pulses. When applying only one pulse, the coil was found to be much less efficient. These experimental studies were shortly reviewed in section 4.3.2.

The simulations which are presented here were performed to provide a better understanding of the coil's effect on stored electrons in general, and to investigate a proposed explanation for the increased efficiency of the coil with multiple pulses. Prior to the simulations it was assumed that this increased efficiency could be related to a total net-effect, which accumulates over multiple pulses of the coil: Such an effect would explain why more electrons are removed by a series of pulses than by one single pulse. By studying the coil's impact on single stored electrons, it is possible to determine if such a net-effect exists.

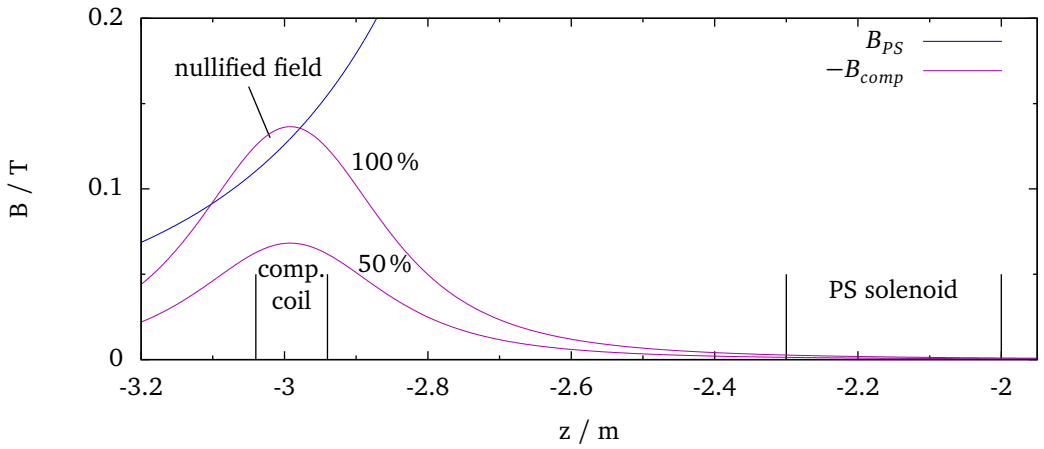


Figure 6.15.: Magnetic field of the compensation coil. The plot shows the magnetic fields of the pre-spectrometer solenoid and the compensation coil along the z-axis. The field of the compensation coil is shown at 50 % and 100 % of its nominal magnitude. It can be seen that the magnetic field is nullified in an area of roughly $-3.0\,\mathrm{m} to -3.1\,\mathrm{m}$ when the coil is fully ramped up.

The measurement of the coil's response function presented in ref. [Hil11] allows to calculate the time constant for the ramping. Figure 6.16 shows the coil's response to a 300 A pulse, and the time to reach 90 % of the nominal field strength is given with $T_{90\%} = 0.17\,\mathrm{s}$. The time constant is then:

$$0.9 = 1 - e^{-T_{90\%}/\tau} \tag{6.14}$$

$$\Rightarrow \tau = -\frac{T_{90\%}}{\ln 0.1} = 0.0738\,\mathrm{s}\,. \tag{6.15}$$

In order to allow the coil to reach its full nominal field strength, ramping times of at least 0.25 s were used in the experiments by B. Hillen, and also in the simulations presented here.

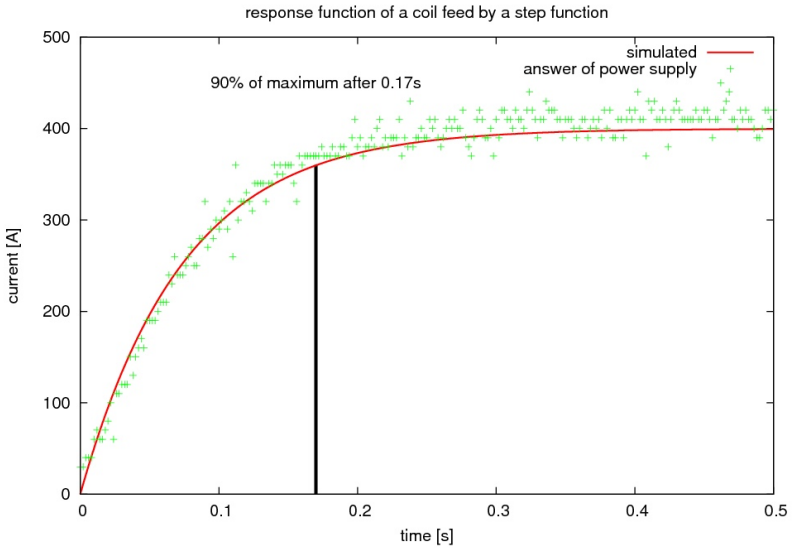


Figure 6.16.: Response function of the pulsed coil. The figure shows the impulse-response of the pulsed coil as employed by B. Hillen. The coil current reaches 90 % of its nominal value after 0.17 s, which corresponds to a time constant of 0.0738 s (see continuous text). Figure taken from [Hil11].

6.4.1. Theoretical background

In the following, the different effects of the magnetic field generated by the pulsed compensation coil will be explained. It is important to keep in mind that in Kassiopeia, electric and magnetic fields are split up into independent modules. To get physically correct results, it is crucial to use an appropriate combination of field modules. This is especially important in the case of the pulsed coil, as the changing magnetic field will also induce an electric field according to the laws of J.C. MAXWELL:

$$\nabla \times \boldsymbol{E} = \mathrm{rot}\boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{6.16}$$

$$\oint_C \boldsymbol{E}\,\mathrm{d}\boldsymbol{l} = \oint_S \mathrm{rot}\boldsymbol{E}\,\mathrm{d}\boldsymbol{A} = -\iint_S \frac{\partial \boldsymbol{B}}{\partial t}\,\mathrm{d}\boldsymbol{A}\,. \tag{6.17}$$

Figure 6.17 shows an electron that is orbiting with a cyclotron radius ρ around a point at distance r from the z-axis. It will then experience an induced electric field in addition to the $\boldsymbol{B}$-field in case the magnetic field is ramped up or down. The latter case is shown here; then the alteration of the magnetic field is directed against the field itself, and an outward drift is induced. This drift effect will be explained in the following, based on the remarks by E. OTTEN in ref. [Ott10].

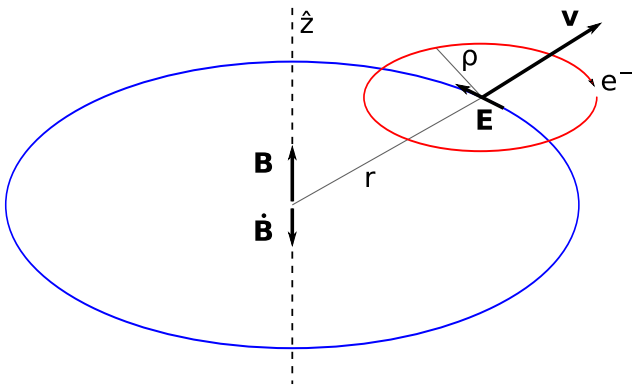


Figure 6.17.: Electron in a down-ramping magnetic field. The electron is experiencing a magnetron drift around the trap axis ($\hat{z}$) on a circular orbit with radius r. This motion is superimposed by the faster cyclotron motion with radius ρ. When the magnetic field in positive z-direction is ramped down, the changing magnetic field is directed anti-parallel to the field, and a positive radial drift $\boldsymbol{v}$ is induced on the electron.

For a magnetic field in z-direction [12] ($\boldsymbol{B} = B_z \boldsymbol{e}_z$), the right half of (6.17) gives:

$$\iint_S \frac{\partial \boldsymbol{B}}{\partial t}\,\mathrm{d}\boldsymbol{A} = -\frac{\partial B_z}{\partial t} \cdot \int_0^{\rho}\int_0^{2\pi} \rho\,\mathrm{d}\rho\,\mathrm{d}\varphi \tag{6.18}$$

$$= -\frac{\partial B_z}{\partial t} \cdot \rho^2 \pi\,. \tag{6.19}$$

The induced electric field will then be in azimuthal direction ($\boldsymbol{E} = E_\varphi \boldsymbol{e}_\varphi$) according to (6.17):

$$\oint \boldsymbol{E}\,\mathrm{d}\boldsymbol{l} = 2\pi\rho E_\varphi \overset{(6.19)}{=} -\frac{\partial B_z}{\partial t} \cdot \rho^2 \pi \tag{6.20}$$

$$\Rightarrow E_\varphi = -\frac{1}{2}\frac{\partial B_z}{\partial t}\rho\,. \tag{6.21}$$

In a similar way, a change in the radial and azimuthal components of the magnetic field will result in an electric field which is induced in a different direction. However, in the investigated case of the pulsed coil, the magnetic component in z-direction is by far the largest.

A major effect of the additional magnetic field by the coil is an increase of the flux tube diameter. This is a direct effect of the altered magnetic field, and is therefore independent of the induced

[12] Within the trap region, the B_z component is by far the largest one of the total magnetic field, especially far away from the pre-spectrometer solenoid where the pulsed coil is located.

electric field discussed above. From the adiabatic conservation of magnetic flux it follows:

$$\Phi = \iint_S \boldsymbol{B}\,\mathrm{d}S \tag{6.22}$$

$$\Rightarrow \Phi_z = B_z \pi \rho^2 = \text{const.} \tag{6.23}$$

In the investigated situation, the coil ramping times are in the order of 0.25 s, while the oscillation frequencies of the trapped electrons are in the order $10^5\,\mathrm{s}^{-1}$ and higher. Therefore the adiabatic case still applies, and the above formula is correct in the context of the pulsed coil.

As Φ_z remains constant at all times, a decreasing magnetic field results in a widening of the flux tube, and vice versa. Therefore the up-ramping of the coil forces an electron that is following the magnetic field lines on a trajectory with higher radius. In the down-ramping phase of the coil the total magnetic field increases, and the inverse effect is taking place. Because the trajectory radius is defined only by the total magnetic field, which goes back to its normal value after a ramping phase is complete, there is no net-effect resulting from the magnetic field alone.

However, the combination of electric and magnetic field results in an additional $E \times B$ drift of the trapped electron. This drift results from the Lorentz equation (4.1) and can be calculated as

$$\boldsymbol{v}_{drift} = \frac{\boldsymbol{E} \times \boldsymbol{B}}{|\boldsymbol{B}|^2}\,. \tag{6.24}$$

Taking the induced azimuthal electric field from (6.21), it becomes clear that the drift has to be in radial direction due to the cross-product in (6.24):

$$\boldsymbol{v}_{drift} = \frac{E_\varphi \boldsymbol{e}_\varphi \times B_z \boldsymbol{e}_z}{B_z^2} \tag{6.25}$$

$$\Rightarrow v_{drift,r} = -\frac{\rho}{2}\frac{\dot{B}_z \cdot B_z}{B_z^2} = -\frac{\rho}{2}\frac{\dot{B}_z}{B_z}\,. \tag{6.26}$$

Therefore a down-ramping magnetic field will results in a radial drift in outwards direction, as shown in figure 6.17. When the magnetic field is ramped up again, the drift changes to inwards direction. This drift therefore increases the effect of the changing flux tube radius that was explained above. The drift speed reaches its maximum magnitude at the beginning of each ramping phase, when the magnetic field changes most rapidly.

As seen in (6.26), the drift speed depends on the electron's distance from the z-axis. Since this distance is not constant while the coil is ramped up and down, it is possible that there is a net-drift which would force the electron on another trajectory after a full coil pulse is complete. This possible net-effect could explain the efficiency of the pulsed coil, and was the main driving force behind the simulations that are discussed in this section.

Note that the drift is a direct effect of the Lorentz equation, which is solved at every simulation step in Kassiopeia, and thus does not have to be implemented explicitly. If the ExactStep-Computer (ESC, see section 5.1) is used, all drifts are automatically included. If one of the

AdiabaticStepComputer modules is used, one must be careful to include the $E \times B$ drift in the simulations. However, in the simulations discussed here, only the ExactStepComputer was employed.

To make simulations of the pulsed coil possible within Kassiopeia, the ramping of the magnetic field B_z and the induced electric field E_φ were implemented in two field modules. The implementation will be explained in the following section.

6.4.2. Implementation of the magnetic and induced electric fields

The coil itself can be fully described with the Magfield module available in Kassiopeia. From the coil specifications in ref. [Hil11], an appropriate geometry input file could easily be put together (see appendix C). To get a magnetic field that is directed against the pre-spectrometer solenoid's field, it was necessary to apply a negative current density to the coil since the direction of current is defined by its sign in Magfield.

However, to make the *pulsed* coil work in Kassiopeia simulations it was necessary to extend the software accordingly. The ramping of the magnetic field was implemented in an additional module which uses the existing Magfield code for the field calculations. The module was originally written by N. WANDKOWSKY (KIT Karlsruhe), but was further extended to account for the non-homogeneous magnetic field that is created by the coil, and to include a more realistic, exponential ramping of the coil. For Kassiopeia to include the induced electric field, an additional module is needed, as otherwise one would get physically incorrect results. This module was also initially written by Wandkowsky, but it too had to be extended to reflect the enhancements applied to the magnetic field module. The exact implementation of both modules is explained below (the full source code is included in appendix B).

Magnetic field

For the magnetic field, a simple approach was used: The static magnetic field of the coil can already be computed directly in Kassiopeia by using the existing Magfield code. Therefore the new module just uses the static field that is computed by Magfield at the current particle position, and the ramping is then superimposed by applying a scale factor in the range $0 to 1$.

The module was furthermore extended to allow for exponential ramping, which better reflects the real behavior of the coil. The coil's time constant can be calculated from ref. [Hil11], and results to $\tau = 0.0738\,\mathrm{s}$ (see (6.15) at the beginning of this section).

For the case of exponential ramping, the implementation of the magnetic field can be written as

$$B_z(t) = \hat{B}_z \cdot f(t) \tag{6.27}$$

$$= \hat{B}_z \cdot \begin{cases} 1 - e^{-t/\tau} & \text{(ramp-down)} \\ e^{-t/\tau} & \text{(ramp-up)} \end{cases}, \tag{6.28}$$

where $f(t)$ is the exponential ramping function and $\hat{B}_z$ is the static magnetic field computed by Magfield.

It is important to consider that an *up-ramping* of the pulsed coil will result in a *decrease* of the total magnetic field, as the field of the compensation coil is directed against the one generated by the pre-spectrometer solenoid. Therefore, if setting up the coil geometry correctly, the *total* magnetic field will be decreased in the ramp-up phase and go back to its normal value in the ramp-down phase. See listing B.1 in the appendix for details on the exact implementation of the coil geometry with Magfield.

Induced electric field

The induced electric field was implemented in a similar way. Like in the case of the ramped magnetic field, the already existing module was extended accordingly.

The electric field is computed directly from the derivative of the ramping function $f(t)$ in (6.27), which corresponds to an exponential modulation. This approach does not alter the results that would be obtained by using the computed magnetic field as input, as the induced electric field only depends on the change of the magnetic field and the magnetic source field $\hat{B}_z$ itself is constant at each simulation step. Therefore, the electric field implementation can be written as:

$$E_\varphi = -\frac{\rho}{2} \cdot \frac{\partial B_z}{\partial t} \tag{6.29}$$

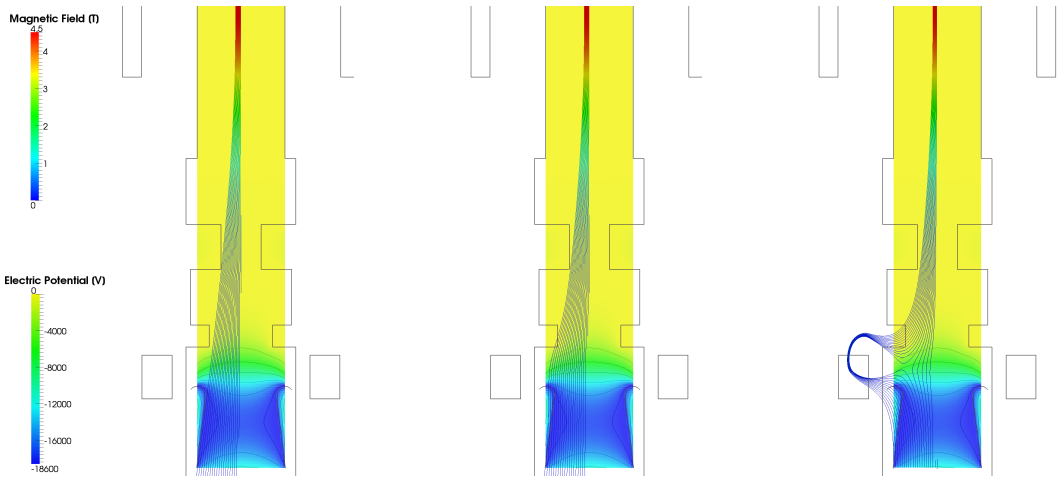
$$= -\frac{\rho}{2} \hat{B}_z \cdot \frac{\partial f(t)}{\partial t} \tag{6.30}$$

$$= -\frac{\rho}{2} \hat{B}_z \cdot \begin{cases} \frac{\partial}{\partial t}(1 - e^{-t/\tau}) & \text{(ramp-down)} \\ \frac{\partial}{\partial t} e^{-t/\tau} & \text{(ramp-up)} \end{cases} \tag{6.31}$$

$$= \frac{\rho}{2\tau} \hat{B}_z \cdot \begin{cases} -e^{-t/\tau} & \text{(ramp-down)} \\ e^{-t/\tau} & \text{(ramp-up)} \end{cases} . \tag{6.32}$$

6.4.3. Validation and example results

The effect of the coil's magnetic field is shown in figure 6.18. The three images each show a two-dimensional map of the combined magnetic fields of the compensation coil and the pre-spectrometer solenoid, in different ramping phases of the compensation coil. The field maps were generated with the *TestFieldmapVTK* program that was presented in section 5.3.1. It can be seen how the coil's magnetic field affects the form of the field lines (i. e. the flux tube) in the trap region. At 100 % of its nominal strength, more than half of the field lines cross an electrode surface, and electrons that are following one of these field lines would be removed from the trap volume. This corresponds to the main effect of the coil, which was discussed above: The widening of the flux tube forces electrons against one of the surrounding electrodes. But as noted, this effect alone does not explain the high efficiency when multiple pulses are applied to the coil.



(a) 0 % **(b)** 50 % **(c)** 100 %

Figure 6.18.: Effect of the compensation coil on the flux tube. The figures show the magnetic field lines in the flux tube due to the magnetic field of the pre-spectrometer solenoid. The two right figures show the effect of the compensating field of the additional coil (at 50 % and 100 % strength). It can be seen how the compensating field disturbs the field lines in a small region of the trap. In this region, stored electrons that are following one of the outer field lines will be forced against the ground electrode, and thus be removed from the trap. However, electrons stored at inner trajectories will not be removed by this effect alone. The field lines are overlaid with a color map and contour lines of the electric potential in the trap region.

Radial drift

An example of the results from the presented implementation is shown in figure 6.19. The four plots were created with a validation program that was build in addition to the two Kassiopeia modules. It uses a simple approach that computes the fields and the resulting radial drift at a fixed z-position within a given time range. The program also allows to include a variation of the radial position at each time-step according to the drift (this mode is called “dynamic calculation”). This more closely resembles the real behavior of a trapped electron, as it will be influenced by the radial drift and change its radial position accordingly, and thereby provides a way to validate the output of the field modules. It also makes it possible to closely examine the effect of the radial drift on stored particles.

Of course, this simple validation is not equivalent to a full Kassiopeia simulation, as the electrons motion through the trap volume and the full spatial dependencies of electric and magnetic fields are not taken into account. Furthermore, the widening of the flux tube is not included here. As explained previously, the widening of the flux tube is a result from the adiabatic conservation of magnetic flux, and provides an additional radial drift of stored electrons which increases the effect of the $E \times B$ drift. In the Kassiopeia simulations, this drift needs not to be added manually, since it is a result of the Lorentz equation which is already computed in the tracking code.

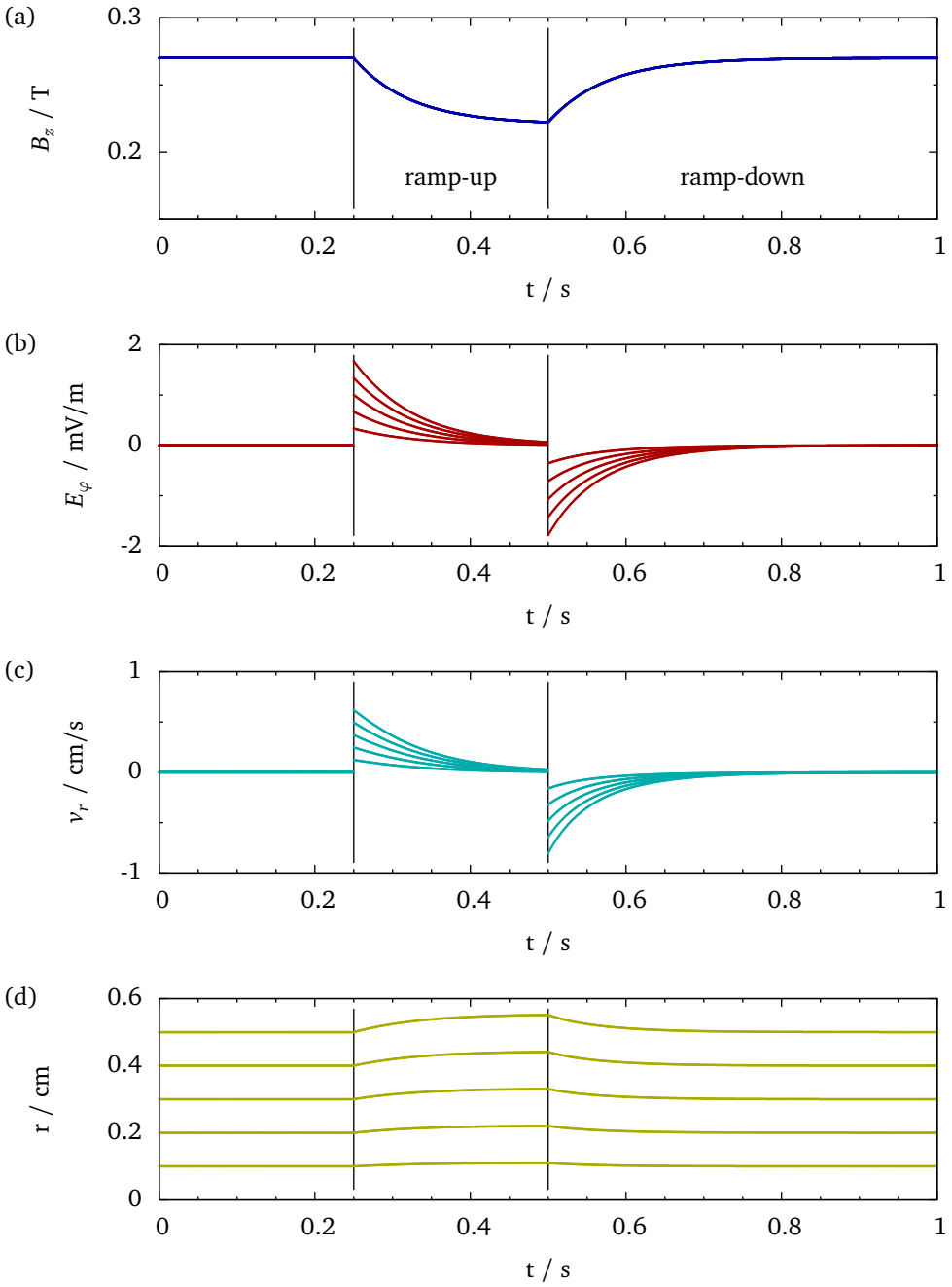


Figure 6.19.: Example output of the pulsed coil implementation. The graphs show the magnetic field at $z = -3\,\mathrm{m}$ for five different radii under the influence of the pulsed coil; the electric field that is induced by the changing magnetic field; the resulting $E \times B$ drift in radial direction; and the according change of radial position due to the drift (see continuous text for details).

The four plots in figure 6.19 show results for an electron at a fixed axial position of $z = -3\,\mathrm{m}$, i. e. near the center of the compensation coil, and with five different initial radial positions. The results will be explained shortly in the following, corresponding to the four given plots:

a) The first plot shows the total magnetic field in axial direction. The ramping of the compensation coil reduces the magnetic field (compare figure 6.15 on page 94).

b) The second plot shows the induced electric field in azimuthal direction. The induced field follows directly from the changing axial magnetic field, and depends on the radial position according to (6.21). The electric field is oriented in different direction sin both ramping phases because the derivative of the magnetic field changes its direction.

c) The third plot shows the radial drift velocity as a result of the perpendicular magnetic and electric fields. Because the electric field depends on the radial position, the drift velocity also has a radial dependency according to (6.26). The radial drift is also used to update the radial position of the electron in the validation program.

d) The last plot shows the resulting radial position, which is updated at each step of the validation routine. The radial drift results in an increase of the radial position in the ramp-up phase, and a decrease in the ramp-down phase. It can be seen that the total drifts in both phases cancels each other out, as no net-change of the radial position is observed after a full ramping phase. With this validation program, the differences in the radial position before and after a ramp are *exactly* zero, meaning that there is absolutely no net-drift observed.

6.4.4. Time-scaling

Another important feature of the implementation is the use of a "time-scaling" parameter. Because the ramping times of the coil are in the range of $0.25\,s to 0.5\,\mathrm{s}$, a full real-time simulation for a single stored electron would need to be in the order of 1 s. Given the step lengths used in Kassiopeia, which are typically in the order of nano-seconds, the simulation would need a huge number of steps in the order of billions even for a single electron track. This is far more than would be achievable in a realistic time-frame[13]. To avoid this problem, an additional time-scaling parameter was implemented in the aforementioned magnetic and electric modules, which effectively speeds up the ramping of the coil by applying an additional factor $t \to kt$ in equations (6.28) and (6.32). This makes it possible to simulate a full coil pulse using Kassiopeia, but drastically reduce the required simulation time.

To produce correct results, it is important to multiply the value of the induced electric field by this factor, too. This can be seen by comparing the total drift when using different time scales: When only applying the scaling factor to the ramping time, the total drift is larger by a factor that equals the time-scaling parameter (compare (6.26)). Obviously, the physical results should be independent of the time-scaling factor that is being used. The need for the additional factor can also be seen by computing the integral of the induced electric field within a given time range. Applying the time-scaling ($t \to kt$) to (6.32), it follows that

$$E_\varphi(t) = \text{const.} \cdot e^{-kt/\tau}\,. \tag{6.33}$$

[13] A simulation of the pulsed coil with only 500 million steps takes about one CPU-day.

(For easier reading, only the ramp-up phase is considered here.) The effective electric field in the time interval $(0,T)$ can then be calculated by

$$\int_0^T E_\varphi(t)\,\mathrm{d}t = \text{const.} \cdot \int_0^T e^{-kt/\tau}\,\mathrm{d}t \tag{6.34}$$

$$\sim \frac{1}{k} \cdot (1 - e^{-kT/\tau}). \tag{6.35}$$

It is apparent that with this implementation the electric field would be reduced by a factor k^{-1} when applying a time-scaling factor $k > 1$. Since the total induced drift directly depends on the induced electric field, the simulation results would be different when time-scaling is used, which is physically incorrect. Therefore, the implementation of the electric field includes the additional factor k, which restores the correct behavior when time-scaling is enabled.

A different approach to get the same result is to add the time-scaling factor k to the magnetic field in (6.27), and then take the derivative according to (6.29):

$$B_z(t) = \hat{B}_z \cdot (1 - e^{-kt/\tau}) \tag{6.36}$$

$$\Rightarrow \frac{\partial B_z(t)}{\partial t} \sim k \cdot e^{-kt/\tau} \tag{6.37}$$

$$\Rightarrow E_\varphi(t) \sim k \cdot e^{-kt/\tau}. \tag{6.38}$$

Using this formula in the integral from (6.34) results in an effective electric field that is independent of the time-scaling factor, hence giving physically correct results:

$$\int_0^T E_\varphi(t)\,\mathrm{d}t = \text{const.} \cdot k \int_0^T e^{-kt/\tau} \tag{6.39}$$

$$\sim 1 - e^{-kT/\tau}. \tag{6.40}$$

Comparing this correct formula to (6.35), which matches the implementation in Kassiopeia, makes clear that an additional factor k has to be included to correct the outcome. The exact implementation is shown in listing B.2 in the appendix; the core part including the additional factor from above is located in the function `GetField` beginning at line 27.

In the simulations which were done in this diploma thesis, a time-scaling factor of 10^4 was used. This ensures that the coil ramping is still slower than the magnetron motion of stored electrons (see section 6.2.2), so that the assumption of adiabatic movement is still correct. As a non-adiabatic simulation would result in visible effects such as loss of total energy, it was possible to confirm that the assumption holds true with a time-scaling factor of 10^4. The time needed for a single-electron simulation with 500 million steps is then in the order one day, which allows to investigate the effect of the pulsed coil on electrons with various sets of starting parameters. A test simulation with a scaling factor of 10^5 showed that adiabaticity is not fully preserved, and therefore would give incorrect results. On the other hand, a scaling factor of only 10^3 would require ten times as much computation time, and was therefore also rejected.

6.4.5. Tracking simulations of the pulsed coil

After the implementation and validation of the pulsed coil in Kassiopeia, tracking simulations were performed to investigate the coil's effect on stored electrons. Because of ramping times of 0.25 s, a flight-time of about at least 0.5 s had to be simulated to include a full ramping of the coil. To examine possible effects that come into account after a ramping phase, simulation times of 1 s were used. The long particle times require to include energy losses by both synchrotron radiation and scattering events, as the overall energy loss can not be neglected.

With a time-scaling factor of 10^4, the electrons' flight-time of 1 s corresponds to a simulation with a length of 10^{-4} s. It is also important to apply an appropriate scaling factor to the energy losses when using a time-scaling factor, since the effective losses correspond to the full particle time that is simulated. This can be done automatically in Kassiopeia by applying a scaling factor to the synchrotron loss, and by effectively increasing the pressure of the residual gas. This approach was used in the tracking simulations of the coil.

As will be seen later, the energy losses due to synchrotron radiation are in the order of $100\,\mathrm{eV} to 1000\,\mathrm{eV}$, depending on the electrons' polar angle and starting energy. Scattering on H_2 molecules was included with a residual gas pressure of 10^{-9} mbar as it has been at the measurements of B. Hillen. Because scattering is a random process, the energy losses can be in a range of a few eV up to several keV if secondary particles are created. These aspects are further discussed in the next section.

An example of the simulation output for an electron with starting energy $E_0 = 1\,\mathrm{keV}$ is shown in figure 6.20. The data points correspond to the outermost axial positions of the electron's trajectory, i. e. the turning points near the main spectrometer electrode[14]. In this case this equals an axial position of $-2.89\,\mathrm{m}$, and is therefore in close distance to the pulsed coil. The small trap length at the used kinetic energy does not allow the electron to travel into the center of the compensation coil at $z = -2.99\,\mathrm{m}$, which is the reason why the mentioned approach to collect data points was used[15]. The effectiveness of the coil increases for higher kinetic energies, when the distance to the coil is even smaller due to the increased trap length (compare section 6.2.1).

The four plots show the following results:

a) The first plot shows the total axial magnetic field as computed by the Kassiopeia field modules. It is a superposition of the static field of the pre-spectrometer solenoid and the time-variant field of the compensation coil.

b) The second plot shows the total azimuthal electric field, which is also taken from the Kassiopeia output. Although there are minor azimuthal components in the static electric field due to the electrode geometry, the major part comes from the induced electric field. If this were not the case, the influence of the magnetic field's ramping would not be as large.

[14] The turning point is the position where the electron is reflected by the electric potential along one axial oscillation.
[15] As will be seen in the next section, electrons that reach the center of the coil are likely to be removed from the trap.

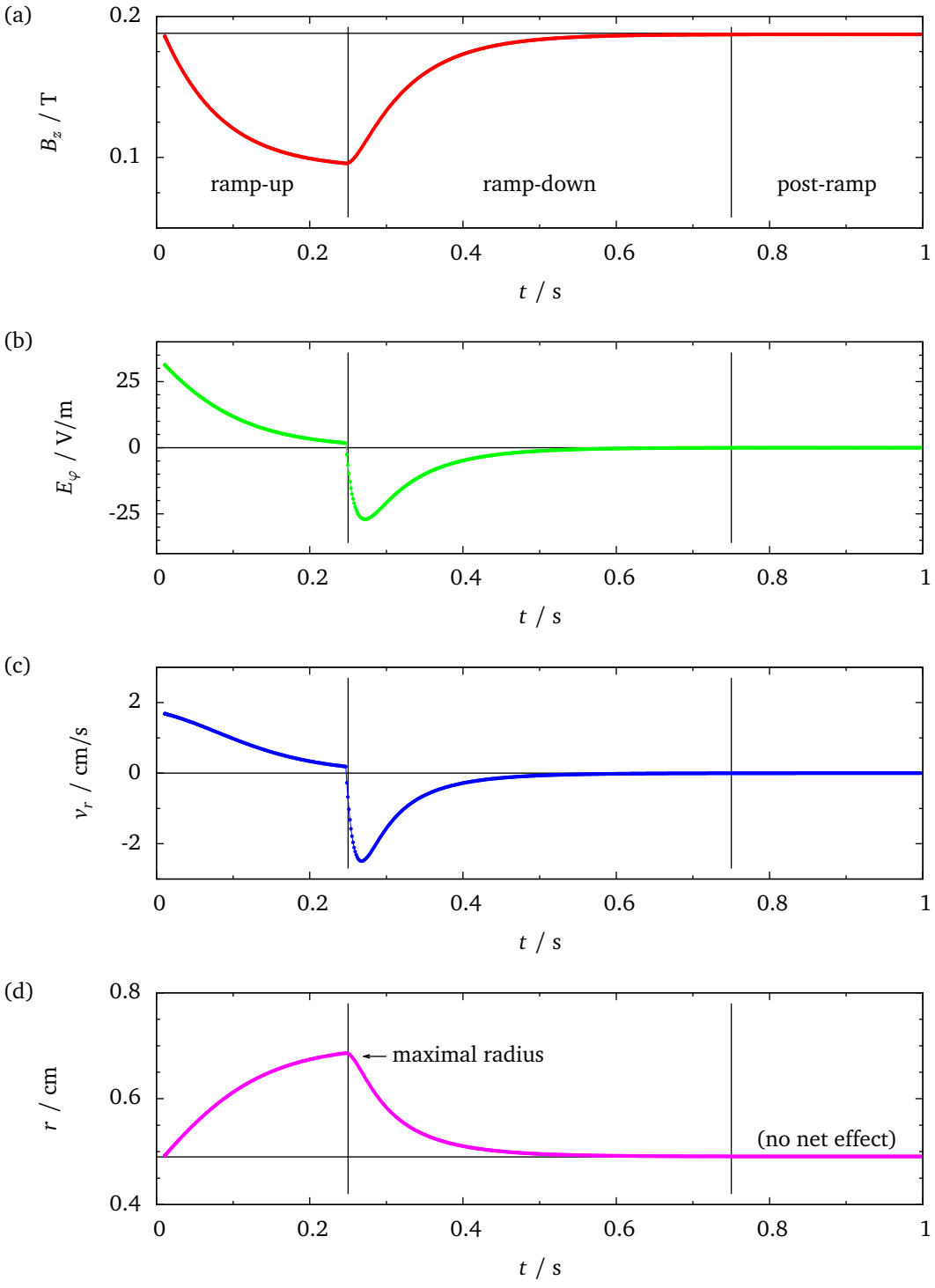


Figure 6.20.: Effect of the pulsed coil on stored electrons. The four plots show the axial magnetic field, azimuthal electric field, radial drift, and trajectory radius of a stored electron while the compensation coil is ramped up and down. The shown data points correspond to the turning points of the electron's trajectory – where the electron is reflected in each axial oscillation period – at a position of $z \approx -2.89\,\mathrm{m}$. The $E \times B$ drift was computed from the shown electric and magnetic fields; the other data is direct output from the simulation.

.

c) The third plot shows the radial $E \times B$ drift velocity of the stored electron. This data is not taken directly from Kassiopeia, but computed from the shown magnetic and electric field components according to (6.24). The reason is that Kassiopeia has no output field which shows only the $E \times B$ drift component; it would only be possible to compute the total radial drift, which includes other effects like the previously mentioned widening of the flux tube.

d) The last plot shows the radial position of the electron, which is affected by both the widening of the flux tube due to the reduced magnetic field in the ramping phase (compare previous section), and the radial drift induced by the magnetic and electric field. It can be seen that the radial movement is directed outwards in the ramp-up phase of the coil, and changes its direction in the ramp-down phase.

Comparing figure 6.20 to the validation results shown in figure 6.19, it is obvious that the Kassiopeia simulations closely match the expected results. The magnetic field of the pre-spectrometer solenoid is decreased by about 50 % at the examined axial position due to the up-ramping of the coil. The changing magnetic field induces an azimuthal electric field, and its direction changes in the two ramping phases. The radial $E \times B$ drift follows this course, as it depends on the azimuthal electric field. Interestingly, the shape of the drift curve is different in the two ramping phases, but of the same magnitude. The different shape can be explained considering the last plot, which shows the radius of the electron's trajectory (still, at an axial position of about $-2.9\,\mathrm{m}$). The curve has the same shape as the one from the validation output, but its amplitude is much larger: The radius increases by about 40 %, whereas the validation output only shows an increase of a few percent. This is because the Kassiopeia simulation includes both the radial $E \times B$ drift and the widening of the flux tube by the decreased magnetic field. The validation program, on the other hand, only shows the drift effect. With this in mind, the shape of the drift curve can be explained by the large increase in radius: According to equation (6.26), the exact drift speed depends on the electron's distance from the z-axis. Therefore, the drift speed is different in the two ramping phases since the trajectory radius is different. In addition, there is a slight radial dependency of the magnetic field, which also contributes to this effect.

Another important result can also be retrieved from figure 6.20: The net-effect that was assumed to exist prior to the simulations is not found in the collected data. Such a net-effect would result in an altered trajectory radius after one coil pulse, which would be visible in the simulation output. As can be seen in the figure, this is not the case, and the final and initial radii are almost exactly the same. This is also shown for in table 6.1 for a number of different starting parameters. For some of the starting radii, a negative (i. e., inwards) net-drift is observed in the simulations. It should be note that this is not excluded by the above assumptions; however, the magnitude is far too small either way to have any significant impact on the electrons' trajectories.

To explain the increased efficiency of the coil with this "net-drift effect", the change in radius would at least have to be in the order of millimeters. This can be estimated by assuming that the coil pulse has to increase the radius in a way that the increase accumulates over multiple pulses. This would move electrons onto other trajectories, from where they would be removed eventually by following pulses. Since a series of five to ten pulses was shown to remove almost all electrons from the trap, a net-effect of some millimeters per pulse would be necessary to

explain this behavior (considering that the radius of surrounding electrodes near the coil is about 7 cm).

E_0 / eV	Δr / mm			
1000	4×10^{-5}	-7×10^{-2}	-7×10^{-2}	-0.3
2000	4×10^{-5}	-7×10^{-2}	-7×10^{-2}	-0.3
3000	4×10^{-5}	-7×10^{-2}	-7×10^{-2}	-0.3
4000	4×10^{-5}	-7×10^{-2}	-7×10^{-2}	—
5000	4×10^{-5}	-7×10^{-2}	-7×10^{-2}	—
6000	4×10^{-5}	-7×10^{-2}	-7×10^{-2}	—
7000	4×10^{-5}	-7×10^{-2}	-7×10^{-2}	—
8000	4×10^{-5}	-7×10^{-2}	—	—
9000	4×10^{-5}	—	—	—
10000	—	—	—	—
$r_0 =$	0.1 mm	1 mm	2 mm	5 mm

Table 6.1.: Change in radial position due to a single coil pulse. The table shows the net-change in the electron's radial position after one pulse of the coil. The shown entries correspond to the outermost turning-points of the trajectory (i. e. closest to the coil); entries marked with a dash correspond to tracks were the electron was removed from the trap, and therefore the net-change can not be measured. The four columns correspond to the four different starting radii which were investigated (r_0, shown in last line), while the rows correspond to the initial kinetic energies (E_0). It can be seen that the net-effect of the pulsed coil on the electrons' radius is in the range of tenths of millimeters, and therefore not strong enough to accumulate over multiple pulses of the coil.

Further simulations with variations in the relevant starting parameters, such as kinetic energy, radial and axial position, or polar angle, confirmed these results. Changes in the coil settings, e. g. longer or shorter ramping phases, also showed no such net-effect. This only leaves the conclusion that such a net effect does in fact not exist, at least not in a way that it can be seen in the performed Kassiopeia simulations. Therefore, the high efficiency of multiple coil pulses can not be explained by the simulations that are presented here. This leaves the conclusion that the simulation must be missing one or more effects on stored electrons, whose implications are yet unknown and could result in a shift of the electrons' trajectories due to a coil pulse. Some possible explanations for this behavior will be given in the next section.

It was mentioned earlier that the performed simulations do not include any interactions between the electrons, which could be an effect that is missing here. The large number of stored electrons could also lead to the build-up of a negative space-charge in the trap region, which in turn might have an impact on the electrons' trajectories. However, such simulations would require a new implementation of a particle tracking application, or at least a thorough redesign of the Kassiopeia software package. Both of these options were not feasible in the context of this diploma thesis.

6.4.6. Electron removal with the pulsed coil

Although the assumed net-effect of the coil-induced radial drift could not be confirmed by the simulations, it was still considered useful to further investigate the coil's effect on stored electrons. Even without a net-drift, electrons on outer trajectories are forced against the surrounding electrode surfaces, which removes them from the trap volume. With the knowledge gained by previous simulations, it is fair to assume that the removal efficiency should depend on various parameters of the electrons, i. e. their initial kinetic energy.

To study the electron-removal, several simulations of single electrons were performed with settings equivalent to the ones described above, i. e. with energy losses due to synchrotron radiation and scattering on H_2 molecules, and a time-scaling factor of 10^4. As each simulation took a computation time of roughly one day, the number of simulations and thus the number of investigated starting parameters had to be limited. Since it could be expected that the most significant impact would come from the initial kinetic energy, this parameter was varied from 1 keV to 14 keV in steps of 1 keV in each simulation run. These runs in turn were performed with four different radial starting positions, and each of those with initial polar angles of 0° or 60°. The total number of computed electron tracks is therefore about 100, resulting in a required total computation time of about 2000 h[16].

The results of these simulations are shown in figure 6.21 and 6.22. Figure 6.21 shows the coil's effectiveness in removing electrons in relation to the initial kinetic energy and the starting radius. In the plot, tracks where the electron was removed from the trap are marked blue, while tracks where the electron was not removed are marked red. It can be seen that electrons with large kinetic energies and/or large starting radii are removed by the coil.

The upper plot in figure 6.22 shows the lifetime of stored electrons that were removed by the pulsed coil. Electrons which were not removed are not shown in the plot; this is the reason why there are no data points at lower kinetic energies. The starting radius is marked by different colors as shown in the legend, while the two different starting angles are marked by different symbols. All electrons were started at the center of the pre-spectrometer solenoid to avoid magnetic mirroring (section 6.2), which would distort the simulation results.

It can be seen that electrons at higher kinetic energies are successfully removed by one pulse of the coil, and their lifetime decreases with higher energies. On the other hand, electrons at lower kinetic energies are not removed. This can be explained by the increased trap length for higher energies, as it was determined in section 6.2.1. Since the coil is located roughly in the same location as the main spectrometer electrode at $z \approx -3\,\mathrm{m}$, stored electrons are more affected by the down-ramping of the magnetic field if they get closer to the electrode, which is the case with higher kinetic energies. This also explains the decrease in lifetime: As the electrons get closer to the coil and thus experience a stronger effect from the coil, they hit the surrounding electrode earlier. In addition, a high polar angle leads to a "missing" kinetic energy in longitudinal direction, and thus lowers the trap length (which was also seen in section 6.2.1). The above explanations therefore also apply here, if one considers that the

[16] This number emphasizes the usefulness of the computing cluster available at the Institute for Nuclear Physics at WWU Münster, as otherwise it would not have been possible to perform that many long-term simulations in acceptable time.

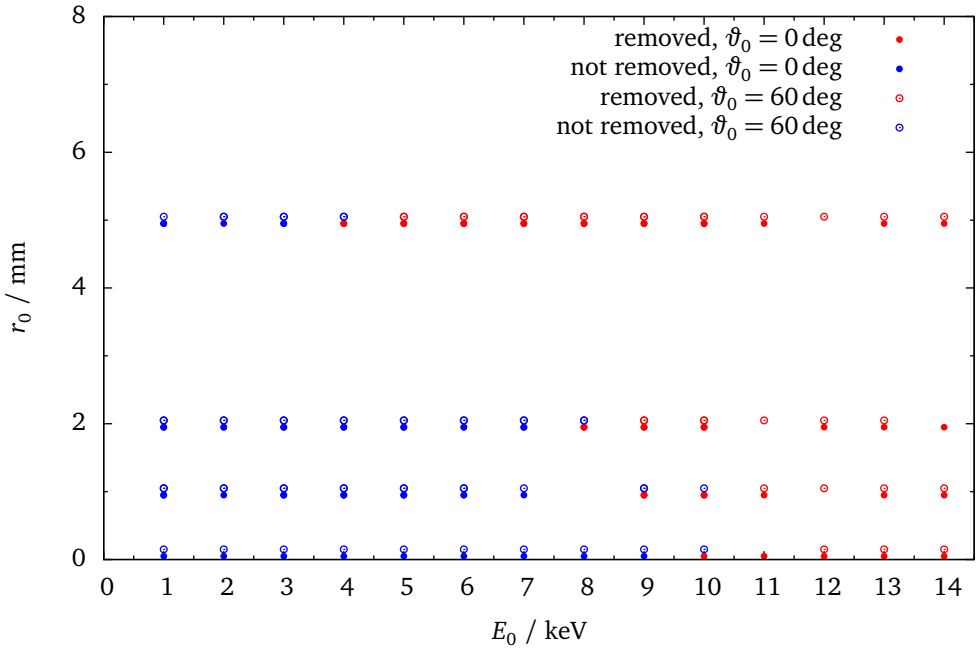


Figure 6.21.: Removal of stored electrons with the pulsed coil. The plot shows the results of tracking simulations where electrons were or were not removed. The plot uses the same data as figure 6.22, but shows the coil effectiveness in relation to the initial kinetic energy E_0 and the initial radial position r_0. The data points for the two initial polar angles $\vartheta_0 = 0°$ and 60° were vertically displaced for better visibility. It can be seen that there exists a certain "region" with low kinetic energies and low starting radii, where electrons are not removed by the coil. With higher energies and/or starting radii the coil works effectively. The boundary between these two regions also depends on the polar angle because of the effectively reduced energy and higher energy losses (see continuous text). (Note: The "missing" points correspond to failed simulation runs, i. e. runs which exited abnormally on the computing cluster.)

effective kinetic energy is reduced in case of $\vartheta_0 = 60°$. Additionally, a high polar angle leads to stronger energy losses. This will be further explained below.

The dependency on the starting radius directly follows from the observed electron trajectories, which are mainly determined by the initial radius. It is obvious that electrons on outer trajectories hit the electrode earlier than electrons that are closer to the z-axis, since they don't need to be moved outwards as much by the coil. Note that the starting radius also determines the minimum kinetic energy that is required for electrons to be removed. The reason is that the total increase in radius within one pulse of the coil correlates strongly with the kinetic energy because then the coil's effect increases. Electrons on inner trajectories therefore need to have higher kinetic energies to be forced against the surrounding electrodes.

The lower half of figure 6.22 shows the relative total energy losses using the same simulation data. Note that the plot uses a logarithmic scaling of the $\Delta E/E$-axis. For a low initial polar angle of 0°, the energy losses are in the order of less than 10 % of the total energy during the

whole lifetime, while for a polar angle of 60°, the energy losses are generally higher. This is due to the fact that mostly transversal energy is lost by synchrotron radiation, and the fraction of kinetic energy which contributes to transversal energy is determined by the polar angle (compare equation (3.2)). The relative total energy loss also decreases for higher energies, which can be explained by the shorter lifetime of such electrons.

While the energy losses due to synchrotron radiation depend only on the total length of the electron's flight path, the scattering losses are a result of random interactions with molecules from the residual gas. While elastic scattering leads to energy losses in the range of eV, and inelastic scattering[17] can result in losses of several keV. This can actually be seen in one simulation that is marked "secondary particle" in the plot. Here the total energy loss is about 2.5 keV at an initial energy of 6 keV. A closer investigation of the simulation data showed that a high-energetic secondary electron was created along the electron's track, resulting in the rather high energy loss.

Influence of scattering processes

In addition to investigating the coil's effect on stored electrons, additional studies were performed to investigate the influence of scattering processes. A better understanding of these processes could help to finally understand the increased efficiency of the pulsed coil when operated with multiple pulses.

As shown in figure 6.21, electrons with low starting angles are more likely to be removed by one ramp of the coil than those with higher angles. The reason is that with high polar angles the longitudinal energy is smaller, and the trap length is effectively reduced. In this case, the electron trajectories end in a greater distance to the coil, and the coil's influence on the electrons is not as large. A scattering process, however, would be able to transform a part of the transversal component of kinetic energy into the longitudinal component ($E_\perp \rightarrow E_\parallel$). Although the transversal energy is eventually lost by synchrotron radiation, this process could enable electrons that were not removed by a previous pulse to gain additional longitudinal energy. In turn, these electrons would possibly be removed by a following pulse of the coil.

The investigation of scattering events allows to estimate the typical time-scale for scattering events to occur, which is shown in table 6.2. The simulations results from 100 tracks with a initial polar angle of 0° show that 27 scattering events took place, and 15 of these resulted in the creation of secondary electrons (i. e. ionization occured). Taking the total simulation time of all tracks into account, this results in a mean scattering rate of roughly $0.5\,\mathrm{s}^{-1}$ at a pressure of 10^{-9} mbar. The mean time-scale for scattering is then $\tau = 1.9\,\mathrm{s}$. Also interesting are the energy losses of the examined electrons: The mean total energy loss due to scattering is only 8.5 eV, while the energy loss due to synchrotron radiation is 654 eV. It should be noted that with elastic scattering, the energy transfer is small and no secondary electrons are produced; therefore the number of scattering events does not include elastic scattering processes.

[17] The first ionization energy of H_2 is 15.4 eV.

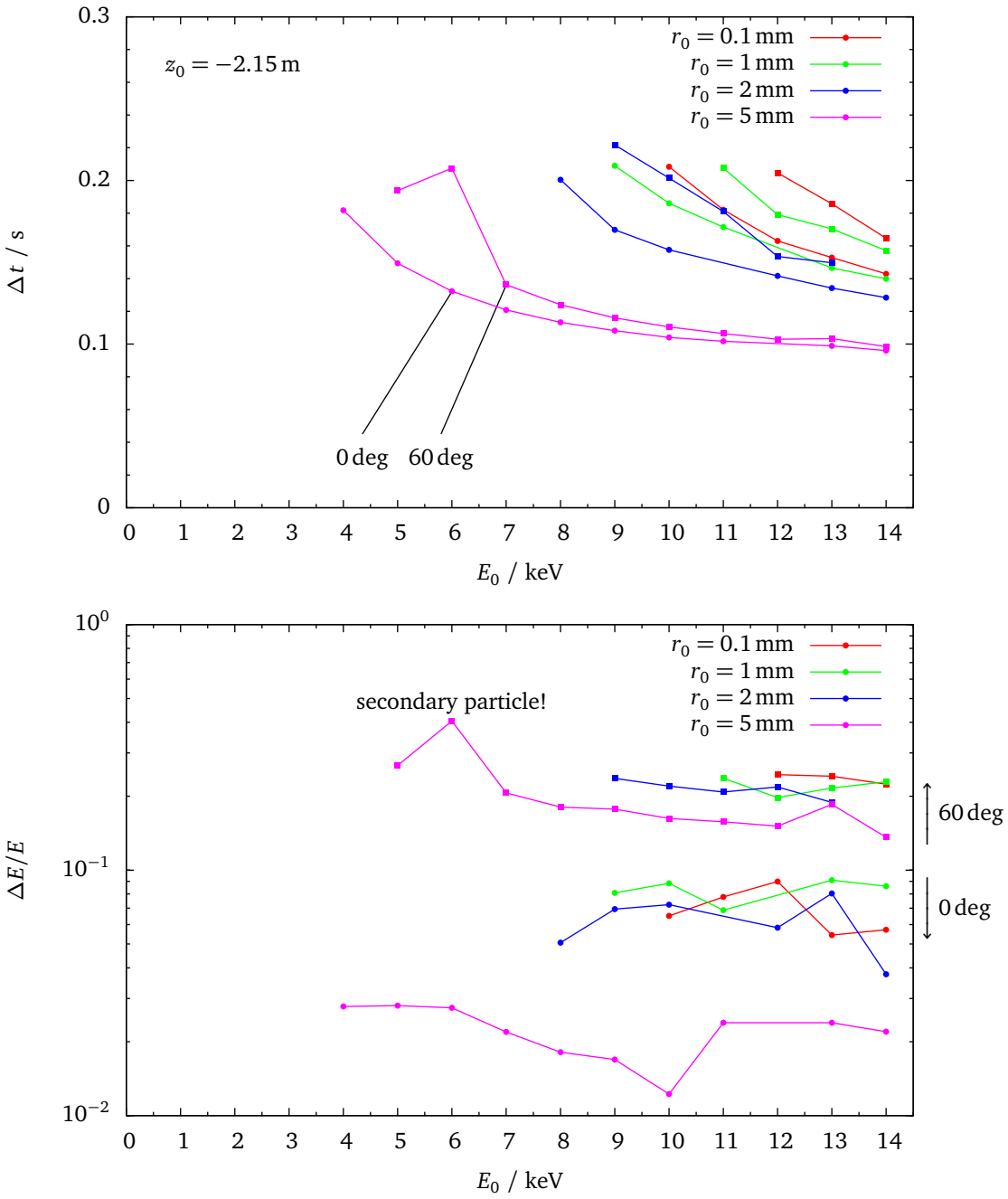


Figure 6.22.: Removal of stored electrons with the pulsed coil. The plots show the results of the pulsed coil simulations, where a single electron was simulated for 1 s real-time in each run (with ramp-up/ramp-down phases of 0.25 s and 0.50 s). In the upper half the storage times of electrons are plotted against their initial kinetic energy. Electrons that were not removed by the pulsed coil (i. e. at lower energies) are not shown in the plot. Each simulation was done with an initial polar angle of 0° (circles) and 60° (squares), respectively. The lower half shows the total energy loss of the electrons within the given time. The main process is synchrotron radiation because of the relatively short storage times (< 0.5 s) and therefore low scattering probability. However, a small contribution to the energy losses also comes from scattering. In the marked run the high energy loss results from a scattering event which created a secondary electron.

		$E_0 = 1\,\mathrm{keV} to 14\,\mathrm{keV}$	
		$\vartheta_0 = 0°$	$\vartheta_0 = 60°$
number of tracks	N	100	79
number of inel. scattering processes	n_{sca}	27	49
number of secondaries	n_{sec}	15	24
total simulation time	$\sum t$	50.5 s	53.7 s
mean total energy loss	$\langle \Delta E \rangle$	663 eV	2222 eV
mean synchrotron energy loss	$\langle \Delta E_{syn} \rangle$	654 eV	2180 eV
mean scattering energy loss	$\langle \Delta E_{sca} \rangle$	8.54 eV	42.3 eV
mean scattering rate	$\langle \Gamma_{sca} \rangle$	$0.535\,\mathrm{s}^{-1}$	$0.912\,\mathrm{s}^{-1}$
mean scattering time	$\langle \tau_{sca} \rangle$	1.87 s	1.10 s

Table 6.2.: Inelastic scattering processes of stored electrons. The table shows the results of a simulation of 179 electron tracks, split up into two sections for low and high initial polar angles ϑ_0 as noted in the first line. It is apparent that the energy losses are mainly due to synchrotron radiation, and only a small fraction of energy is lost in scattering processes due to the low pressure of $p = 10^{-9}$ mbar. The energy loss due to scattering includes elastic scattering as well as excitation and ionization. The observed number of secondary electrons created allows to estimate the typical time-scale of such scattering events. With the investigated starting angles, they are in the range of $\tau \approx 1\,\mathrm{s} to 2\,\mathrm{s}$ (see continuous text).

With the used simulation settings[18], it is not possible to retrieve the exact number of elastic scattering processes from the simulation output.

As mentioned above, scattering processes could result in a transfer of transversal to kinetic energy. Therefore, the simulation of scattering processes on H_2 in Kassiopeia was further investigated. Figure 6.23 shows the dependency on the cross-sections of elastic scattering, excitation and ionization as it is implemented in Kassiopeia. It can be seen that in the energy region that is covered by the simulations presented in this diploma thesis ($E > 1\,\mathrm{keV}$), ionization is the dominant process. Elastic scattering, on the other hand, is suppressed. Additionally the total cross-section decreases with higher kinetic energies, there the scattering probability is further reduced for high-energetic electrons, i. e. with $E \leq 10\,\mathrm{keV}$.

Figure 6.24 shows the energy-loss spectrum of electrons scattering on H_2 molecules. It can be seen that the excitation process has two major peaks[19] at 13 eV and 15 eV, and is limited to a range of $11\,\mathrm{eV} to 15\,\mathrm{eV}$. Ionization has a peak at roughly 16 eV and the probability decreases for higher energy losses. The energy losses due to inelastic scattering are limited to the sub-eV-range, as almost no energy is transferred in these processes. However, all of these processes can change the momentum vector and thus the polar angle of primary electrons.

Finally, the changes in the electrons' polar angles due to the three scattering processes are also shown in figure 6.24. It can be seen that the angular changes due to excitation and

[18] Especially the setting of `StepIteration=100` makes it impossible to investigate each single step of the tracking simulation. This settings is needed, however, as otherwise the produced output files would grow too big with the number of simulation steps being in the 10^9-range for each track.

[19] Note that the plot uses a 0.5 eV-binning.

ionization are much smaller than those which correspond to elastic scattering. Consequently, elastic scattering should be the main process that is able to significantly change the electrons' momentum vector, and therefore transfer large fractions of transversal into longitudinal kinetic energy. The figure also shows that ionization can contribute to such changes in the electrons' momentum, as both processes show a tail towards higher angular changes, and in fact have the same probability for high angular changes above roughly 30°. This result implies that inelastic scattering processes could indeed play a major role in changing the electrons' momenta. This idea is further supported by the higher cross-section for ionization in the relevant energy range above 1 keV.

In addition, the angular spectrum depends strongly on the kinetic energy, i. e. the angular changes are shifted towards smaller angles for higher energies. As the energy of stored electrons changes along their trajectories and is also reduced by synchrotron radiation, the influence of scattering processes can also change over time. The polar angle of stored electrons could therefore still be affected by both elastic and inelastic scattering processes. However, the exact implications of these result remain unclear, and a further investigation is suggested.

It is also possible to estimate the typical scattering time from the following equation [Ess04]:

$$\tau(E) = \frac{1}{v \cdot N \cdot \sum \sigma} \sim \frac{1}{vp}, \tag{6.41}$$

where v is the electron's velocity and p is the residual gas pressure. With $p = 10^{-9}$ mbar and an electron energy of 10 keV, the scattering time results to $\tau \approx 0.5$ s using the above formula. This result is smaller than the results gained from simulations (1 s to 2 s). One reason could be the large energy loss due to synchrotron radiation, which effectively reduces the available kinetic energy in relatively short time, therefore increasing the value of τ.

The time-scale of a *full* ramping of the coil is of the order of 1 s, where the ramp-up phase alone takes only 0.25 s. In the experimental studies of B. Hillen, the pauses between multiple pulses of the coil were also in the range of several seconds. The total time between two pulses is therefore larger than the typical time-scale for scattering. On the other hand, the loss of transversal energy due to synchrotron radiation also has a time-range of several seconds.[20] It is obvious that once all transversal energy is lost by this effect, it can no longer be transformed into longitudinal energy by the aforementioned scattering processes. Table 6.2 shows that the total synchrotron losses of stored electrons are fairly high compared to the losses by scattering in the investigated time-range of ≤ 1 s.

Although it seems possible for stored electrons to transfer transversal into longitudinal kinetic energy by elastic scattering processes, the exact impact on scattering processes on the stored electrons need to be investigated more closely in the future. Furthermore it should be noted that the rather small number of 179 simulated electron tracks, which resulted in a total of 79 inelastic scattering events, does not allow to provide an exact estimation of the scattering time-scales. Therefore, the existing data is not fully excluding the effect discussed above. The

[20] With $\Gamma \approx -8\,\mathrm{s}^{-1}$, an electron with a transversal energy of 18 keV would need $\tau \approx 1$ s to loose all its transversal energy by synchrotron radiation in a magnetic field of 4.5 T [Hil11]. The resulting time-scale is somewhat larger due to the inhomogeneity of the magnetic field, see also section 4.3.

long time required to compute a single electron track prevented to perform more simulations within the course of this diploma thesis.

In conclusion, further simulations and a detailed analysis of the mentioned effects are needed to create more representative results. However, the results presented here could be a first hint to finally understand the increased efficiency of the coil when multiple pulses are applied, as it was seen in the previous experiments by B. Hillen.

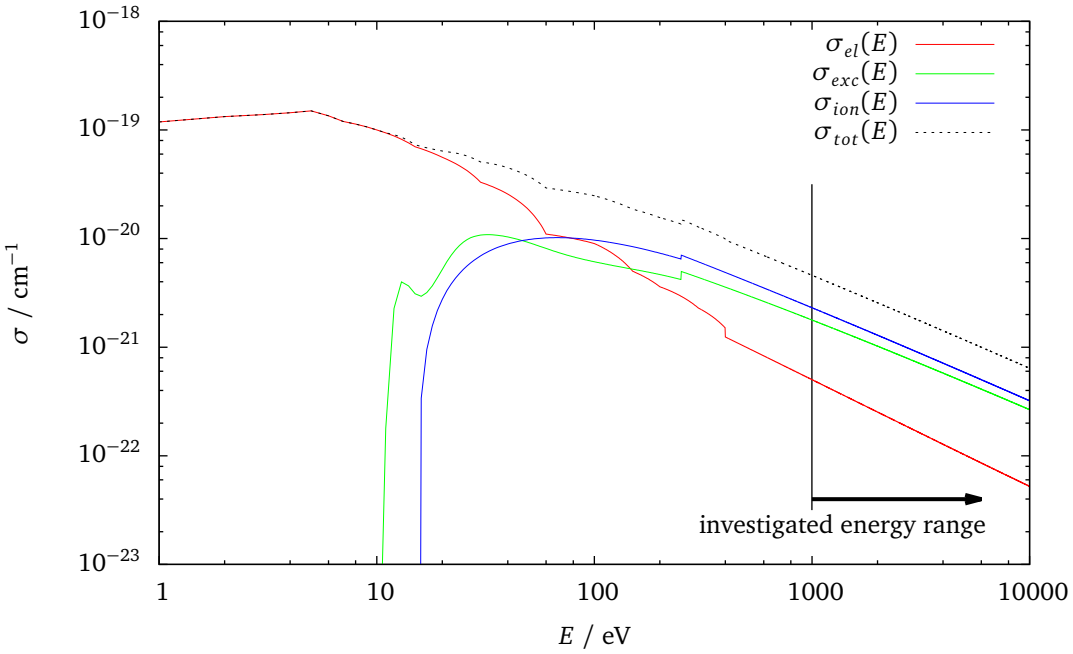


Figure 6.23.: H_2-scattering in Kassiopeia. The plot shows the energy dependency of the cross-sections for elastic scattering, excitation and ionization on H_2, together with the total scattering cross-section. The results were obtained using calculations by FERENC GLÜCK, which are also implemented in Kassiopeia to simulate scattering processes [Glu12]. In the energy range that was investigated in the presented simulations (i. e. at $E \geq 1\,\mathrm{keV}$), ionization is the dominant process.

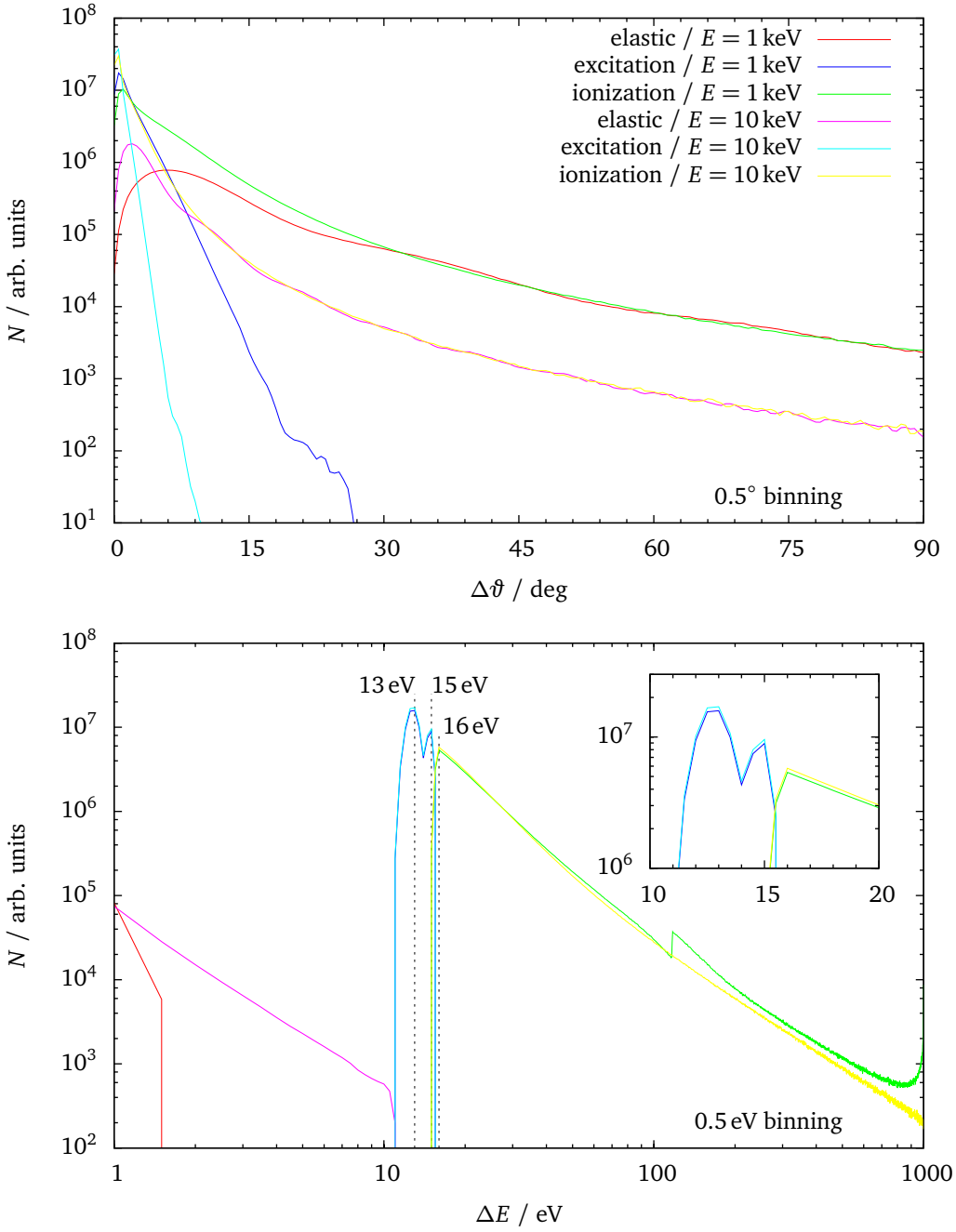


Figure 6.24.: H_2-scattering in Kassiopeia (continued). The first plot shows the change in the polar angle of scattering electrons for two kinetic energies of 1 keV and 10 keV, and for the three scattering processes. The second plot shows the corresponding energy-loss spectrum for the same two kinetic energies. The spectrum shows three peaks at 13 eV, 15 eV and 16 eV (also shown in the inset). The first two peaks correspond to excitation, the last to ionization. The shown spectra were obtained by simulating 10^8 scattering processes using the aforementioned calculations (see figure 6.23). Both plots use the same coloring scheme.

7. Summary

The goal of the KATRIN experiment is to determine the absolute mass of the electron anti-neutrino. This will be achieved by precisely measuring the endpoint-region of the Tritium beta-decay spectrum. With a sensitivity of 0.2 eV at 90% C.L., the experiment will improve the sensitivity of previous kinematic measurements by one order of magnitude.

A MAC-E filter is employed to analyze the beta-spectrum with high precision and large angular acceptance, both key features to a successful measurement. The MAC-E filter is based on a combination of magnetic collimation and an electric retarding potential, and functions as a high-resolution integrating spectrometer. To pre-select the interesting electron in front of the main spectrometer, an additional MAC-E filter is employed as a pre-spectrometer. Although this setup aims at reducing the background rate by lowering the rate of electrons that enter the main spectrometer, it also results in an intrinsic Penning trap between the two spectrometers. As this trap is located directly in the beam line of the decay electrons, it can disturb the measurement and create a large amount of background from trapped electrons.

The Penning trap between the spectrometers was previously investigated in several diploma and Ph.D. theses. Experimental studies showed that the Penning trap would indeed pose a threat to the measurement in terms of additional background. The three proposed counter-measures were all found to be effective in removing electrons from the trap, and therefore prevent the creation of background. However, these previous studies left open a number of questions concerning the exact mechanisms of the electron-removal methods. Additionally, the exact characteristics of the Penning trap can not be estimated easily due to the complex setup of the trap, and were therefore not known exactly beforehand. It should be noted that the pulsed coil was not meant to be applied at the spectrometer junction in the final KATRIN setup due to space restrictions. However, the studies of the pulsed coil served as a test-case for a possible application of the same principle at the main spectrometer.

In this diploma thesis, tracking simulations of stored electrons were performed to further understand the exact storage conditions of the trap and the behavior of stored electrons. Additionally, two of the proposed counter-measures – the electron catcher and the pulsed

compensation coil – were investigated. The simulations were performed using the particle tracking software Kassiopeia, which was developed in the KATRIN collaboration. With Kassiopeia it was possible to simulate flight paths of electrons through electromagnetic fields. In order to simulate the aforementioned electron-removal methods, the Kassiopeia package had to be extended, most notably by implementing the time-dependent magnetic and electric fields related to the pulsed coil. The simulation results presented here can be divided into three sections:

- An analysis of the motion of trapped electrons allowed to investigate the dimensions of the trap and its dependency on the electrons' parameters, and to measure the frequencies of the three components of motion (cyclotron motion, axial oscillation, and magnetron drift).
- Simulations of the electron catcher lead to a better understanding of the electron catcher's effect on stored electrons. It was shown that the storage times mainly depend on the magnetron frequency, and that the electron catcher is able to remove all electrons from the trap if it is moved into the center of the flux tube.
- The simulations of the pulsed coil were successful in providing insights on the behavior of electrons moving around in the trap while coil pulses are applied. It was shown that high-energetic electrons can be removed by single pulses of the coil; however, electrons with low energies are only removed if they are already on an outer trajectory in the flux tube. Furthermore, a proposed net-drift of electrons due to repeated pulses of the coil could not be confirmed by simulations, meaning that either such an effect does not exist or that some important effect is not yet included in the simulations.

  An alternative explanation could be the transfer of transversal to longitudinal kinetic energy by scattering processes. The simulations allowed to estimate the characteristic timescale of scattering processes to lie between the timescale of coil pulses and the timescale of the full loss of transversal energy due to synchrotron radiation. It would therefore be possible for electrons to gain longitudinal kinetic energy by scattering, and therefore be more affected by following coil pulses, resulting in the removal of such electrons from the trap. In conclusion, this idea could explain the high removal efficiency of multiple coil pulses that was found in previous experimental studies. It was shown that inelastic scattering processes could play a major rule in such transfers of kinetic energy. However, a closer investigation of this topic is needed to gain additional knowledge of the effect, and to determine its influence on stored electrons.

The results presented in this diploma thesis also show that Kassiopeia is suitable to simulate trapped electrons with high accuracy, even with the rather complex geometric setup that was used. Its modular concept allowed to extend parts of the code wherever necessary. In the future of the KATRIN experiment, the simulation software will become even more important, e. g. in terms of providing methods to analyze the measurement data.

Furthermore, a first implementation of visualization techniques was added to Kassiopeia. Visualization of the simulation geometry and the generated tracks was found to be very helpful in designing and verifying simulation setups, and in analyzing the output of the simulations. In future, the complexity of the simulations will likely increase. The geometry management software Kreator, which was also developed in this diploma thesis, will help users to construct simulation geometries from a KATRIN-global database.

A. KARMA example script

This is an example of a script file that uses the Kassiopeia Run Manager (see section 5.1.4 on page 51). The example will start three sub-runs sequentially, each using the same geometry and tracking configuration, but with different starting angles of the electron.

A.1. Script structure

The general structure is as follows: In the header part of the script, all of the requires library modules are loaded. These are either Ruby standard libraries, or part of KARMA and located in the `.lib` subdirectory. Thereafter (ll. 25ff.), all the objects needed for the simulation are created and configured. This scheme basically follows the structure of the Kassiopeia configuration files, e. g. a particle generator is defined by combining four particle creators, which in turn define starting position, direction, energy and time. After all these objects are configured, they have to be connected to a region which is then used by Kassiopeia. In the example, the basic `KatrinExperimentalHall` region is used together with another one that corresponds to the inner pre-spectrometer volume. Some of the step computer extensions are only used inside this inner region, e. g. the drift and gyration modules.

In the last part of the example script (ll. 141ff.), the simulation sub-runs are configured and executed. The script will start several runs in the range of given run ids (in this example, three runs are started). Inside the run loop (ll. 165ff.), the run id is used to define a different starting angle for each run. Finally one run is started in each iteration by the `config.start` function. This function will wait for a run to complete before starting the next one. It is also possible to start a run in the background and directly proceed to the next iteration, as can be seen in the following, commented-out line. This obviously only makes sense on computer systems with a multi-core processor, as otherwise the processes running in parallel will slow down each other. When working on a batch system, like the MAF cluster of the Institute for Nuclear Physics at

WWU Münster, the simulation runs can also be started on different nodes automatically. This is also included in the example script at ll. 230ff. and uses the Torque batch system.

A.2. Example script

Listing A.1: Karma-example.rb

```
#!/usr/bin/env ruby
#-*- coding: utf8 -*-
#
# Karma - the Kassiopeia Run Manager
#


require 'erb'
require 'ftools'

$LOAD_PATH << './.lib'
require 'generatorconfig'
require 'geometryconfig'
require 'stepstrategyconfig'
require 'fieldconfig'
require 'outputconfig'
require 'userconfig'
require 'kassiopeiaconfig'

# Set to true if you don't want fancy colored output
require 'colors'
$DisableColors = false


################################################################################

# In this section all Karma elements which are needed later should be defined.
# This applies to Electric/Magnetic fields, Step Computers etc.



## Geometry Config - see lib/geometryconfig.rb for details

# At least one root geometry is needed. By default, the ConeGeometry element
# creates a geometry which is 40 m long and 20 m wide.
experimentalHall        = ConeGeometry.new("KATRINExperimentalHall")

# You can also define other conical geometries
prespecVolume          = ConeGeometry.new("PrespecVolume")

prespecVolume.Z1           = -1.35
prespecVolume.Z2           =  1.35
prespecVolume.R1           =  0.80
prespecVolume.R2           =  0.80


## Field Config

# The Elcd 3.2 prespec electrodes will be used in this example,
# together with the standard prespec magnets.
prespec_electrodes      = "input_prespec_full_half.el32"
prespec_coils           = "ps_magnet.mag3"

prespecElcd32           = KAFCAElfield32Ferenc.new("PrespecElfield32", prespec_electrodes)
prespecMag3             = KAFCAMagfield3Ferenc.new("PrespecMagfield3", prespec_coils)


```

```
58
59 ## Generator config - see lib/generatorconfig.rb for details
60
61 # To define a particle generator, you first need to set up some particle
62 # creators. These are independent of each other and can then be combined.
63 # Note that all attributes can be set later, which is used in this example
64 # to create a couple of runs with variable particle direction.
65 fixedPosition           = PAGEPositionCreatorFix.new("myPosFixed")
66 fixedDirection          = PAGEDirectionCreatorFix.new("myDirFixed")
67 equalTime               = PAGETimeCreatorEquidistant.new("myTimeEqidistant")
68 fixedEnergy             = PAGEEnergyCreatorFix.new("myEnergyFix")
69
70 # The four creators are combined into a particle generator here.
71 defaultGenerator        = PAGEGenerator.new("Default", fixedPosition, fixedDirection,
       ↪ equalTime, fixedEnergy)
72
73
74
75 ## Step Stragety Config - see lib/stepstrategyconfig.rb for details
76
77 # This example uses the adiabatic step computer
78 # If no step computer is defined, Karma will fall back to the exact step cpmputer.
79 asc                     = AdiabaticStepComputer.new("ASC")
80
81 # We also use the Gyration, Drift and Synchrotron module in this example
82 gyration                = Gyration.new("Gyration")
83 drift                   = Drift.new("Drift")
84 synchrotron             = Synchrotron.new("Synchrotron")
85
86 # ASC is used together with StepSizeEnergy and StepSizeSynchrotron
87 # Note that all these have their own parameters, which are set to
88 # reasonable default values if not specified.
89 energyStep              = StepSizeEnergy.new("StepSizeEnergy")
90 synchrotronStep         = StepSizeSynchrotron.new("StepSizeSynchrotron")
91
92 # These two exit conditions will be used together
93 zPosition               = ExitConditionZPosition.new("ZPosition")
94 maxSteps                = ExitConditionMaxSteps.new("MaxSteps", 25000)
95
96
97
98 ## Output config - see lib/outputconfig.rb for details
99 # NOTE: Not implemented since the OutputToolbox is still under construction
100
101
102
103 ## User config - see lib/userconfig.rb for details
104
105 # The UserConfiguration is used to set up debug messages,
106 # disable toolboxes or specify config and data paths for Kassiopeia.
107 userConfig              = UserConfiguration.new("Track")
108
109
110
111 ## Global config - see lib/kassiopeiaconfig.rb for details
112
113 # The global configuration defines regions and connects single modules
114 # like step computers to these. At least one region is needed.
115 rootRegion              = KassiopeiaRegion.new("ROOT", experimentalHall, asc)
116 prespecRegion           = KassiopeiaRegion.new("PrespecRegion", prespecVolume, asc)
117
118
119 # Add step modules defined above
120 rootRegion.addStepControl(gyration)
121 rootRegion.addStepControl(energyStep)
122 rootRegion.addStepControl(maxSteps)
123
124 # Some of these step controls are only used inside the prespec region
```

```
125 prespecRegion.addStepControl(gyration)
126 prespecRegion.addStepControl(maxSteps)
127 prespecRegion.addStepControl(drift)
128 prespecRegion.addStepControl(synchrotron)
129 prespecRegion.addStepControl(synchrotronStep)
130
131 # Add E/M fields defined above
132 rootRegion.addElectricField(prespecElcd32)
133 rootRegion.addMagneticField(prespecMag3)
134
135
136 # Add prespec region to the root region
137 # You *must* call this after defining all step computers, fields etc.
138 rootRegion.addRegion(prespecRegion)
139
140
141 ################################################################################
142
143 # Here the elements defined above are used to create some Kassiopeia runs
144 first_run   = 1
145 last_run    = 3
146
147 # Number of Events to create per run (!)
148 num_events  = 1
149
150
151 # Run prefix is created automatically by the script filename,
152 # e.g. Karma-example.rb would give "example" as prefix.
153 scriptname = File.basename($0, File.extname($0))
154 tmp = scriptname.split("-", 2)
155 if tmp.length == 2
156     prefix = tmp[1]
157 end
158
159 # We use a loop here to create a number of different runs in thsi example
160 # The Run ID which will also be used to get the name of the output directory
161 # Karma puts all run-related files into a directory named ./<prefix>/<run_NNN>/
162 # (or just ./<run_NNN> if no prefix was defined).
163
164 num_runs = 1 + last_run - first_run
165 (first_run..last_run).each do |run_id|
166
167     puts bold(blue( "Processing Run ID %03d" % run_id ))
168
169
170     # Here some attributes of the previously created Karma objects are set,
171     # the particle direction is set to a different angle for each run.
172     fixedPosition.Position      = [ 0.01, 0, -2.50 ]        # in front of the prespec magnet
173     fixedDirection.Theta        = (run_id-1) * 15           # gives 0, 15, 30 deg int his
        ↪ example
174     fixedEnergy.Energy          = 20000                     # rather high electron energy
175
176     # Kassiopeia will look for any required files first in the current working
177     # directory and the in the OptionHome and DataHome paths.
178     # B y default these are set to $KASSIOPEIASYS/etc and $KASSIOPEIASYS/Data,
179     # respecitvely, and can be influenecd by setting the KASSIOPEIASYS environment
180     # variable. If the variable is not set, the current directory where the
181     # KARMA script is executed will be used.
182     # It is also possible to set both paths by yourself here. In this case you
183     # must make sure that all needed files can be accessed by Kassiopeia, e.g.
184     # by copying over geometry input files into the directory.
185     #userConfig.OptionHome = "./"
186     #userConfig.DataHome   = "./"
187
188     # The KassiopeiaConfiguratrion class merges all the other elements,
189     # creates the Kassiopeia config files from the templates and is responsible
190     # to start the simulation.
191     config                      = KassiopeiaConfiguration.new(run_id, prefix, userConfig)
```

```
192     config.NEvents              = num_events
193
194     # Additional files which are needed inside the run directory can be given
195     # here. This can be used to copy files specific to a single run into the
196     # corresponding run directory. (Kassiopeia will prefer local files to those
197     # found in OptionHome or DataHome; see above.)
198     config.addRunFiles($0)    # Example: Copy the script itself into the run directory
199
200     # This method sets up the run directory and creates the config files
201     # YOU CAN'T CHANGE ANY CONFIG SETTINGS AFTER THIS LINE!
202     config.setup
203
204     # This will initialize the random generator of Kassioepia.
205     # By default, a constant seed of 54321 is used. If you want to get
206     # statistics data from a couple of runs, it is very important to use
207     # a different seed for each run! This can be achieved by this method, which
208     # uses the system time in microseconds as seed.
209     #config.seed
210
211     # Execute the run script directly, which starts Kassiopeia on this machine.
212     # Note that all runs are processed sequentially, not in parallel.
213     # To process runs in parallel, run the start function with argument "true".
214     # This will start the runs sequentially in the background, and your system
215     # will distribute the processes among your CPUs. You can also use the
216     # batchSubmit method described below.
217     config.start          # start runs one after the other
218     #config.start(true)    # start runs in background
219
220     # Alternatively, you can submit the run script to a batch queue.
221     # The batchSubmit method uses qsub to submit the job. The job will be named
222     # "KARMA[<prefix>-<run_NNN>]". If no queue is given, the short queue is used.
223     # Skipping the second parameter will create a "dry run", i.e. no job will
224     # be submitted, but all necessary files are created. To actually submit a
225     # job, set the second parameter to true.
226     # You can also add a list of required resources. These will be passed on
227     # directly to the batch system, so that it can manage your job in a way
228     # to fulfill your requirements. This is most useful e.g. for number of cores,
229     # required memory or CPU time (walltime). Please see the manual of your batch
230     # system for more information on this topic.
231     #config.batchSubmit("batch_queue")                              # dry run, no submission
232     #config.batchSubmit("batch_queue", true)                       # submit jobs to queue
233     #config.batchSubmit("batch_queue", true, "walltime=86400")     # request extra resources
234
235     puts ""
236
237 end  # run_id
238
239 ## EOF ##
```

B. Implementation of the pulsed coil in Kassiopeia

In this appendix section the exact implementation of the pulsed coil – i. e., the ramped magnetic and the induced electric fields – in Kassiopeia is shown. Due to Kassiopeia's modularized structure, the classes presented here make use of a number of external functions, as well as the general structure of Kassiopeia's toolbox system.

The according C source files are shown in the following two code snippets. The corresponding header files and other Kassiopeia classes are not shown here. The full Kassiopeia source code including this implementation can be retrieved from the KATRIN SVN server[1].

[1] https://nuserv.uni-muenster.de/websvn/listing.php?repname=KATRIN&path=/Kassiopeia/trunk/

Listing B.1: KSLinearMagneticField.cxx

```
namespace Kassiopeia
{
    KSLinearMagneticField::KSLinearMagneticField()
    {
        fMagfield.SetXYZ(0,0,0);

        fAmplitudeType = "";
        fAmplitude.SetXYZ(0,0,0);
        fSourceMagfieldLabel = "";
        fSourceMagfield = 0;

        fRampType = "";
        fRampUpTime = 0.;
        fRampDownTime = 0.;
        fTimeConstant = 1.;
        fRampDelay = 0;

        fTimeScalingFactor = 1.;
        fEnhancementFactor = 1.;

        fMaxFieldFactor = 1.;
        fFullRampTime = fRampUpTime + fRampDownTime;
    }

    KSLinearMagneticField::~KSLinearMagneticField()
    {
    }

    Magfield* KSLinearMagneticField::SourceMagfield()
    {
        if (fSourceMagfield == 0) {
            // try to get field from toolbox
            KSFieldToolbox* Fields = static_cast< KSFieldToolbox* >(
                ↪ KSCoreManager::GetInstance()->GetDownstreamManager("Field") );
            fSourceMagfield = Fields->GetMagField(fSourceMagfieldLabel);

            if (fSourceMagfield == 0) {
                E = KSException::eFatalError;
                E << "field called <" << fSourceMagfieldLabel << "> has not been
                    ↪ registered!";
                CatchException(E);
            }
        }
        return fSourceMagfield;
    }

    TVector3 KSLinearMagneticField::GetField(const TVector3& p, const Double_t time)
    {
        Double_t t;
        TVector3 Amplitude;
        Double_t Modulation;

        t = time * fTimeScalingFactor;
        //cout << "At t = " << t << " (real: " << time << ")" << endl;

        Amplitude  = GetAmplitude(p, t);
        Modulation = GetModulation(t);
        //cout << "  Magnetic field Amplitude: " << Amplitude.Z() << endl;
        //cout << "  Magnetic field Modulation: " << Modulation << endl;

        fMagfield = Amplitude * Modulation * fEnhancementFactor;
        //cout << "Magnetic field is " << fMagfield.Mag() << " at time " << t << endl;

        return fMagfield;
    }

```

```
65  Double_t KSLinearMagneticField::GetPhi(const TVector3& p, const Double_t time)
66  {
67      return GetField(p,time).Dot(p);
68  }
69
70  Double_t KSLinearMagneticField::GetModulation(Double_t time)
71  {
72      if (fRampType == "Linear")
73      {
74          Double_t field = 0.0;
75          if ( (fRampUpTime > 0.) && (time >= fRampDelay) && (time <
                ↪ fRampUpTime+fRampDelay) ) {
76              field = ((time-fRampDelay) / fRampUpTime);
77          }
78          else if( (fRampDownTime > 0.) && (time >= fRampUpTime+fRampDelay) && (time <
                ↪ fFullRampTime+fRampDelay) ) {
79              field = (1. - ((time-(fRampUpTime+fRampDelay)) / fRampDownTime));
80          }
81          return field;
82      }
83      else if (fRampType == "Exponential")
84      {
85          Double_t field = 0.0;
86          if ( (fRampUpTime > 0.) && (time >= fRampDelay) && (time <
                ↪ fRampUpTime+fRampDelay) ) {
87              field = ( 1. - exp(-(time-fRampDelay) / fTimeConstant) );
88              // save the maximum achieved field for the ramp-down phase
89              // this implementation is correct since the field is strictly increasing in
                    ↪ ramp-up
90              if (fabs(field) > 0.) { fMaxFieldFactor = field; }
91          }
92          else if( (fRampDownTime > 0.) && (time >= fRampUpTime+fRampDelay) && (time <
                ↪ fFullRampTime+fRampDelay) ) {
93              field = fMaxFieldFactor * exp(-(time-(fRampUpTime+fRampDelay)) /
                    ↪ fTimeConstant);
94          }
95          return field;
96      }
97      else {
98          E = KSException::eFatalError;
99          E << "selected ramp type <" << fRampType << "> is unknown!";
100         CatchException(E);
101     }
102
103     return 0.;
104 }
105
106 Double_t KSLinearMagneticField::GetDerivatedModulation(Double_t time)
107 {
108     if (fRampType == "Linear")
109     {
110         Double_t field = 0.0;
111         if ( (fRampUpTime > 0.) && (time >= fRampDelay) && (time <
                ↪ fRampUpTime+fRampDelay) ) {
112             return (1. / fRampUpTime);
113         }
114         else if( (fRampDownTime > 0.) && (time >= fRampUpTime+fRampDelay) && (time <
                ↪ fFullRampTime+fRampDelay) ) {
115             return (-1. / fRampDownTime);
116         }
117         return field;
118     }
119     else if (fRampType == "Exponential")
120     {
121         Double_t field = 0.0;
122         if ( (fRampUpTime > 0.) && (time >= fRampDelay) && (time <
                ↪ fRampUpTime+fRampDelay) ) {
123             field = ( 1./fTimeConstant * exp(-(time-fRampDelay) / fTimeConstant) );
```

```
124             }
125             else if( (fRampDownTime > 0.) && (time >= fRampUpTime+fRampDelay) && (time <
                    ↪ fFullRampTime+fRampDelay) ) {
126                 field = fMaxFieldFactor * (-1./fTimeConstant *
                        ↪ exp(-(time-(fRampUpTime+fRampDelay)) / fTimeConstant));
127             }
128             return field;
129         }
130         else {
131             E = KSException::eFatalError;
132             E << "selected ramp type <" << fRampType << "> is unknown!";
133             CatchException(E);
134         }
135 
136         return 0.;
137     }
138 
139     TVector3 KSLinearMagneticField::GetAmplitude(const TVector3& p, Double_t /*time*/)
140     {
141         if ((fAmplitudeType == "SpaciallyConstant") || (fAmplitudeType ==
                ↪ "SpatiallyConstant"))
142         {
143             return fAmplitude;
144         }
145         /*
146         else if (fAmplitudeType == "StandingWave")
147         {
148             return
                    ↪ fAmplitude*TMath::Cos(KaConst::Pi()*p.Z()/(KaConst::C()/GetFrequency(p,t)));
149         }
150         */
151         else if (fAmplitudeType == "Magfield")
152         {
153             if (SourceMagfield() != 0) {
154                 return SourceMagfield()->GetField(p /*, time*/ );
155             }
156         }
157         else {
158             E = KSException::eFatalError;
159             E << "selected amplitude type <" << fAmplitudeType << "> is unknown!";
160             CatchException(E);
161         }
162 
163         return TVector3(0,0,0);
164     }
165 
166 } // end namespace Kassiopeia
```

Listing B.2: KSInducedAzimuthalElectricField.cxx

```
namespace Kassiopeia
{
    KSInducedAzimuthalElectricField::KSInducedAzimuthalElectricField()
    {
        fElfield.SetXYZ(0,0,0);

        fInducingMagfieldLabel = "";
        fInducingMagfield = NULL;

        fEnhancementFactor = 1.;
    }

    KSInducedAzimuthalElectricField::~KSInducedAzimuthalElectricField()
    {
    }

    KSLinearMagneticField* KSInducedAzimuthalElectricField::InducingMagfield()
    {
        if (fInducingMagfield == 0) {
            // try to get field from toolbox
            KSFieldToolbox* Fields = static_cast< KSFieldToolbox* >(
                ↪ KSCoreManager::GetInstance()->GetDownstreamManager("Field") );
            fInducingMagfield =
                ↪ dynamic_cast<KSLinearMagneticField*>(Fields->GetMagField(fInducingMagfieldLabel));
            if (fInducingMagfield == 0) {
                E = KSException::eFatalError;
                E << "field called <" << fInducingMagfieldLabel << "> has not been
                    ↪ registered!";
                CatchException(E);
            }
        }
        return fInducingMagfield;
    }

    TVector3 KSInducedAzimuthalElectricField::GetField(const TVector3& p, const Double_t
        ↪ time)
    {
        Double_t r, t;
        TVector3 aziDirection;
        TVector3 Amplitude;
        Double_t Modulation;

        KSLinearMagneticField *magfield = InducingMagfield();

        t = time * magfield->GetTimeScalingFactor();
        //cout << "At t = " << t << " (real: " << time << ")" << endl;

        r = p.Perp();
        aziDirection = 1./r * TVector3( -p.Y(), p.X(), 0. );
        //cout << "  Azimuthal vector: ";
        //aziDirection.Print();

        //// This would be the straight-forward implementation for the azimuthal unit vector:
        //TVector3 zDirection(0,0,1);
        //TVector3 rDirection( p.X(), p.Y(), 0 );
        //TVector3 aziDirection = zDirection.Cross(rDirection).Unit();
        //cout << "  Azimuthal vector: ";
        //aziDirection.Print();

        Amplitude  = magfield->GetAmplitude(p, t);
        Modulation = magfield->GetDerivatedModulation(t);
        //cout << "  Magnetic field Amplitude: " << Amplitude.Z() << endl;
        //cout << "  Magnetic field Modulation: " << Modulation << endl;
        TVector3 Magfield = Amplitude * Modulation; // *
            ↪ fInducingMagfield->GetTimeScalingFactor()

```

```
62          fElfield = aziDirection * Magfield.Z() * (-r/2.) * fEnhancementFactor;
63          //cout << "  Electric field: " << fElfield.Mag();
64
65          return fElfield;
66      }
67
68      Double_t KSInducedAzimuthalElectricField::GetPhi(const TVector3& p, const Double_t time)
69      {
70          return -1.*GetField(p,time).Dot(p);
71      }
72
73  }
```

C. Simulation geometry

C.1. Related software

For the creation of the electrode geometry, the Python-based builder program developed by S. VÖCKING was employed. [Voe08] It provides a scriptable interface to create and place electrode elements, and allows to write the defined elements to a Elcd-compatible file. The used script file is not shown completely here, as a large part consists of function definitions that are not relevant in this context. Instead, only the electrode parameters and the setup of the single electrode elements is shown. The file was taken over from Hillen, who used it in his PhD thesis [Hil11].

The pre-spectrometer geometry, which is not included in the script, was taken from the file `input_prespec_full_half.el32` included in Kassiopeia. In order to merge both geometry files manually, the ground electrode in the original pre-spectrometer geometry had to be changed accordingly to not overlap with the ground electrode already defined in the test setup geometry. A sample of the resulting geometry file is given in the next section on page C7. Figure C.2 shows the geometries as used for the simulations.

Listing C.1: builder.py (partial)

```
1  from linalg import Vector, matrix_rotation_y, matrix_rotation_z, matrix_scale
2  from electrodes import Group, Rect, Wire, Cone
3  from tooth import Tooth
4  from math import atan2, cos, hypot, pi, sin, sqrt, tan, asin, acos
5  from ring import Ring
6
7  class Builder:
8      """This class does the actual building"""
9      def __init__(self):
10         self.mainspec   = None
11         self.scale      = 10
12         self.power      = 2
13         self.nrot       = 70
14         self.bigscale   = 40
15
```

```
    def build(self, filename):
        """Build a geometry from the given parameters and write to filename"""
        testkammer_setup = Group()
        wires   = Group()
        scale   = 80
        power   = 2
        nrot    = 70

        # Daten

        #Position vom Flanschende
        end_spektrometer_zachse = -1.89
        end_spektrometer_xachse = 0.1

        #Daten fuer das Uebergangstueck
        radius_uebergang    = 0.1
        length_uebergang    = -0.6055   #hier stand vorher 0.4055
        potential_uebergang = 0.0
        base_uebergang = Vector(0,0,end_spektrometer_zachse)

        #Daten fuer das Ventil
        reduzierventil_base = Vector(0.0, 0.0, end_spektrometer_zachse + length_uebergang)
        inner_radius_reduzierventil = 0.1
        outer_radius_reduzierventil = 0.125
        potential_reduzierventil    = 0.0

        radius_ventilzylinder = 0.125
        length_ventilzylinder = -0.150
        base_ventilzylinder   = Vector(0.0, 0.0, end_spektrometer_zachse + length_uebergang)

        #Daten fuer die Zusatzkammer
        #Daten fuer den Reduzierflansch nach Ventil
        reduzierflansch_nach_ventil_base         = Vector(0.0, 0.0, end_spektrometer_zachse
            ↪ + length_uebergang + length_ventilzylinder)
        inner_radius_reduzierflansch_nach_ventil = 0.0455
        outer_radius_reduzierflansch_nach_ventil = 0.125

        #Daten fuer Drahtscanner Kammer - erster Teilanschnitt mit 91cm Durchmesser
        radius_drahtkammer_erster_abschnitt = inner_radius_reduzierflansch_nach_ventil
        length_drahtkammer_erster_abschnitt = -0.102    #Aus Zeichnung abgeschaetzte 73mm
            ↪ vom Kammerabschnitt + 29mm vom Reduzierflansch
        base_drahtkammer_erster_abschnitt   = Vector(0.0, 0.0,
            ↪ reduzierflansch_nach_ventil_base[2])

        #Daten fuer den Uebergang zwischen Abschnitt 1 und 2 der Drahtkammer
        base_reduzierflansch_abschnitt_1und2         = Vector(0.0, 0.0,
            ↪ base_drahtkammer_erster_abschnitt[2] + length_drahtkammer_erster_abschnitt)
        outer_radius_reduzierflansch_abschnitt_1und2 = 0.115

        #Daten fuer Drahtscanner Kammer - zweiter Teilanschnitt mit 230mm Durchmesser
        length_drahtkammer_zweiter_abschnitt = -0.127     # Aus Zeichnung abgeschaetzte
            ↪ 250mm - 73 vom ersten Abschnitt und 50 vom dritten Abschnitt
        base_drahtkammer_zweiter_abschnitt   = Vector(0.0, 0.0,
            ↪ base_drahtkammer_erster_abschnitt[2] + length_drahtkammer_erster_abschnitt)

        #Daten fuer den Uebergang zwischen Abschnitt 2 und 3 der Drahtkammer
        base_reduzierflansch_abschnitt_2und3         = Vector(0.0, 0.0,
            ↪ base_drahtkammer_zweiter_abschnitt[2] + length_drahtkammer_zweiter_abschnitt)
        inner_radius_reduzierflansch_abschnitt_2und3 = 0.072

        #Daten fuer Drahtscanner Kammer - dritter Teilanschnitt mit 140mm Durchmesser
        length_drahtkammer_dritter_abschnitt = -0.05       # Aus Zeichnung abgeschaetzter
            ↪ Rest von 250mm - 72mm vom ersten und 127mm vom mittleren Teil
        base_drahtkammer_dritter_abschnitt   = Vector(0.0, 0.0,
            ↪ base_drahtkammer_zweiter_abschnitt[2] + length_drahtkammer_zweiter_abschnitt)

        #Daten fuer den Verbindungsflansch zwischen Elektroden Kammer und Drahtscanner Kammer
```

```
74          base_verbindungsflansch         = Vector(0.0, 0.0,
                ↪ base_drahtkammer_dritter_abschnitt[2] + length_drahtkammer_dritter_abschnitt)
75          outer_radius_verbindungsflansch = 0.125
76
77          #Daten fuer die Vakuumkammer, die das Hauptspektromter simuliert
78          radius_testzylinder    = outer_radius_verbindungsflansch
79          length_testzylinder    = -0.385   #Das sind 385mm von der Kammer + 150mm =
                ↪ Ventildurchmesser des 250er VAT Ventils
80          potential_testzylinder = 0.0
81          base_testzylinder      = Vector(0,0,base_verbindungsflansch[2])
82
83          #Daten fuer Reduzierflansch
84          reduzierflansch_base = Vector(0.0, 0.0, base_testzylinder[2] + length_testzylinder)
85          inner_radius         = 0.1
86          outer_radius         = 0.125
87          potential_reduzierflansch = 0.0
88
89          #Daten fuer letztes Stueck zum Endflansch
90          radius_endstueck    = inner_radius
91          length_endstueck    = -0.041
92          potential_endstueck = 0.0
93          base_endstueck      = Vector(0,0,reduzierflansch_base[2])
94
95          #Daten fuer Endflansch
96          endflansch_base         = Vector(0.0, 0.0, base_endstueck[2] + length_endstueck)
97          inner_radius_endflansch = 0.075
98          outer_radius_endflansch = inner_radius
99          potential_endflansch    = 0.0
100
101         #Daten fuer Elektrode
102         electrode_begin_z   = -3.04
103         electrode_begin_x   = 0.08
104         electrode_end_z     = -3.2
105         electrode_end_x     = 0.101
106         potential_electrode = -18600.0
107
108         #Daten fuer 3-eckige Elektrode
109         elektrodeflansch_base  = Vector(0.0, 0.0, electrode_begin_z)
110         elektrode_inner_radius = electrode_begin_x
111         elektrode_outer_radius = electrode_end_x - 0.015
112
113         #Daten fuer Blende auf Erdpotential
114         blende_base         = Vector(0.0, 0.0, electrode_begin_z + 0.20)
115         inner_radius_blende = 0.05
116         outer_radius_blende = 0.125
117         potential_blende    = 0
118
119         #Daten fuer lange Blende
120         #gibt keine extra Variabeln
121
122         #Daten fuer runde Elektrode
123         end    = Vector(electrode_begin_x, 0.0, electrode_begin_z)
124         start  = Vector(electrode_begin_x + 0.035, 0.0, electrode_begin_z + 0.018)
125         radius = 0.02
126         disc1  = 20
127
128         #Daten fuer den Reduzierflansch zur E-gun
129         length_reduzierflansch_egun = -0.025
130         base_reduzierflansch_egun   = Vector(0, 0, endflansch_base[2])
131
132         #Daten fuer den Basisflansch der E-gun
133         egun_basis_base      = Vector(0.0, 0.0, base_reduzierflansch_egun[2] +
                ↪ length_reduzierflansch_egun)
134         potential_egun_basis = 0.0
135
136         #Daten fuer die Kermaikstange der E-gun
137         radius_keramik_egun    = 0.0025
138         length_keramik_egun    = 0.115
```

```
139         base_keramik_egun      = Vector(0.0, 0.0, egun_basis_base[2])
140         potential_keramik_egun = 0.0
141
142         #Daten fuer Kugel der E-gun
143         radius_kugel_egun = 0.015
144         end_kugel_egun    = Vector(0.001, 0.0, base_keramik_egun[2] + length_keramik_egun)
145         start_kugel_egun  = Vector(0.001, 0.0, base_keramik_egun[2] + length_keramik_egun +
                ↪ radius_kugel_egun * 2)
146         disc1_kugel_egun               = 20
147         potential_electrode_kugel_egun = -7000
148
149         #Daten fuer optionale Blende der e-gun
150         radius_blende_egun    = 0.045
151         length_blende_egun    = 0.185
152         base_blende_egun      = Vector(0.0, 0.0, egun_basis_base[2])
153         potential_blende_egun = 0
154
155         blende_egun_eingang_base = Vector(0.0, 0.0, egun_basis_base[2] + length_blende_egun)
156         inner_radius_blende_egun = 0.0075
157
158
159         # Setup
160
161         #Uebergangsstueck
162         uebergangsstueck_cylinder = self.build_cylinder( radius_uebergang, length_uebergang,
                ↪ base_uebergang,  potential_uebergang, 80)
163         testkammer_setup.append(uebergangsstueck_cylinder)
164
165         #Ventilflansch
166         reduzierventil = self.build_flange(reduzierventil_base, inner_radius_reduzierventil,
                ↪ outer_radius_reduzierventil, potential_reduzierventil)
167         testkammer_setup.append(reduzierventil)
168
169         #Ventilzylinder
170         ventil_cylinder = self.build_cylinder(radius_ventilzylinder, length_ventilzylinder,
                ↪ base_ventilzylinder, 0.0, 20)
171         testkammer_setup.append(ventil_cylinder)
172
173         #Reduzierflansch nach Ventil
174         reduzierflansch_nach_ventil = self.build_flange(reduzierflansch_nach_ventil_base,
                ↪ inner_radius_reduzierflansch_nach_ventil,
                ↪ outer_radius_reduzierflansch_nach_ventil, 0.0)
175         testkammer_setup.append(reduzierflansch_nach_ventil)
176
177         #Drahtscanner Kammer - erster Teilabschitt
178         drahtkammer_erster_abschnitt =
                ↪ self.build_cylinder(radius_drahtkammer_erster_abschnitt,
                ↪ length_drahtkammer_erster_abschnitt, base_drahtkammer_erster_abschnitt, 0.0,
                ↪ 25)
179         testkammer_setup.append(drahtkammer_erster_abschnitt)
180
181         #Uebergang zwischen Abschitt 1 und 2
182         reduzierflansch_abschnitt_1und2 =
                ↪ self.build_flange(base_reduzierflansch_abschnitt_1und2,
                ↪ inner_radius_reduzierflansch_nach_ventil,
                ↪ outer_radius_reduzierflansch_abschnitt_1und2, 0.0)
183         testkammer_setup.append(reduzierflansch_abschnitt_1und2)
184
185         #Drahtscanner Kammer - zweiter Abschnitt
186         drahtkammer_zweiter_abschnitt =
                ↪ self.build_cylinder(outer_radius_reduzierflansch_abschnitt_1und2,
                ↪ length_drahtkammer_zweiter_abschnitt, base_drahtkammer_zweiter_abschnitt,
                ↪ 0.0, 50)
187         testkammer_setup.append(drahtkammer_zweiter_abschnitt)
188
189         #Uebergang zwischen Abschitt 2 und 3
190         reduzierflansch_abschnitt_2und3 =
                ↪ self.build_flange(base_reduzierflansch_abschnitt_2und3,
```

```
            ↪ inner_radius_reduzierflansch_abschnitt_2und3,
            ↪ outer_radius_reduzierflansch_abschnitt_1und2, 0.0)
191         testkammer_setup.append(reduzierflansch_abschnitt_2und3)
192
193         #Drahtscanner Kammer - dritter Abschnitt
194         drahtkammer_dritter_abschnitt =
            ↪ self.build_cylinder(inner_radius_reduzierflansch_abschnitt_2und3,
            ↪ length_drahtkammer_dritter_abschnitt, base_drahtkammer_dritter_abschnitt,
            ↪ 0.0, 25)
195         testkammer_setup.append(drahtkammer_dritter_abschnitt)
196
197         #Verbindungsflansch Drahtscanner Kammer und Elektroden Kammer
198         verbindungsflansch = self.build_flange(base_verbindungsflansch,
            ↪ inner_radius_reduzierflansch_abschnitt_2und3,
            ↪ outer_radius_verbindungsflansch, 0.0)
199         testkammer_setup.append(verbindungsflansch)
200
201         #Vakuumkammer, die das Hauptspektrometer simuliert
202         testzylinder_cylinder = self.build_cylinder( radius_testzylinder,
            ↪ length_testzylinder, base_testzylinder, potential_testzylinder, 100)
203         testkammer_setup.append(testzylinder_cylinder)
204
205         #Reduzierflansch
206         reduzierflansch = self.build_flange(reduzierflansch_base, inner_radius,
            ↪ outer_radius, potential_reduzierflansch)
207         testkammer_setup.append(reduzierflansch)
208
209         #letztes Stueck zum Endflansch
210         endstueck_cylinder = self.build_cylinder( radius_endstueck, length_endstueck,
            ↪ base_endstueck, potential_endstueck, 5)
211         testkammer_setup.append(endstueck_cylinder)
212
213         #Endflansch
214         endflansch = self.build_flange(endflansch_base, inner_radius_endflansch,
            ↪ outer_radius_endflansch, potential_endflansch)
215         testkammer_setup.append(endflansch)
216
217         #Elektrode
218         electrode = Cone(electrode_begin_z, electrode_begin_x, electrode_end_z,
            ↪ electrode_end_x, potential_electrode)
219         rectangles = electrode.discretise(scale, power, nrot)
220         testkammer_setup.append(rectangles)
221
222         #Elektrode wird 3-eckig
223         #elektrodeflansch = self.build_flange(elektrodeflansch_base, elektrode_inner_radius,
            ↪ elektrode_outer_radius, potential_electrode)
224         #testkammer_setup.append(elektrodeflansch)
225
226         #Blende auf Erdpotential
227         #blende = self.build_flange(blende_base, inner_radius_blende, outer_radius_blende,
            ↪ potential_blende)
228         #testkammer_setup.append(blende)
229
230         #lange Blende
231         #lange_blende = Cone(blende_base[2], inner_radius_blende, blende_base[2] - 0.05,
            ↪ inner_radius_blende + 0.03, 0)
232         #rectangles = lange_blende.discretise(scale, power, nrot)
233         #testkammer_setup.append(rectangles)
234
235         #runde Elektrode
236         rundeElektrode = self.build_defcircle(end,start,radius, disc1, potential_electrode,
            ↪ nrot)
237         testkammer_setup.append(rundeElektrode)
238
239         #Reduzierflansch zur E-gun
240         reduzierflansch_egun = self.build_cylinder(inner_radius_endflansch,
            ↪ length_reduzierflansch_egun, base_reduzierflansch_egun, potential_endstueck,
            ↪ 5)
```

```
241         testkammer_setup.append(reduzierflansch_egun)
242 
243         #E-gun Basis
244         egun_basis = self.build_flange(egun_basis_base, 0.0, inner_radius_endflansch,
                ↪ potential_egun_basis)
245         testkammer_setup.append(egun_basis)
246 
247         #E-gun Keramik
248         keramik_egun = self.build_cylinder(radius_keramik_egun, length_keramik_egun,
                ↪ base_keramik_egun, potential_keramik_egun, 5)
249         testkammer_setup.append(keramik_egun)
250 
251         #Kugel der E-gun
252         kugel_egun = self.build_defcircle(end_kugel_egun, start_kugel_egun,
                ↪ radius_kugel_egun, disc1_kugel_egun, potential_electrode_kugel_egun, nrot)
253         testkammer_setup.append(kugel_egun)
254 
255         #optionale Blende fuer die egun
256         #blende_egun = self.build_cylinder(radius_blende_egun, length_blende_egun,
                ↪ base_blende_egun, potential_blende_egun, 25)
257         #testkammer_setup.append(blende_egun)
258         #blende_egun_eingang = self.build_flange(blende_egun_eingang_base,
                ↪ inner_radius_blende_egun, radius_blende_egun, potential_blende_egun)
259         #testkammer_setup.append(blende_egun_eingang)
260 
261 
262         # Output
263 
264         #print "(positive) wire dU: ", self.wire_dU
265         print base_testzylinder
266 
267         if filename:
268             testkammer_setup.elcd3_3(filename)
```

C.2. Geometry files

The full electrode geometry for the test setup is not shown here due to its length of about 350 lines. To give an idea at how the file is constructed, only a few lines from the beginning are shown (C.2). Using the builder script from the previous section, it would be possible to reproduce the whole file. The geometry for Hillen's pulsed coil consists of just one element and is shown below, too (C.3). The parameters were taken from the settings of a `bfield3` simulation done by Hillen; as shown in figure C.1. For details on the file format of the geometry files, see [Glu04] and [Glu06b]. Note that in the next release of Kassiopeia, the input file format is most likely to change to a more user-friendly, high-level format based on XML.

```
Anzahl der (*Spulen) : 1

 Nr.  POSITION[cm]  RADIUS[cm]  THICKNESS[cm]  LENGTH[cm]   WINDINGS  THIN COILS   CURRENT[A]  ANGLE[°]  OFFSET[cm]
  1         299.30       15.70         4.80       10.00  100.000000     20.00   38640.00        0.0        0.00

Iterationsbereich:
   XMIN[cm]    XMAX[cm]   STEPX[cm]
     200.00      316.00        1.00
   YMIN[cm]    YMAX[cm]   STEPY[cm]
       0.00        0.29        6.59
   ZMIN[cm]    ZMAX[cm]   STEPZ[cm]
       0.00        0.10        1.00
```

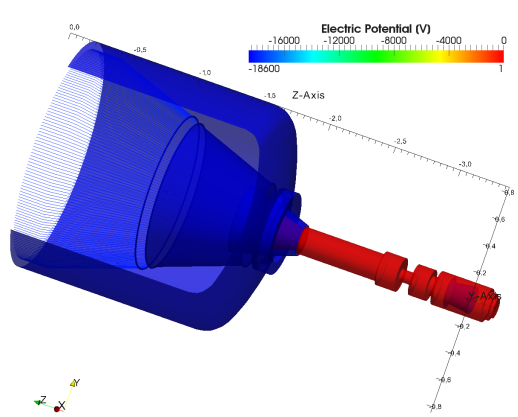
Figure C.1.: Parameters of the pulsed coil. These are the parameters which were used in `bfield3` simulations done by Hillen. Figure taken from [Hil12].

Listing C.2: aufbau+prespec.el32 (partial)

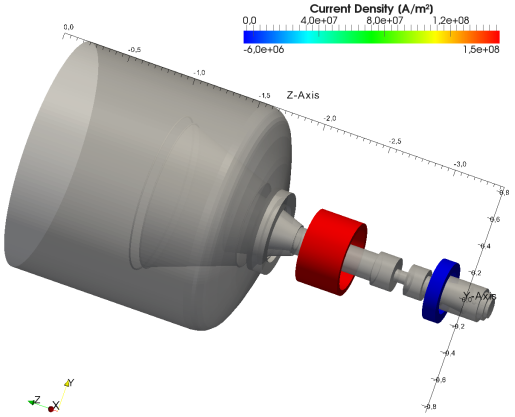
```
348
-1.372 0.84 0 0.84 -18600 12
-1.427 0.831 -1.372 0.84 -18600 5
-1.453 0.82 -1.427 0.831 -18600 5
-1.465 0.812 -1.453 0.82 -18600 5
-1.497 0.786 -1.465 0.812 -18600 6
-1.514 0.764 -1.497 0.786 -18600 6
-1.525 0.743 -1.514 0.764 -18600 6
-1.562 0.664 -1.525 0.743 -18600 8
-1.611 0.531 -1.562 0.664 -18600 8
-1.627 0.482 -1.611 0.531 -18600 6
-1.669 0.305 -1.627 0.482 -18600 8
-1.669 0.305 -1.629 0.305 -18600 4
-1.629 0.25 -1.629 0.305 -18600 4
-1.689 0.25 -1.629 0.25 -18600 3
-1.689 0.1923077 -1.689 0.25 -18600 4
-1.689 0.193077 -1.68904 0.1926 -18600 1
-1.68915 0.19213 -1.68904 0.1926 -18600 1
-1.68934 0.19168 -1.68915 0.19213 -18600 1
-1.68959 0.19127 -1.68934 0.19168 -18600 1
-1.6899 0.1909 -1.68959 0.19127 -18600 1
-1.69027 0.19059 -1.6899 0.1909 -18600 1
-1.69068 0.19034 -1.69027 0.19059 -18600 1
-1.69113 0.19015 -1.69068 0.19034 -18600 1
-1.6916 0.19004 -1.69113 0.19015 -18600 1
```

Listing C.3: hillencoil.mag3

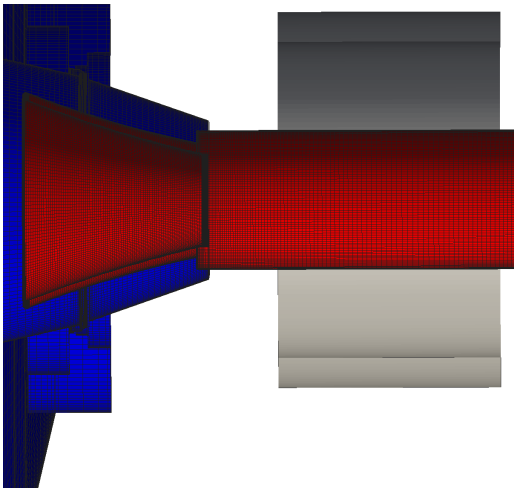
```
    1
-6.250000e+06  0.00 0.00 -2.943000e+00  0.00 0.00 -3.043000e+00  1.570000e-01 2.250000e-01
    ↪ 100
```



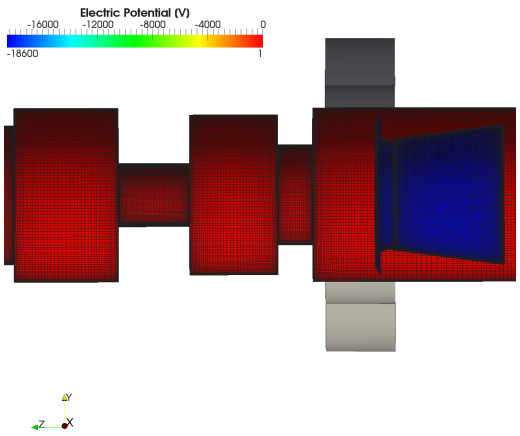
(a) Elcd geometry.



(b) Magfield geometry.



(c) Detail of the pre-spectrometer junction.



(d) Detail of the main spectrometer electrode.

Figure C.2.: Simulation geometry used in this work. Figure a shows the electrode geometry, consisting of the pre-spectrometer with conical and wire electrodes and the test setup used by Hillen for his experimental studies. The objects are colored by electric potential. Figure b shows the coil geometries on top of the electrode geometry. The red-colored coil is the pre-spectrometer solenoid, while the pulsed coil is shown in blue (color according to the respective current densities). Figures c and d shows details of the geometry at the ends of the Penning trap region. On the left, the junction to the pre-spectrometer is shown as cross-cut. It can be seen how the ground electrode reaches into the pre-spectrometer vessel. The figure on the right shows the opposite end of the setup, including the special electrode which electrically simulates the main spectrometer. The position of the pulsed coil with can be seen, too.

List of Figures

List of Tables

Listings

Bibliography

[Bla05] K. Blaum, *High-accuracy mass spectrometry with stored ions*, Phys. Rep. **425** (2006) 1–78, doi:10.1016/j.physrep.2005.10.011

[Cor11] T.J. Corona, KATRIN internal document, 2011, BSCW: General Meetings and Review Panels - Collaboration Meetings - 21. Collaboration Meeting

[Cor12] T.J. Corona, private communication, 2012

[Dig03] A.S. Dighe et al., *Detecting the Neutrino Mass Hierarchy with a Supernova at IceCube*, arXiv:hep-ph/0303210v3

[DB12] The Daya Bay collaboration, *Observation of electron-antineutrino disappearance at Daya Bay*, Phys. Rev. Lett. **108**, 171803 (2012), arXiv:1203.1669v2 [hep-ex]

[DC12] The Double Chooz collaboration, *Indication for the disappearance of reactor* ν_e *in the Double Chooz experiment*, arXiv:1112.6353v3 [hep-ex]

[Doe11] P. Doe, KATRIN internal document, 2011, BSCW: General Meetings and Review Panels - Collaboration Meetings - 21. Collaboration Meeting

[Ess04] K. Essig, *Diploma thesis*, Helmholtz-Institut für Strahlen- und Kernphysik, Rheinische Friedrich-Wilhelms-Universität Bonn, 2004

[Fur10] D. Furse, KATRIN internal document, 2010, BSCW: General Meetings and Review Panels - Collaboration Meetings - 18. Collaboration Meeting

[For11a] J. Formaggio, KATRIN internal document, 2011, BSCW: General Meetings and Review Panels - Collaboration Meetings - 21. Collaboration Meeting

[For11b] J. Formaggio et al., *Solving for Micro- and Macro- Scale Electrostatic Configurations Using the Robin Hood Algorithm*, arXiv:1111.5035v1 [physics.comp-ph]

[Fur11] D. Furse, KATRIN internal document, 2011, BSCW: General Meetings and Review Panels - Collaboration Meetings - 21. Collaboration Meeting

[Fur12] D. Furse, private communication, 2011–2012

[Glu04] F. Glück, KATRIN internal document, 2004, BSCW: EMD - Programs - electric fields

[Glu06a] F. Glück, KATRIN internal document, 2006, BSCW: EMD - Programs - electric fields

[Glu06b] F. Glück, KATRIN internal document, 2006, BSCW: EMD - Programs - magnetic fields

[Glu08] F. Glück, KATRIN internal document, 2008, BSCW: General Meetings and Review Panels - Collaboration Meetings - 15. Collaboration Meeting

[Glu10] F. Glück, KATRIN internal document, 2010, BSCW: EMD - Programs

[Glu11] F. Glück, *Axisymmetric magnetic field calculation with zonal harmonic expansion*, Progress In Electromagnetics Research **B**, Vol. 32, 351–388 (2011), doi:10.2528/PIERB11042108

[Glu12] F. Glück, private communication, 2012

[Hil11] B. Hillen, *PhD thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2011

[Hil12] B. Hillen, private communication, 2011–2012

[Hug08] K. Hugenberg, *Diploma thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2008

[Wol11] J. Wolf, KATRIN internal communication, 2011

[Kam08] The KamLAND collaboration, *Precision Measurement of Neutrino Oscillation Parameters with KamLAND*, Phys. Rev. Lett. **100**, 221803 (2008), arXiv:0801.4589v3 [hep-ex]

[Kass] *The Comprehensive Guide to Kassiopeia*, KATRIN internal document (included in the Kassiopeia source code at `Documentation/UserGuide/`)

[KAT04] The KATRIN collaboration, *KATRIN Design Report 2004*, FZKA Scientific Report 7090

[KAT06] Komitee für Astroteilchenphysik (KAT), Kosmische Spurensuche - Astroteilchenphysik in Deutschland

[Lei12] B. Leiber, KATRIN internal document, 2012, BSCW: General Meetings and Review Panels - Collaboration Meetings - 22. Collaboration Meeting

[MIN11] The MINOS collaboration, *Measurement of the neutrino mass splitting and flavor mixing by MINOS*, Phys. Rev. Lett. **106**, 181801 (2011), arXiv:1103.0340v1 [hep-ex]

[Obl11] N. Oblath, KATRIN internal document, 2011, BSCW: General Meetings and Review Panels - Collaboration Meetings - 21. Collaboration Meeting

[OP10] The OPERA collaboration, *Observation of a first ν_τ candidate in the OPERA experiment in the CNGS beam*, Phys. Lett. B **691**, 138–145 (2010), arXiv:1006.1623v1 [hep-ex]

[Ott09] E.W. Otten & C. Weinheimer, *Neutrino mass limit from tritium β-decay*, Rept. Prog. Phys. **71**, 086201 (2008), arXiv:0909.2104v1 [hep-ex]

[Ott10] E. Otten, *Removal of stored particles in MAC-E filters*, KATRIN internal communication, 2010

[Pau30] W. Pauli, *Collected Scientific Papers*, Eds. Kronig & Weisskopf, Vol. 2, 1316–1317

[Pic92] A. Picard et al., *A solenoid retarding spectrometer with high resolution and transmission for* keV *electrons*, Nuclear Instruments and Methods in Phys. Res. **B63** (1992) 345–358, doi:10.1016/0168-583X(92)95119-C

[PV] ParaView – Open Source Scientific Visualization, www.paraview.org

[Py] Python programming language, www.python.org

[Rei59] R. Reines & C.L. Cowan, *Free Antineutrino Absorption Cross Section. I. Measurement of the Free Antineutrino Absorption Cross Section by Protons*, Phys. Rev. **113**, 273 (1959), doi:10.1103/PhysRev.113.273

[REN12] The RENO collaboration, *Observation of Reactor Electron Antineutrino Disappearance in the RENO Experiment*, arXiv:1204.0626v2 [hep-ex]

[Ruby] Ruby programming language, www.ruby-lang.org

[ROOT] ROOT – A Data Analysis System, root.cern.ch

[San03] S. Sánchez Majos, *Diploma thesis*, Institut für Physik, Johanes-Gutenberg-Universität Mainz, 2003

[SK08] The Super-Kamiokande collaboration, *Solar neutrino measurements in Super-Kamiokande-II*, Phys. Rev. D **78**, 032002 (2008), arXiv:0803.4312 [hep-ex]

[SNO01] The SNO collaboration, *Measurement of the rate of* $\nu_e + d \rightarrow p + p + e^-$ *interactions produced by* ^8B *solar neutrinos at the Sudbury Neutrino Observatory*, Phys. Rev. Lett. **87**, 071301 (2001), arXiv:nucl-ex/0106015v2

[SNO11] The SNO collaboration, *Combined Analysis of all Three Phases of Solar Neutrino Data from the Sudbury Neutrino Observatory*, arXiv:1109.0763 [nucl-ex]

[Stei12] N. Steinbrink, *Diploma thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2012

[Thu07] T. Thümmler, *PhD thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2007

[Val04] K. Valerius, *Diploma thesis*, Helmholtz-Institut für Strahlen- und Kernphysik, Rheinische Friedrich-Wilhelms-Universität Bonn, 2004

[Val09] K. Valerius, *PhD thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2009

[Voe08] S. Vöcking, *Diploma thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2008

[VTK] VTK – The Visualization Toolkit, www.vtk.org

[WC] Wikimedia Commons, commons.wikimedia.org

[Wol08] I. Wolff, *Diploma thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2008

[Zac09] M. Zacher, *Diploma thesis*, Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2009

[Zac10] M. Zacher, KATRIN internal document, 2010, BSCW: General Meetings and Review Panels - Collaboration Meetings - 18. Collaboration Meeting

[Zac13] M. Zacher, *PhD thesis* (in preparation), Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, 2013

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die zur Entstehung dieser Diplomarbeit beigetragen haben.

Ganz besonders bedanke ich mich bei Prof. Dr. Christian Weinheimer für die Vergabe dieses interessanten Themas und die Betreuung während meiner Arbeit; außerdem für die Möglichkeit so viele verschiedene Tagungen und Konferenzen zu besuchen. Gesondert erwähnen möchte ich hier den KATRIN Simulation Workshop in Berkeley im Dezember 2012, an dem ich teilnehmen durfte. Herr Weinheimer ermöglichte mir außerdem, bei der Betreuung des IKP-Rechenclusters mitzuwirken und so neue Erfahrungen in diesem nicht alltäglichen Bereich zu sammeln. Weiterhin danke ich Prof. Dr. Johannes Wessels für die Übernahme der Zweitkorrektur dieser Arbeit. Michael Zacher und Volker Hannen danke ich für die umfassende Unterstützung, die vielen nützlichen Tipps und die Durchsicht meiner Arbeit. Bedanken möchte ich mich auch bei all meinen Kollegen im Institut für Kernphysik in Münster für die nette Arbeitsatmosphäre und die vielen Hilfestellungen, vor allem natürlich bei Daniel, Nicho und den anderen „Mitbewohnern" meines Büros. Mein Dank geht außerdem an die Kollegen am KIT Karlsruhe und an den verschiedenen amerikanischen Universitäten, mit denen ich während meiner Arbeit zusammenarbeiten konnte. Weiterhin bedanke ich mich bei meinen Kommilitonen Stephan, Tobi, Lukas und all den anderen, die mir in meinem Studium immer zur Seite standen. Mein besonderer Dank gilt zudem Bastian, der mich nicht nur im Studium, sondern bereits während meiner Schulzeit begleitet hat und immer ein offenes Ohr für meine kleineren und größeren Probleme hatte. Zu guter Letzt bedanken möchte ich mich bei meiner (erweiterten) Familie, die mich wo immer nötig nach Kräften unterstützt hat, und ganz besonders bei meinem Vater Hans-Gerd – ohne dich wäre ich niemals bis hierhin gekommen!

Eigenständigkeitserklärung

Ich versichere, dass ich diese Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und Zitate kenntlich gemacht habe.

Jan David Behrens