Department of Physics and Astronomy
University of Heidelberg

Bachelor Thesis in Physics
submitted by

Manuel Wittner

born in Bietigheim-Bissingen (Germany)

2014

Study of the Applicability of Markov Chain Monte Carlo Methods to the Statistical Separation of Electron Sources via the Impact Parameter for ALICE

This Bachelor Thesis has been carried out by Manuel Wittner at the
Physikalisches Institut Heidelberg
under the supervision of
Prof. Dr. Johanna Stachel

Abstract

One particularly interesting measurement detected by the ALICE (A Large Ion Collider Experiment) set-up are electrons from the decay of heavy-flavor quarks, such as charm and especially beauty quarks. In order to investigate these electrons, it is crucial to distinguish them from background electrons, which are created for example through photon conversion or Dalitz decays as well as to distinguish the charm from the beauty electrons. For this distinction, the impact parameter distribution of the electrons is used. One important task is to find out the contribution of the individual sources of electrons to the impact parameter total distribution containing the electrons of sources. To find these contributions, the so called 'strength factors' of the sources, usually a maximum likelihood fit is done, whereby the strength factors are free parameters of the fit. However, in a maximum likelihood fit there are some disadvantages, e.g. only an approximated uncertainty of the fit value exists and it is impossible to use prior knowledge about the strength factors.

According to 'Bayes' theorem', the likelihood can be used to build a so called 'posterior distribution', which is basically a probability distribution for the free parameters, whereby prior knowledge can be used. Sampling this posterior distribution would cancel the disadvantages of the maximum likelihood method. Hence, in this bachelor thesis a Markov Chain Monte Carlo (MCMC) sampling algorithm is implemented and applied on the posterior distribution. Furthermore, some properties like the correlation of the sampling points, the convergence of the algorithm, and the applicability of MCMC methods to this problem in general are investigated.

Zusammenfassung

Eine besonders interessante Messung des ALICE (A Large Ion Collider Experiment) Detektors beinhaltet die Elektronen, die aus dem Zerfall schwerer Quarks wie dem Charm- und vorallem dem Beauty-Quark entstehen. Um jene Elektronen zu untersuchen, ist es essenziell, sie sowohl von den Hintergrund-Elektronen, die beispielsweise durch Paarerzeugung von Photonen oder durch Dalitz-Zerfälle entstehen, als auch voneinander zu unterscheiden. Für diese Unterscheidung wird die Stoßparameterverteilung der Elektronen benutzt. Eine wichtige Aufgabe ist, die Anteile der einzelnen Elektronquellen an der gesamten Stoßparameterverteilung herauszufinden. Um diese Anteile, die sogenannten „Stärkefaktoren“, herauszufinden, wird gewöhnlich ein Maximum-Likelihood-Fit mit den Stärkefaktoren als freie Parameter gemacht. Allerdings birgt ein solcher Fit manche Nachteile, wie z.B., dass die Unsicherheiten der Fitwerte nur näherungsweise bestimmt werden können und es unmöglich ist, Vorwissen über die Stärkefaktoren miteinzubeziehen.

Nach dem „Satz von Bayes“ kann aus der Likelihood eine sogenannte „Posterior-Verteilung“ gebildet werden, die im Wesentlichen eine Wahrscheinlichkeitsverteilung für die freien Parameter unter Einbezug von Vorwissen darstellt. Die Nachteile der Maximum-Likelihood-Methode würden durch das Samplen der Posterior-Verteilung aufgehoben werden. Deshalb wird in dieser Bachelorarbeit ein Markov-Chain-Monte-Carlo-Algorithmus implementiert und auf die Posterior-Verteilung angewendet. Desweiteren werden einige Eigenschaften wie beispielsweise die Korrelation der Sample-Punkte, die Konvergenz des Algorithmus oder die generelle Anwendbarkeit dieser Methode auf das Problem untersucht.

Contents

1 Introduction

1.1 Particle Physics

1.1.1 The Standard Model

The Standard Model Of Particle Physics (SM) forms the theoretical foundation of the modern physical world view in the microscopic regime. According to it, all matter consists of elementary particles, which can be classified into quarks, leptons, gauge bosons, and the Higgs boson. Among the quarks, one can distinguish between positive charged and negative charged ones, as well as between three generations. There are also three generations of charged and uncharged leptons, which are also called 'neutrinos'. Similarly, there are antiparticles with an inverse charge Q for every quark and lepton.

The three fundamental interactions, the 'Strong Interaction', the 'Weak Interaction', and the 'Electromagnetic Interaction', are transmitted via the respective gauge bosons, which are the 'Gluon', the 'Z^0- and $W^{\pm}$-boson' and the 'Photon'. The fourth interaction, gravitation, is not part of the SM. While the weak interaction can act on all quarks and leptons, the electromagnetic interaction can only do so on charged particles. The strong interaction can act on all gluons and quarks and hence on all particles which are built of quarks,

leptons		quarks		bosons
$Q=-1$	$Q=0$	$Q=2/3$	$Q=-1/3$	H
e	ν_e	u	d	γ
μ	ν_μ	c	s	Z^0, $W^{\pm}$
τ	ν_τ	t	b	g

Table 1.1: Classification of elementary particles in the SM.

which is why nuclei can be bound together to build a nucleus. Since the so called color charge, which is carried by all quarks and gluons, has never been observed isolated, but always neutral in color, one can assume that it is impossible to observe single quarks. Instead, quarks build so called hadrons, i.e. doublets or triplets of quarks which cancel each other with respect to the color. This phenomenon is called "confinement".

1.1.2 Quark-Gluon Plasma

According to Lattice Quantum Field Theory, when temperature trespasses a critical value of about 160 MeV [12] (with the Boltzmann constant $k_B \equiv 1$) or when the baryochemic potential gets too high, quarks do not exist in the hadronic state (confinement) anymore, but they form a so called "*Quark-Gluon Plasma* (QGP)". A theoretical explanation for this is, that the strong interaction becomes weaker as the distance of the quarks gets smaller. In this state, quarks which are not bound into hadrons can exist. Theoretical descriptions do not consider this system as single particles but as matter. Hence, thermodynamic terms like temperature, phases or phase transitions can be used to describe the QGP. However, this view requires that there are many particles involved in the QGP as well as that there is local equilibrium such that macroscopic quantities like temperature, pressure, energy or entropy density can be defined. Hence, the lifetime of the system must be of a bigger order than the inverse rate of interactions such that thermal equilibrium has enough time to appear. In order to fulfill this requirements, one needs a many particle collision with high energy. Therefore, in the ALICE experiment, collisions of heavy nuclei at high energies are observed, from which one knows that they can create enough particles and a high enough temperature to form QGP [12].

Investigating QGP is of great interest in order to obtain insights about the strong interaction in general, but also about the time shortly after the big bang, since one assumes that the universe has been in this state in its first few fractions of a second, and maybe also about neutron stars, because there are theoretical assumptions that there is QGP in the inside of some neutron stars. However, there is no experimental proof for this.

1.2 The ALICE Experiment

1.2.1 General Overview

In nature, one expects QGP only in the inside of neutron stars. Since these systems are experimentally inaccessible, in order to investigate QGP one has to produce it in the laboratory. Therefore, the strategy of ALICE is to observe pp, p-Pb, and Pb-Pb collisions at ultra relativistic energies, because one expects QGP to occur in latter ones. Hereby, the pp and p-Pb collisions are used as references to control the Pb-Pb collisions. The particles are accelerated by the Large Hadron Collider (LHC) to a center of mass energy of $\sqrt{s_{NN}} = 2.76$ TeV [3]. Past the collision, the QGP is formed and hadronizes after about 10^{-22} s [12]. Additionally, in the collision many particles are created, which experience energy loss through the QGP. The ALICE detector is able to identify and track many of those particles or their decay products so that the properties of the QGP can be investigated. The major aims of ALICE are to find out these properties of the QGP like the critical temperature, speed of sound, degrees of freedom or transport coefficients. With these properties, one can for example garner insights about QCD with many particles [15].

1.2.2 The ALICE Detector

The ALICE detector has been constructed with the objective to study the evolution of the system of the QGP in space and time as precisely as possible. Hence, many different kinds of subdetectors are combined to provide an extensive overview of the observed system. The detector is divided into the central barrel, which consists of several subdetectors cylindrically arranged around the collision point, whereby the axis of the cylinder concurs with the beam axis, and the forward muon spectrometer. A large solenoid magnet is placed around the central barrel and provides a magnetic field of 0.5 T in order to bend the trajectories of charged particles so that they can be identified. Since electrons are detected in the central barrel, a short overview about the used subdetectors for electron analyses follows.

The innermost subdetector is the so called "*Inner Tracking System* (ITS)",

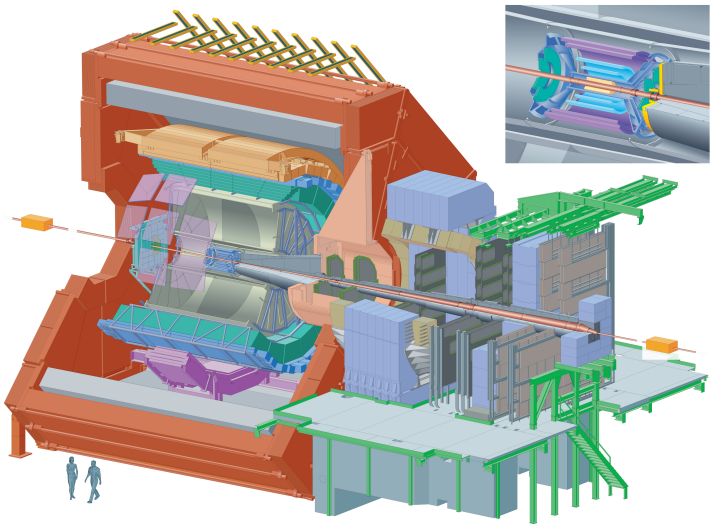


Figure 1.1: ALICE detector, overview [2].

which is made of six layers of silicon detectors. Its purpose is the identification of particles up to low momenta, and most important the determination of the primary vertex and secondary vertices. It contributes to the determination of the impact parameter strongest in the global tracking, which will be of great importance for the separation of electron sources and will also be dealt with in this thesis. Additionally the ITS enhances the momentum and angular resolution [16, 2, 21].

Around the ITS, the "*Time Projection Chamber* (TPC)" is arranged. The gas detector represents the most important subdetector and consists of a chamber, which is filled with 90 % neon and 10 % carbon dioxide. However, during the data acquisition, the gas mixture consisted of neon, carbon dioxide, and nitrogen with the proportions $90 - 10 - 5$ [5]. In between the end caps, there is a high voltage electrode causing the electrons which are produced from ionisations of the gas atoms by the incoming particles to move to the end caps. There, the electron signal is amplified and read out by a multi-wire proportional chamber. Since the drift velocity of the electrons in the gas is constant due to the homogeneous electric field and interactions with the gas, the arrival time of the electrons can be used to calculate the coordinate parallel to the E-field of the trajectory of the incoming particle. Therefrom comes the name of

the TPC. The other two coordinates are determined by the position where the drift electron arrives at the end cap. The TPC produces the largest amount of data of all subdetectors and, just as the ITS, provides tracking and particle identification information. The particle identification (PID) is based on the different energy losses of particles with same momenta but different masses. These energy losses yield the TPC signal [18, 2, 21].

The "*Transition Radiation Detector* (TRD)" consists of 522 detector chambers, which are arranged cylindrically around the particle beam such that there lie 5 rings of chambers along the beam axis. These rings are separated into 18 sectors of the azimutal angle and in each of these 90 positions there are 6 layers of chambers piled up. This arrangement of the chambers would yield an amount of 540 chambers, yet there is a "hole" of 3×6 chambers for the ALICE Photon Spectrometer (PHOS), which is why there are only 522 chambers. The chambers consist each of a drift chamber and a radiator. Incoming charged particles with $\gamma = E/m \gtrapprox 1000$, which is only for electrons with high energy the case [8], create transition radiation when passing the radiator and ionize the gas in the drift chamber. Afterwards, if the transition photon has enough energy, it can create additional ionizations of the xenon gas in the drift chamber. Therefore, the TRD signal is the combination of specific energy loss and transition radiation. At the end of each drift chamber, a Multi-Wire Proportional Chamber (MWPC) is placed. A voltage is applied between the wires and the radiator such that the freed electrons drift towards the MWPC, where they are read out by means of gas amplification [8]. This signal looks different for electrons and pions even at high momenta and thus they can be separated, where the TPC cannot do it. Hence, the TRD aims mainly at the improvement of electron identification and tracking. Additionally, the TRD is used as a trigger detector for ALICE [19, 2, 21].

The "*Time Of Flight* (TOF)" detector measures the time particles need to get from the primary vertex to the TOF. It consists of 6 layers of readout channels, which are spread cylindrically around the beam axis. In between these layers there is gas, which is ionized by the incoming particles. This arrangement is called 'Multigap Resistive Plate Chamber'. The TOF is needed for particle identifications at low momenta, because the signal of the TPC

would overlap too much or different kinds of particles are at the crossing points of the TPC signal and hence one would not be able to distinguish between particle species [17, 2, 21].

1.2.3 Heavy Flavor Physics

After the heavy nuclei collision, the QGP is formed and many quarks of all kinds are produced. Additionally, in the quark gluon plasma, light quarks can be created through thermal production. Since thermal production is strongly correlated with the mass of particles, heavy quarks like charm, beauty or bottom quarks are produced almost completely in the initial scattering process on a very short time scale [21]. While the top quarks decay nearly immediately after their production, the charm and beauty quarks hadronize, traverse the QGP and decay afterwards. For example, one can consider a charm quark, which is produced together with an anti-charm quark almost always through gluon-gluon interactions in the initial scattering process. With a probability of 56.5 %, they hadronize into a neutral D and $\bar{D}$ meson, which decay typically a fraction of a millimeter away from the primary vertex, which is the location where the heavy nuclei collision took place [15]. In this fraction of a millimeter, the charm and beauty quark transmigrate the QGP and experience its evolution in space and time. Since the top quark decays shortly after its production, the distance it moves is negligible. Yet, the top quark can decay into a bottom quark, which then traverses the whole QGP. The fact that the heavy quarks transmigrate the whole QGP and have a mass, which is larger than the maximum initial QGP temperature, is the reason why the charm and bottom quark are of great interest to observe. By observing their energy loss in the QGP medium, one can garner information about QCD for many particles as well as about the QGP in general [15]. However, heavy quarks cannot be detected apartly, since they are subject to confinement. For this reason they hadronize into baryons or mesons. Besides the hidden heavy flavor J/Ψ and Υ mesons and the open heavy flavor Λ_c baryon the most frequently created heavy flavor hadrons are the D and B meson, which consist of a charm and bottom quark respectively and a lighter quark. Hence, they are also open heavy flavor

mesons. They can decay into an electron [21]:

$$\begin{aligned} H_c &\rightarrow e^{\pm} + X \\ H_b &\rightarrow e^{\pm} + X \end{aligned} \tag{1.1}$$

H_c and H_b denote here hadrons, which contain a charm or a beauty quark. In general, semileptonic decays of these hadrons have quite large branching ratios of $9.6 \pm 0.4\,\%$ for $c \rightarrow l^+ + X$ and $20.5 \pm 0.7\,\%$ for $b \rightarrow l^+ + X$, which makes them attractive to observe because they allow for higher statistics [7]. Since the D and B meson have a long lifetime, they often show a detectable distance to the primary vertex before they decay. The detected electrons and positrons from the semileptonic decays are impure, because there are also electrons produced through pair creation of photons in the detector material and Dalitz electrons, which are primary electrons mainly produced through Dalitz decays of pions. Since one is mainly interested in the heavy flavor electrons, particularly in the beauty electrons due to their long lifetime, one wants to know the distribution of the electrons along a quantity called the *impact parameter*.

The impact parameter can be considered as something similar as the *distance of closest approach* (DCA). In Figure 1.2, one can see the primary vertex, which is the location where the collision of the lead nuclei has occurred. The mother particle, e.g. a c or b quark, is produced. The mother particle travels through the QGP, hadronizes into a D or B meson and decays into an electron and other particles at the location of the secondary vertex. After the detection of the electron, its trajectory can be reconstructed and extrapolated, as if the electron had already existed before its creation at the secondary vertex. Then, one defines the DCA as the closest distance between the reconstructed trajectory and the primary vertex. The impact parameter is the DCA provided with a sign, which implies, whether the reconstructed electron track goes past on the right side or the left side of the mother particle. It should be mentioned that every decay is also possible mirrored at the track of the mother particle with all particles replaced by their antiparticles. However, in this case the impact parameter changes its sign, which leads to a mirrored impact parameter distribution. This changes the look of the asymmetric distribution of the con-

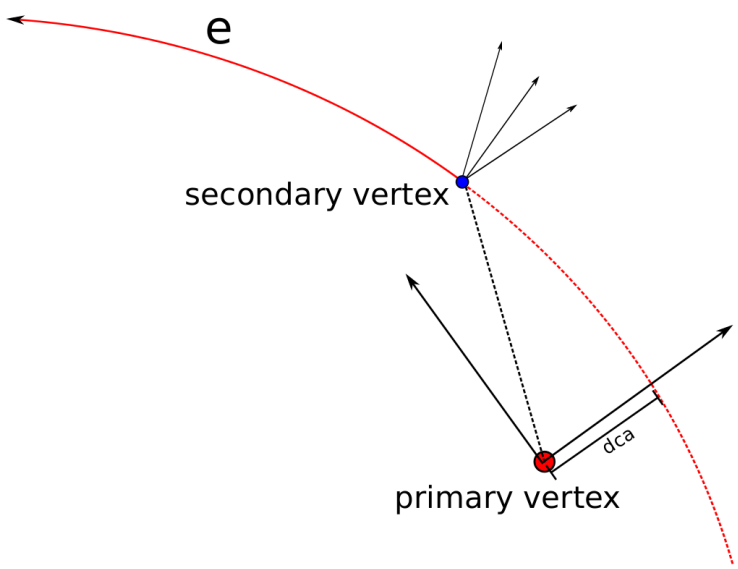


Figure 1.2: Illustration of the DCA [21].

version electrons, which are electrons created through pair creation of photons in the detector material. Additionally, the magnetic field in the detector has a different impact on differently charged particles. Hence, the impact parameter is often multiplied with the charge of the particle, so that the distributions look the same. The long lifetime of the D and B meson and consequently the detectable distances of the secondary vertex to the primary vertex lead to broadened distributions for the impact parameters of the detected electrons. This is of great importance in order to distinguish the sources of the electrons.

Considering now the impact parameter distribution of the detected electrons in Figure 1.3, one sees the total electron signal represented by the black dots. The transverse momentum p_T, which is the momentum of a particle perpendicular to the beam axis, is cut to $1.5 < p_T < 2.0$ GeV/c. This distribution is a superposition of distributions of several sources, such as electrons created from charm or beauty decays, photon conversion in the detector material, or from Dalitz decays of neutral pions [21]. Since one is interested in the signal of the electrons of the first two sources, charm and beauty decays, it is useful to know the strength factors of the different sources, which are basically the relative contributions of them. In this plot, the signal of the various sources

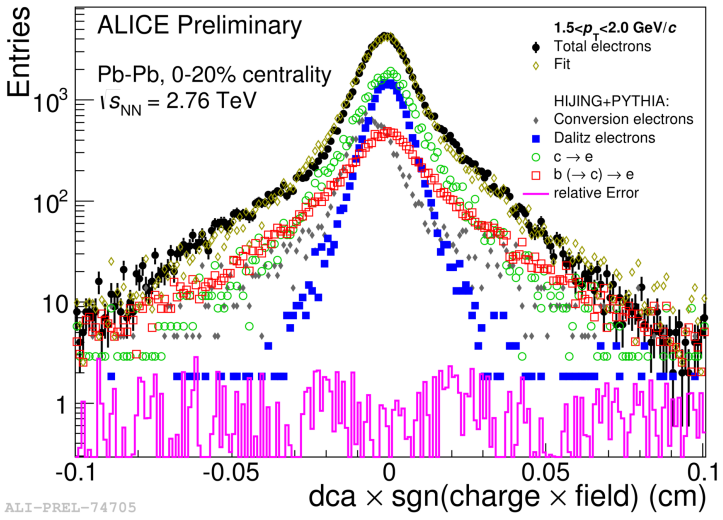


Figure 1.3: Impact parameter distribution of the electrons as a superposition of the distributions from several sources [9].

are so called templates, which are Monte Carlo simulations of full events and their detector response, multiplied with respective strength factors obtained from a fit. The fitting routine will be explained in more detail in chapter 2 [20, 21].

1.3 Statistical Methods

In this section some methods, which are used and dealt with in the chapters ahead are treated. From the common likelihood, which is used in maximum likelihood fits, according to Bayesian statistics a posterior distribution can be built. Then, a short introduction to Markov Chain Monte Carlo methods, which can be used to sample this posterior distribution, is given. At last the mathematical correct likelihood for the present problem of electron source separation is derived.

1.3.1 The Maximum Likelihood Method

A general problem in data analysis is the following: there is a vector of measurement data $\vec{x}$ and a theoretical model which describes the data. The model

depends on the free parameters $\vec{\lambda}$. In order to get the best estimate for $\vec{\lambda}$, one wants to know how well the data fits the model for a given set of parameters. A measure for this is the so called *likelihood* $L(\vec{\lambda}, \vec{x})$. The likelihood can be considered as the probability for a given set of parameters $\vec{\lambda}$ to get the measurement data $\vec{x}$ and hence equals the conditional probability:

$$L(\vec{\lambda}, \vec{x}) = p(\vec{x}|\vec{\lambda}) \tag{1.2}$$

It is important to mention that there is no information about the absolute probability $p(\vec{\lambda})$ of a given set of parameters in the likelihood.

A common technique to get the best estimate for $\vec{\lambda}$ is its variation until the likelihood is maximized. This procedure is called a *maximum likelihood fit*. Since a constant positive factor does not change the maximum of a function, the normalization of the likelihood is irrelevant for the position of the maximum. Regarding the N components x_i of $\vec{x}$ as different measurement points or equivalently as realizations of a random variable, the likelihood of the whole measurement $\vec{x}$ is the product of all N single probabilities to measure x_i with the parameters $\vec{\lambda}$:

$$L(\vec{\lambda}, \vec{x}) = \prod_{i=1}^{N} p(x_i|\vec{\lambda}) \tag{1.3}$$

Due to computational reasons, it is common to calculate the maximum of the logarithm of the likelihood, whose maximum is the same as for the absolute likelihood:

$$\log L(\vec{\lambda}, \vec{x}) = \sum_{i=1}^{N} \log p(x_i|\vec{\lambda}) \tag{1.4}$$

This definition of a likelihood is called the "unbinned likelihood", because every single measurement point is considered on its own.

However, there is another approach called the "binned likelihood". Hereby, the measurement points are inserted into a histogram and N is not the number of points any more but the number of bins in this histogram. From the theoretical model, the expected number of entries f_i in the i-th bin can be calculated, while the measurement data yields d_i entries in it. Hence, the probability to get d_i entries in the i-th bin, while f_i entries are expected, is

Poisson distributed with the mean value $\mu = f_i$:

$$p(d_i|\vec{\lambda}) = \frac{f_i^{d_i}\,\mathrm{e}^{-f_i}}{d_i!} \tag{1.5}$$

The total likelihood is then the product of this probability for all bins. The logarithm of it is

$$\log L(\vec{\lambda}, \vec{d}) = \sum_{i=1}^{N} d_i \log f_i - f_i \tag{1.6}$$

whereby the denominator $d_i!$ from formular 1.5 has been dropped, because it is a constant, which does not depend on the f_i. It should be mentioned that the f_i are dependent on $\vec{\lambda}$. Thus the fitting parameters are contained in the likelihood, so that they can be varied to maximize it.

1.3.2 Bayesian Statistics

In classical statistics, the probability of an event is a measure for the frequency with which the event would occur, if one repeated an infinite amount of trials. One can consider a set of several trials of a random variable X, distributed according to a probability distribution $p(X|\vec{\lambda})$, whereby $\vec{\lambda}$ is a set of parameters which on p depends. The maximum-likelihood method can be used to get the best estimate for $\vec{\lambda}$. This can be considered as equivalent to the problem of the section before, where there have been measurement data $\vec{x}$ instead of a random variable X. Afterwards, one can construct a confidence interval for each sample of X such that the true value of $\vec{\lambda}$ lies in 95% of all intervals created in this way. However, one has to note that this statement is not equivalent to the one that the true value lies in each interval with a probability of 95%, because in the classical frequentist's view the true value is not a random variable, yet only unknown.

In Bayesian statistics, the probability is considered as the degree of belief one has about the true value of $\vec{\lambda}$. Hence, although in the Bayesian view the true value is also not a random variable, it can be assigned a probability. Additionally, prior knowledge is attached to the experiment in the form of a factor $p(\vec{\lambda})$, which is simply called the "prior probability distribution". It can

be considered as the probability distribution of having a certain set of parameters $\vec{\lambda}$ without knowing about the measurement X. It is the 'personal belief' about the $\vec{\lambda}$, which can be formed through previous measurements, theoretical calculations or pure reason. Regarding the outcome of an experiment $p(X, \vec{\lambda})$ as fixed information, the so called "posterior distribution" can be calculated according to Bayes' theorem:

$$p(\vec{\lambda}|X) = \frac{p(X|\vec{\lambda}) \cdot p(\vec{\lambda})}{p(X)} \tag{1.7}$$

where $p(X|\vec{\lambda})$ is the likelihood and $p(X) = \int_{\vec{\lambda}} p(X|\vec{\lambda})p(\vec{\lambda})d\vec{\lambda}$ is a normalization constant, which does not depend on $\vec{\lambda}$. One can see that the posterior distribution is always a conditional probability depending on the knowledge available about the experiment. Full knowledge of the posterior distribution, which is gained by sampling it, has several advantages compared to the maximum likelihood method:

- Uncertainties for the best estimate can be calculated correctly by sampling the posterior distribution and determining the standard deviation of the sample. In many maximization algorithms, however, the uncertainties are calculated only approximately by considering the probability distribution referred to $\vec{\lambda}$ as Gaussian or instead of uncertainties a confidence interval is given.

- Prior knowledge can be used in the form of the prior distribution $p(\vec{\lambda})$. E.g., one could multiply a Heaviside step function

$$p(\vec{\lambda}) = \begin{cases} 0, & \vec{\lambda} < 0 \\ 1, & \vec{\lambda} \geq 0 \end{cases} \tag{1.8}$$

  to the likelihood in order to suppress negative values for $\vec{\lambda}$ or one could also use results from earlier measurements to improve the inference. In the maximum likelihood method, this could lead to a non-differentiable function, so that numerical maximization becomes problematic.

- The posterior distribution can be asymmetric. In this case, its expectation value delivers a better estimate than its maximum.

- Even for a high-dimensional $\vec{\lambda}$, efficient fitting algorithms are available through *Markov Chain Monte Carlo* (MCMC) methods [6].

1.3.3 Monte Carlo Methods

Sampling a function $f(x)$ in general means to consider it as a probability distribution, which presupposes that it is normalized to 1, and creating realizations of a random variable which is distributed according to $f(x)$. These realizations are the sample instances of $f(x)$ and are usually saved into an array or some other kind of list.

One rather simple way to create random variables according to a normalized probability function $f(x)$ is the *inverse cdf method*, whereby cdf stands for cumulative distribution function. The cdf of a probability density $f(x)$ is defined by:

$$F(x) = \int_{-\infty}^{x} f(x')dx' \tag{1.9}$$

In other words the cdf $F(x)$ is the total probability that a random variable X, which follows the probability density $f(x)$, takes on a value $X \leq x$. If one knows the cdf $F(x)$ of a probability density $f(x)$ and if $F(x)$ is invertible, it is easy to sample from $f(x)$, since one only has to be able to sample from a uniform distribution $U_{[0,1]}$ between 0 and 1. It can be shown that creating random variables according to $F^{-1}(U)$ delivers the wanted sample from $f(x)$. However, this method is only useful for uncomplicated distributions $f(x)$ as an exponential function, since in general it is not possible to construct $F(x)$ [13].

Another method is the so called *acceptance rejection sampling*, which is one of the simplest Monte Carlo methods. It requires a so called proposal distribution $prop(x)$, which is simply a probability density from which it is easy to sample from like a Gaussian or a uniform distribution. The proposal

distribution has to fulfill the property

$$c \cdot prop(x) \geq f(x) \; \forall \, x \tag{1.10}$$

whereby c is a constant and $f(x)$ is the known probability density one wants to sample from. A sample instance is created as follows:

1. Create a random number $v \sim prop(x)$ distributed according to the proposal distribution.
2. Build the acceptance ratio $\rho = \frac{f(v)}{c \cdot prop(v)}$.
3. Create a uniform distributed random number $u \sim U_{[0,1]}$ from 0 to 1.
4. If $u \leq \rho$ accept, else reject.
5. Repeat step 1.

The acceptance rejection method is quite useful for a low number of dimensions, since it can sample almost every function and the sample instances are completely independent. Yet, for a high dimensional function the "space" between $c \cdot prop(x)$ and $f(x)$ grows exponentially with the number of dimensions, which is referred to as *curse of dimensionality* [4], and hence the acceptance rate converges to zero, which raises the necessary steps or calculation time beyond all measure.

For high dimensional problems, another Monte Carlo method shows its strength, which is the so called *Metropolis Algorithm* (MA). Since the sample instances created by it represent a Markov chain, which is a set of random numbers, where one number always only depends on the number generated before, the MA is a *Markov Chain Monte Carlo* method. To sample from a function another function, which is easy to sample from, is needed to serve as a proposal distribution $prop(x)$. However, a created proposal is now not the value to look at anymore, as in case of the acceptance rejection method, but it is the step size which the Markov chain performs. The only constraint for the proposal distribution in the MA is that it is symmetric for negative and positive values. A common choice for $prop(x)$ is a Gaussian, but also a

uniform distribution is possible. To create sample instances from the function $f(x)$ the following procedure is applied:

1. Choose a starting position and set it as your old position $x_{old} = x_{start}$.
2. Generate a random number $p \sim prop(x)$ distributed according to the proposal distribution serving as a proposal.
3. Go to the new position $x_{new} = x_{old} + p$.
4. Build the acceptance ratio $\rho = \min\left(1, \frac{f(x_{new})}{f(x_{old})}\right)$.
5. Create a uniform distributed random number $u \sim U_{[0,1]}$ from 0 to 1.
6. If $u < \rho$ accept, else reject.
7. Repeat step 2.

Hereby it is to mention, that accepting a step means to save x_{new} into the list of sample instances and move to the new position or in other words set $x_{old} = x_{new}$. Rejecting means to save x_{old} into the list of sample instances and remain at the current position, which is for example the case in Figure 1.8. This differs from the acceptance rejection method, where nothing is saved into the list of sample instances until a step is accepted [13]. In order to visualize this procedure an example is given in Figures 1.4 to 1.9:

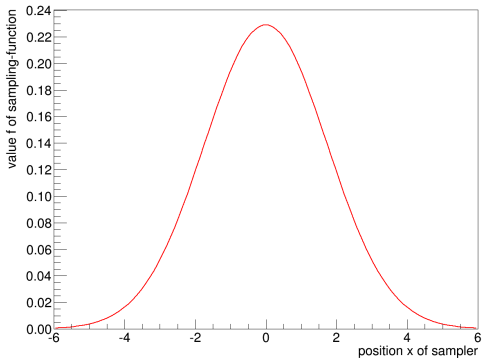
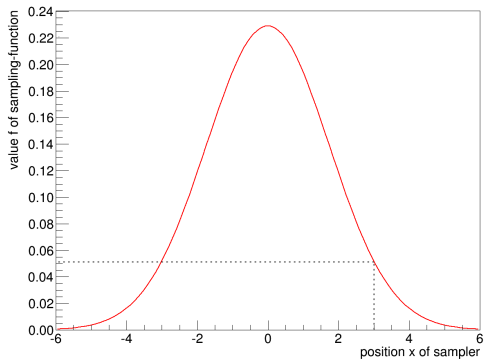
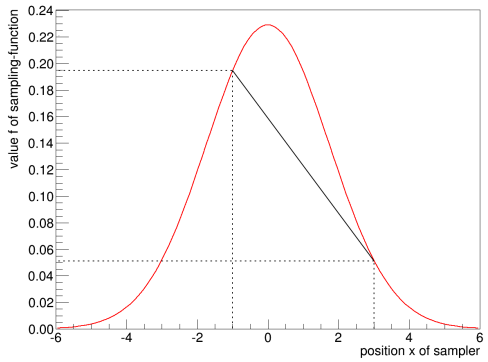


Figure 1.4: Gaussian function sampled with the Metropolis algorithm.



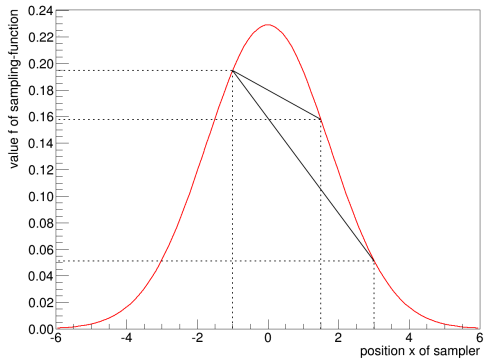
$$x_{start} = 3$$

Figure 1.5: A starting point is chosen.



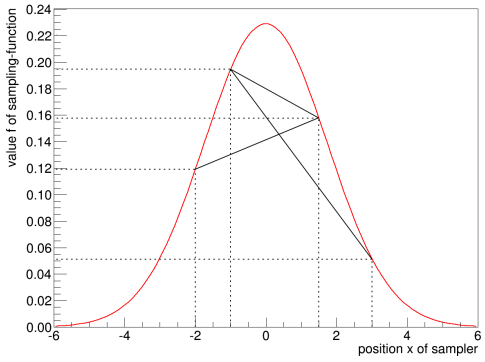
$$x_{old} = x_{start} = 3$$
$$p = -4$$
$$x_{new} = x_{old} + p = -1$$
$$\rho = \min\left(1, \frac{f(x_{new})}{f(x_{old})}\right) = \min(1, 3.75) = 1$$
$$u = 0.9$$
$$\rho > u \Rightarrow \textbf{accept}$$

Figure 1.6: For $f(x_{new}) > f(x_{old})$ the step is always accepted.



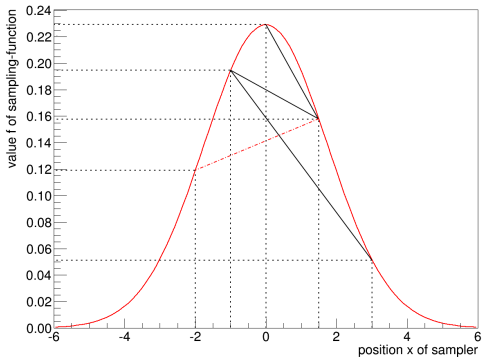
$$x_{old} = -1$$
$$p = 2.5$$
$$x_{new} = x_{old} + p = 1.5$$
$$\rho = \min\left(1, \frac{f(x_{new})}{f(x_{old})}\right) = \min(1, 0.81) = 0.81$$
$$u = 0.4$$
$$\rho > u \Rightarrow \textbf{accept}$$

Figure 1.7: For $f(x_{new}) < f(x_{old})$ it depends on u whether a step is accepted.



$$x_{old} = 1.5$$
$$p = -3.5$$
$$x_{new} = x_{old} + p = -2$$
$$\rho = \min\left(1, \frac{f(x_{new})}{f(x_{old})}\right) = \min(1, 0.75) = 0.75$$
$$u = 0.8$$
$$\rho < u \Rightarrow \textbf{reject}$$

Figure 1.8: For $\rho < u$ the step is rejected.



$$x_{old} = 1.5$$
$$p = -1.5$$
$$x_{new} = x_{old} + p = 0$$
$$\rho = \min\left(1, \frac{f(x_{new})}{f(x_{old})}\right) = \min(1, 1.45) = 1$$
$$u = 0.6$$
$$\rho > u \Rightarrow \textbf{accept}$$

Figure 1.9: For $\rho > u$ the step is accepted again.

Though for low-dimensional functions, acceptance rejection sampling can be much more efficient because the sample instances are independent, for high-dimensional functions the Metropolis algorithm is the tool of choice. The reason for this is that one does not "throw away" that many steps as in the acceptance rejection sampling due to the big space between the proposal function and the function one wants to sample. Although the number of accepted steps depends on the dimension, one can simply adjust the width of the proposal function to change it such that one has maximum efficiency, which is rather difficult in acceptance rejection sampling. Hereby, 'Maximum efficiency' means that the algorithm shows least variance for the same number of steps. However, difficulties in the MA arise, if the optimum widths of the proposal distributions

for maximum efficiency are not the same in every dimension, but differ from each other for the individual dimensions. Hence, it is a tough task to align these widths correctly for every dimension, which has to be dealt with.

1.3.4 Autocorrelation

In the ideal case, the sample instances of the MA represent a Markov chain. Hence, each of them should only depend on the one before. In reality, however, a small width of the proposal distribution $prop(x)$ leads to a small proposal p. In that case, it will often be $f(x_{new}) \approx f(x_{old})$ so that steps will be accepted more frequently than if the sample instances were independent for more than one step between them. This phenomenon constrains the efficiency of the algorithm, since in order to proceed through the whole function $f(x)$ one has to go much more steps than with independent sample instances. On the other hand, if one chooses a large width for $prop(x)$, almost everytime one will have $f(x_{new}) \approx 0$ so that most of the steps are rejected. This is even worse due to two reasons. Firstly, the sample instances are not independent either, since they often stay at the same position x_{old}. Secondly, since almost all steps are rejected, it takes many steps to compensate for this. A useful value to summarize the impact of the width of the proposal distribution is the so called *Acceptance Rate* (AR), which is defined as the number of accepted steps divided by the number of total steps. This should not be mistaken for the acceptance ratio ρ in the definition of the MA. Eventually, the aim is to find the AR with best efficiency, which means to find the AR for which the variance of the sample instances is minimal for a constant number of total steps. Hereby, the algorithm still has to converge to f. Otherwise one could choose a proposal width with an AR of zero, which would lead to a variance of the sample instances of zero, because the algorithm would always stay on the same position.

A good measure for checking the independence of the sample instances is the correlation coefficient between the sample instances also known as *autocorrelation function*. It is defined by the covariance of those sample instances that are a given number of steps apart from each other. In general, the covariance

of a set of two random variables a and b is defined as [6]

$$Cov(a,b) = E[(a-\bar{a})(b-\bar{b})] \tag{1.11}$$

whereby $\bar{a}$ and $\bar{b}$ are the expectation values of a and b. Therefrom, one defines the correlation coefficient of a and b as the normalized covariance:

$$Corr(a,b) = \frac{Cov(a,b)}{\sigma_a \sigma_b} \tag{1.12}$$

whereby σ_a and σ_b are the standard deviations of a and b. The correlation coefficient always fulfils $-1 \leq Corr(a,b) \leq 1$. Hereby, $Corr(a,b) \approx 1$ means that a and b show a strong positive linear correlation, while $Corr(a,b) \approx -1$ means they are strongly negative linearly correlated. According to [13], the autocorrelation function of a set of random variables or a sample of a function X is defined as

$$Auto(lag) = \frac{C(lag)}{C(0)} \tag{1.13}$$

$$\text{with} \quad C(lag) = \frac{1}{N-lag+1} \sum_{i=0}^{N-lag} (X_i - \bar{X})(X_{i+lag} - \bar{X})$$

whereby N is the size of the sample or the number of sample instances, X_i is the i-th sample instance of X, and $\bar{X}$ is the mean of all sample instances. The lag is the number of steps two sample instances are apart from each other. In order to understand this formula, one can imagine that from the sample instances of X pairs X_i and X_{i+lag}, which are lag steps apart, are created and considered as two sets of a random variable. The function $C(lag)$ returns the covariance of these two sets. For $lag = 0$ it returns simply the variance of the sample X. Therefore, the autocorrelation function is just the correlation coefficient of all sample instances with a distance of lag steps.

In order to determine how large the AR should be, it is interesting to know the autocorrelation of the MCMC sample for different lags. This allows to estimate how many steps from one sample instance are needed to get an independent one. In other words, this determines how many steps in general are

needed to get a representative sample of the function, which should be many more than the amount one needs to get independent sample instances.

1.3.5 Likelihood For The Separation Of Electron Sources

In this last section of the Introduction chapter, the likelihood L for the separation problem related to Figure 1.3 is deduced. From this likelihood, the posterior distribution is built according to Formula 1.7. In the chapters ahead, this posterior is sampled with the MA in order to get an estimate for the values of the strength factors.

One can consider to have measurement data recorded by the detector in the form of a histogram. In Figure 1.3, these are the black dots, which represent the impact parameter distribution of the detected electrons. The number of entries in the i-th bin shall be d_i. Additionally, we know the distributions of the M different sources j, which are available in the form of Monte Carlo simulations of full events and their detector response. These Monte Carlo simulations are used to create a histogram for each electron source, the so called *templates*. Therefrom, according to [14], we can calculate the estimated number of entries in the i-th bin as

$$f_i = N_D \sum_{j=1}^{M} \frac{P_j a_{ij}}{N_j} \tag{1.14}$$

where $N_D = \sum_{i=1}^{N} d_i$ is the total number of entries in the histogram with N bins containing the measurement data, a_{ij} is the number of entries in the i-th bin of the template for the j-th source, and $N_j = \sum_{i=1}^{N} a_{ij}$ is the total number of entries in the template for the j-th source. P_j is the strength factor of the j-th source and one has as constraint $\sum_{j=1}^{M} P_j = 1$. For calculations it is convenient to normalize the P_j to $p_j = P_j N_D / N_j$ such that 1.14 becomes

$$f_i = \sum_{j=1}^{M} p_j a_{ij}. \tag{1.15}$$

Next, one could try to minimize

$$\chi^2 = \sum_{i=1}^{N} \frac{(d_i - f_i)^2}{d_i} \tag{1.16}$$

to get an estimate for the p_j values. However, this approach assumes a Gaussian distribution of the d_i around the mean value of f_i. Although this is approximately true for large d_i, the distribution for low d_i is not Gaussian anymore but Poissonian.

Hence, according to Formula 1.6 the logarithm of the likelihood is

$$\log L_{prel} = \sum_{i=1}^{N} d_i \log f_i - f_i \tag{1.17}$$

where the index 'prel' has been added, because this is not the final likelihood but just a preliminary one. It would be correct if the templates had infinite statistics, yet they do not.

The templates a_{ij} are sampled and therefore underlie statistical fluctuations. Admittedly, these fluctuations are damped by the factor N_D/N_j according to Formula 1.14, but they still have to be taken into account. Therefore, the likelihood has to be modified. Considering the template of source j, the Monte Carlo simulation yields a value a_{ij} for the i-th bin. This value is a random variable distributed according to a Poisson distribution with some unknown true value A_{ij}. Hence, the probability that the template yields a_{ij} in the i-th bin is

$$p(a_{ij}|A_{ij}) = \frac{A_{ij}^{a_{ij}}\, \mathrm{e}^{-A_{ij}}}{a_{ij}!}. \tag{1.18}$$

These A_{ij} are additional free parameters in the likelihood. This final likelihood is now the product of the probabilities over all bins that the bin i has a certain bin content d_i, while one expects f_i, multiplied with the product of the probabilities over all bins and sources that the bin of a template has the

bin content a_{ij}, while one expects A_{ij}:

$$L = \prod_{i=1}^{N} \left(\frac{f_i^{d_i} \, \mathrm{e}^{-f_i}}{d_i!} \prod_{j=1}^{M} \frac{A_{ij}^{a_{ij}} \, \mathrm{e}^{-A_{ij}}}{a_{ij}!} \right). \tag{1.19}$$

The logarithm is:

$$\log L = \sum_{i=1}^{N} \left[d_i \log f_i - f_i + \sum_{j=1}^{M} \left(a_{ij} \log A_{ij} - A_{ij} \right) \right] \tag{1.20}$$

[14] whereby the denominators $d_i!$ and $a_{ij}!$ have been dropped again, because they do not depend on the fit parameters. Since the expected bin content is now

$$f_i = \sum_{j=1}^{M} p_j A_{ij}, \tag{1.21}$$

the free parameters of this equation are M times $p_j \sim P_j$ and N times A_{ij}. This likelihood has $M \times (N + 1)$ dimensions. Having for example a number of bins of $N = 200$ and $M = 4$ different electron sources, one obtains a 804-dimensional sampling problem. This is the reason why Markov Chain Monte Carlo is appropriate for the electron separation.

2 Approaching The Metropolis Algorithm

In this chapter, the Metropolis Algorithm (MA) shall be applied to simple functions, like a 1-dimensional and 1000-dimensional Gaussian. By means of whose, some properties, e.g. the convergence of the algorithm or its autocorrelation, and some concepts, e.g. the burn in and adaptive proposal, shall be dealt with in order to obtain some general insights about the MA.

2.1 Sampling A One-Dimensional Gaussian

2.1.1 Definition Of Sampling Function And Proposal Distribution

For convenient application of the MA, a ROOT class which can be used to sample one-dimensional functions has been implemented. This class is used to apply the MA to a Gaussian function

$$f(x) = \frac{4}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{2.1}$$

with $\mu = 3$ and $\sigma = \sqrt{2}$ and sample it in order to calculate the expectation value $\bar{x}$ of the sample, which tends to μ for a large number of sample instances. The Gaussian function in Formula 2.1 could represent a posterior distribution e.g. of a fitting problem like the one in chapter 3. This procedure intends to check in general the proper functioning of the MA and the implemented class.

For the proposal distribution also a Gaussian function

$$prop(x) = \frac{1}{\sqrt{2\pi\sigma_{prop}^2}} \exp\left(\frac{-x^2}{2\sigma_{prop}^2}\right) \tag{2.2}$$

with the proposal width $\sigma_{prop} = \sqrt{3}$ is used. For a one-dimensional Gaussian target distribution an Acceptance Rate (AR) of $\approx 44\%$ provides best efficiency [1]. Hence, σ_{prop} has been set to a value which does fulfill this.

2.1.2 Dependence On Starting Position

According to Section 1.3.3, the MA requires a pre-selected value x_{start} functioning as a starting position. If this starting position was in a region of low probability, it would need many steps until the Markov chain is in the region of high probability. If the number of steps is not large enough, this leads to a shift of the sample towards the starting point, which results in expectation value that does not fit the true value. Therefore, one can drop the first B sample instances so that the expectation value is calculated without them. Thus, the starting position becomes unknown without using any knowledge about the posterior distribution. B is called the *burn in* of the sample. Since one has less statistics with a larger B, it should be as low as possible. Although the optimum value for the burn in is arbitrary, in this thesis, it is set to constitute not more than 25% of the number of sample instances, because otherwise the reduction of the statistics is too significant [13]. In order to test whether the sample is independent on the starting position, one can run the MA several times and check if the results are the same.

2.1.3 Results

In the Figures 2.1 - 2.5 the sampling function $f(x)$ and the distribution of the sample instances normalized to the function is shown. With an increasing number of steps this distribution corresponds more and more to the sampling function.

In Table 2.1 the results of a Markov Chain Monte Carlo (MCMC) fit with

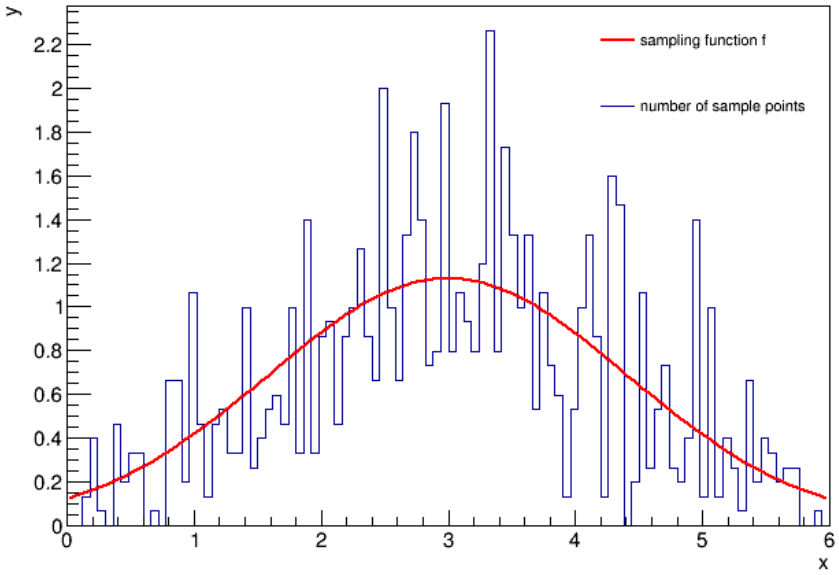


Figure 2.1: Sample of Gaussian with 1000 steps.

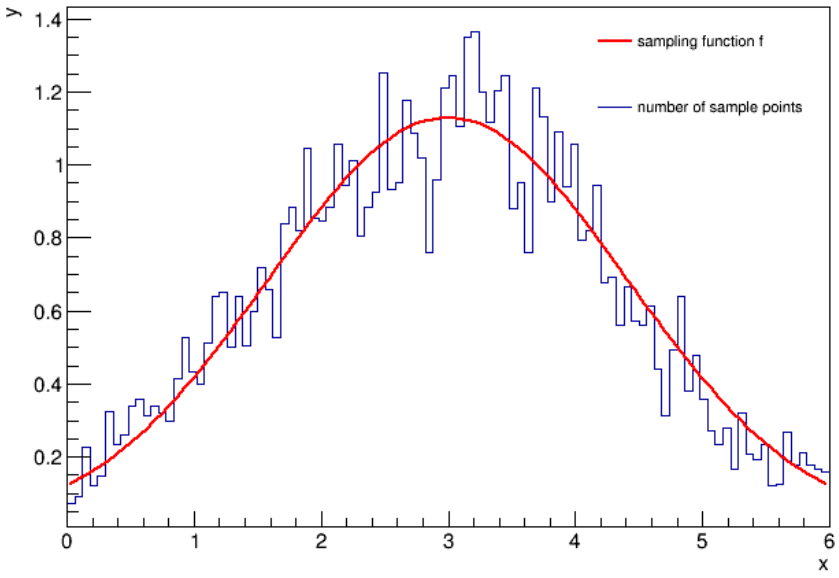


Figure 2.2: Sample of Gaussian with 10000 steps.

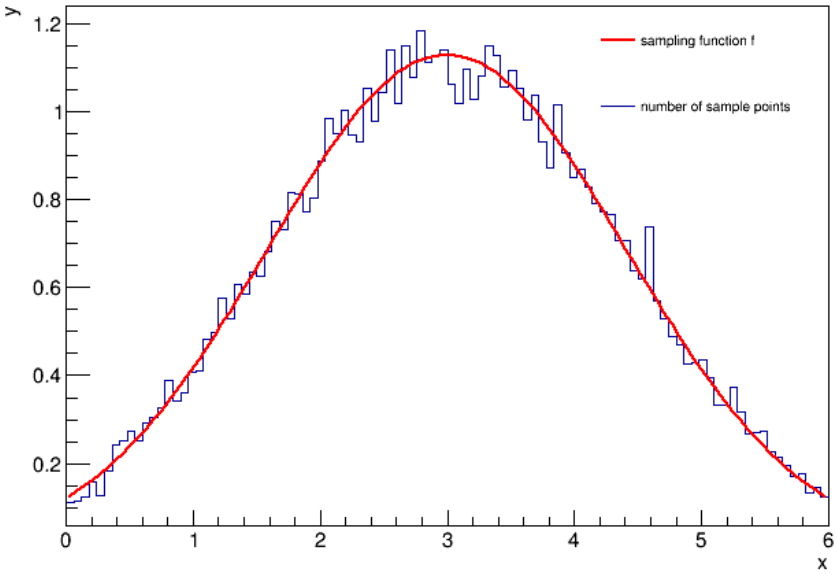


Figure 2.3: Sample of Gaussian with 10^5 steps.

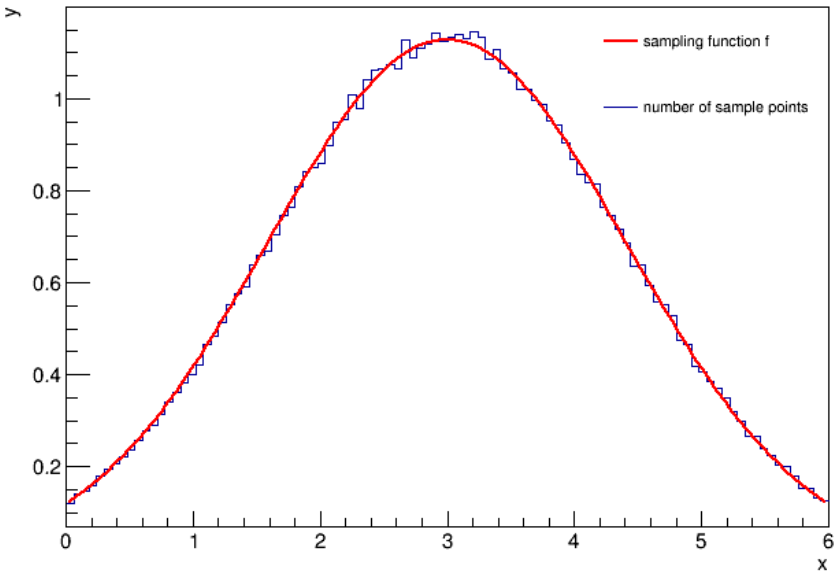


Figure 2.4: Sample of Gaussian with 10^6 steps.

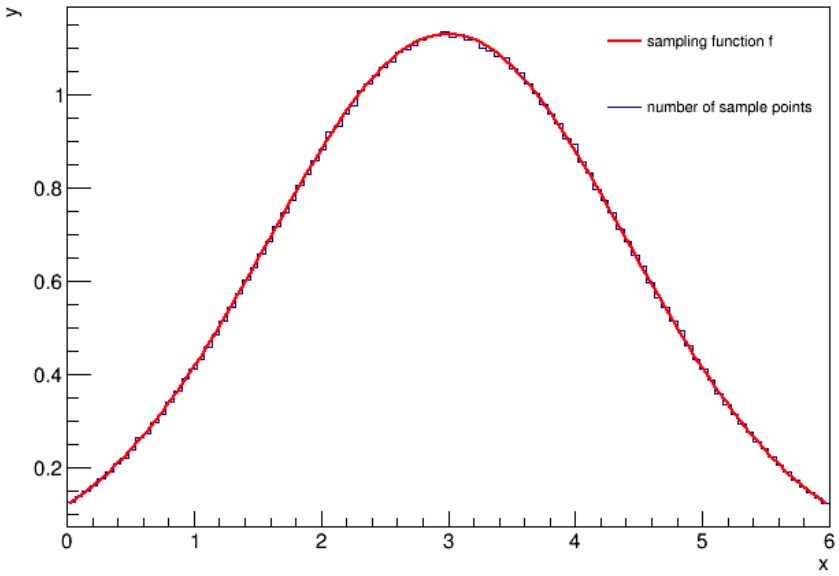


Figure 2.5: Sample of Gaussian with 10^7 steps.

10^7 steps plus $2.5 \cdot 10^6$ burn in steps and an AR of 48.3% as well as a maximum likelihood fit on $f(x)$ are presented. The maximum likelihood fit, which was run with the ROOT class TMinuit, yields a value with less deviation from the real value $\mu = 3$ and is calculated in less time. Hereby, it is important to mention that the computation, as all computations in this bachelor thesis, was run on the server 'alice-serv9', which is one of the storage nodes of ALICE in the Physikalisches Institut Heidelberg. However, the mean value $\bar{x} = 2.997$ found by the MA is also quite close to μ and the uncertainty $\sigma_x = 1.414$, which is the standard deviation of the sample instances, is also very close to the expected standard deviation $\sigma = \sqrt{2}$ of the sampling function. The uncertainty of the maximum likelihood fit is calculated through the gradients of the fit parameters by TMinuit. In general, the posterior distribution does not have to be symmetric and hence the maximum does not always coincide with the expectation value. In this example, however, it does. One can see that for low dimensional target distributions the MA does not have the best efficiency, which means other algorithms can be more precise with the same number of steps. Yet, it provides a correct estimate for the error of the fit.

fit method	fit value for μ	uncertainty	calculation time [s]
MCMC	2.997	1.414	18.8
maximum likelihood	3	1.883	$1.3 \cdot 10^{-3}$

Table 2.1: Results of MCMC and maximum likelihood fit on Gaussian.

In conclusion, we see that the algorithm converges correctly to the sampling function.

2.2 Sampling An N-Dimensional Gaussian

The MA can easily be generalized for N-dimensional sampling functions by using N proposal functions and thus creating a vector of proposals. A step is accepted as usual, if $u < \min\left(1, \frac{f(\vec{x}_{new})}{f(\vec{x}_{old})}\right)$, whereby the vector arrows symbolise the N dimensions of x and are not drawn from now on.

2.2.1 Definition Of Sampling Function

In order to investigate the convergence and other properties of the MA for high dimensions, the ROOT class is modified such that it is able to sample from N-dimensional target distributions. A 1000-dimensional Gaussian function is sampled with it. The concrete form of the Gaussian is

$$f(x) = \prod_{i=1}^{1000} \left[\frac{4}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(x_i - \mu_i)^2}{2\sigma_i^2} \right) \right] \tag{2.3}$$

whereby x_i is the i-th component of the vector x. μ_i and σ_i are the mean and width for the single dimensions and in this case one has $\mu_i = 3$ and $\sigma_i = \sqrt{2}\ \forall\, i$.

The N proposal functions are all equal to Formula 2.2 with the same σ_{prop} for all dimensions. The value for σ_{prop} is determined by a procedure called *adaptive proposal*.

2.2.2 Adaptive Proposal

Since the proposal width has to be adjusted such that the algorithm has maximum efficiency, it is a good idea to do this automatically. One can consider sampling an N-dimensional, spherical normal distribution, i.e. a normal distribution with the same width in each dimension, with the MA. It can be shown, that the AR with the best efficiency for this sample tends to $\approx 23\%$ for $N \to \infty$ [1]. Hence, one can periodically adjust the proposal width until the AR reaches its optimum value. Since in this case this is done only during the burn in phase, this is just a pseudoadaptive strategy. For the regulating process, the recommendation of [13] is used, after which the proposal width σ_{prop} is adjusted in the following way: after every step t, it is replaced by a modified proposal width

$$\sigma_{prop,t+1} = \sigma_{prop,t} \exp\left(\frac{\alpha_t - \alpha^*}{t^{0.7}}\right) \tag{2.4}$$

whereby α_t is the AR until the t-th step, i.e. the number of accepted steps divided by t, and α^* is the optimum AR (in this case 25%). This method increases σ_{prop} for $\alpha_t > \alpha^*$ and decreases it for $\alpha_t < \alpha^*$. In order to get a faster adjustment to the optimum proposal width, additionally on every hundredth step an addend is added to $\sigma_{prop,t}$, which is proportional to the deviation of the current AR from the optimum one α^*. In detail, this looks the following:

$$\text{For } t \bmod 100 = 0\text{: } \sigma_{prop,t} \to \sigma_{prop,t}\left(1 + \frac{\alpha' - \alpha^*}{10}\right) \tag{2.5}$$

whereby the factor $\sigma_{prop,t}/10$ scales the addend to one order below the actual proposal width. The t mod 100 has to be applied, because a certain amount of sample instances is needed, to calculate the current AR α', which is the number of accepted steps from the last 100 sample instances divided by 100. This procedure can be considered as some kind of proportional-integral controller, whereby the proportional part corresponds to Formula 2.5 and the integral part corresponds to Formula 2.4. Yet, one has to consider that the regulating process is not proportional to the integral of the AR but only to a quantity

that is dependent on the integral. Thus, the optimum AR can be successfully adjusted.

However, problems arise when the optimum proposal widths of the various dimensions differ from each other. Indeed, this method of adaptive proposal will always reach the desired AR, yet, for multidimensional functions the AR is not the only measure for a good efficiency, but furthermore the correct proposal width for each dimension has to be determined. Hence in Chapter 3, when the electron separation problem is dealt with, adaptive proposal is not applied.

2.2.3 Results

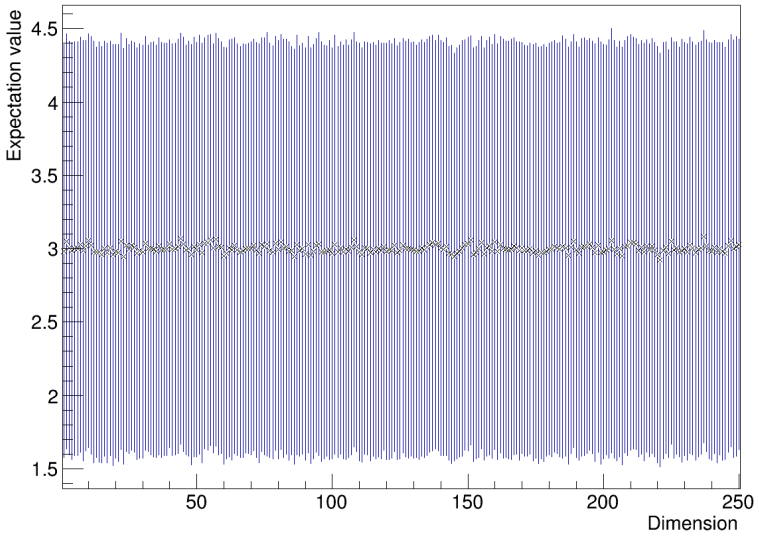


Figure 2.6: Expectation value of sample for dimensions 1-250. The error bars indicate the standard deviation of the sample instances.

In the Figures 2.6 - 2.9, the expectation values of the sample instances for each dimension are shown. They all accumulate around the real values of $\mu_i = 3 \ \forall i$. The error bars represent the standard deviations of the sample instances for each dimension, which also fit the real values $\sigma_i = \sqrt{2} \ \forall i$. To summarize the results quantitatively, the deviations of the fitted expectation values and standard deviations to their real values for each dimension are

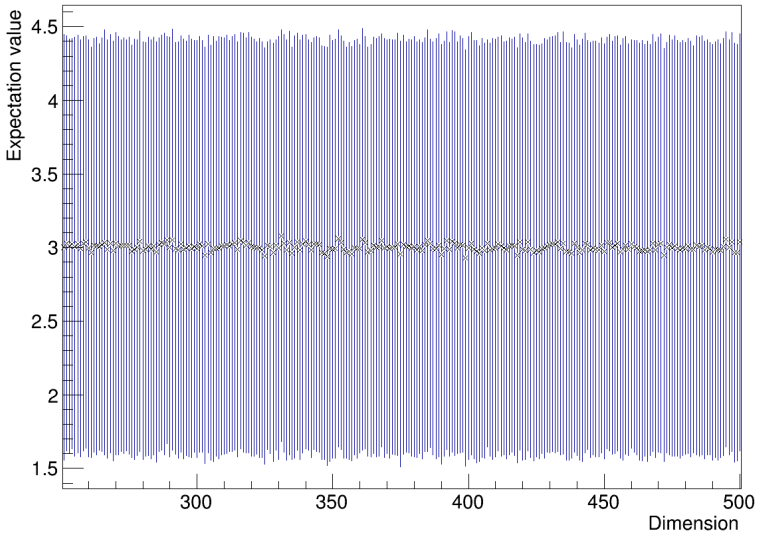


Figure 2.7: Expectation value of sample for dimensions 251-500. The error bars indicate the standard deviation of the sample instances.

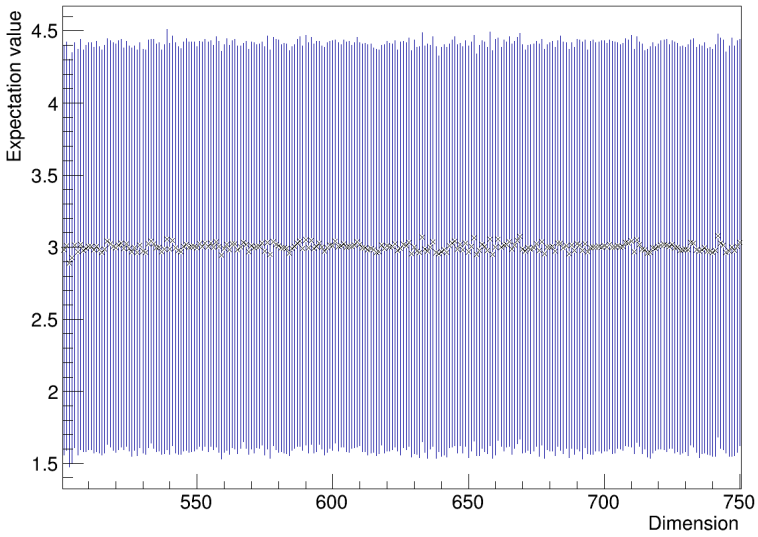


Figure 2.8: Expectation value of sample for dimensions 501-750. The error bars indicate the standard deviation of the sample instances.

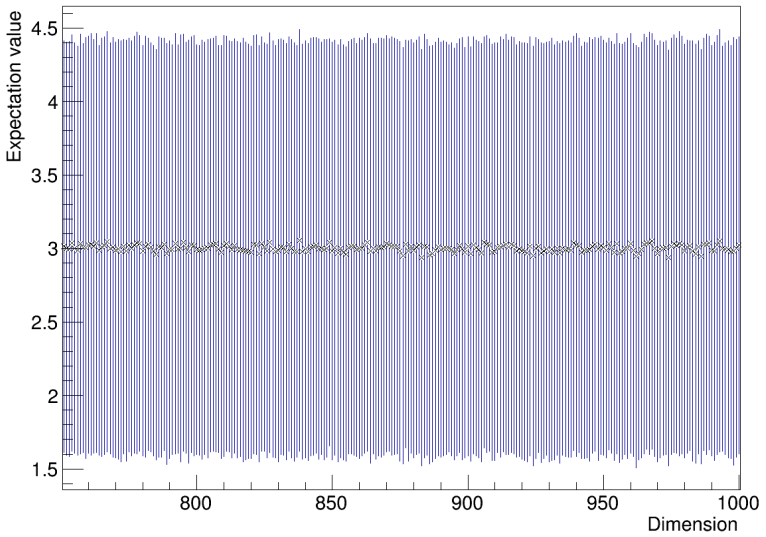


Figure 2.9: Expectation value of sample for dimensions 751-1000. The error bars indicate the standard deviation of the sample instances.

calculated and averaged. In detail this means:

$$Dev_{\mu}(x) = \frac{1}{N}\sum_{i=1}^{N}|\bar{x}_i - \mu_i| \tag{2.6}$$

$$Dev_{\sigma}(x) = \frac{1}{N}\sum_{i=1}^{N}|\sigma_{x_i} - \sigma_i| \tag{2.7}$$

whereby $N = 1000$ is the number of dimensions and $\bar{x}_i$ and σ_{x_i} are the mean value and the standard deviation of the sample instances for the i-th dimension. These deviations and some other quantities are given in Table 2.2.

Since the standard deviation σ_x of the sample instances is approximately $\sqrt{2}$ for all dimensions, the uncertainty of the mean value for each individual dimension can be estimated as

$$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{N-1}} \approx \sqrt{\frac{2}{10^7 - 1}} \approx 0.00045. \tag{2.8}$$

One has $Dev_{\mu}(x) = 0.0011 < 3\sigma_{\bar{x}} \approx 0.0013$, which means that most expec-

quantity	value
number of steps	10^7
calculation time	32441 s
AR	26.29%
σ_{prop} (proposal width)	0.102
μ (real expectation value)	3
σ (real standard deviation)	$\sqrt{2}$
$Dev_\mu(x)$ (mean deviation from μ)	0.0011
$Dev_\sigma(x)$ (mean deviation from σ)	0.00968

Table 2.2: Results of MCMC fit applied to a 1000-dimensional Gaussian.

tation values $\bar{x}_i$ do not deviate significantly from $\mu_i = 3\ \forall i$. Hence, the fitted parameters describe the sampling function appropriately.

2.2.4 Autocorrelation

Since the proposal width is quite small compared to the width of the posterior distribution, one can assume that the autocorrelation is very high. In order to test this, the autocorrelation function $Auto(lag)$ of a sample of a 1000-dimensional Gaussian function is calculated according to Formula 1.13 for a representative number of dimensions and lags, i.e. the distance in steps between two sample instances (see section 1.3.4). The function is shown in Figure 2.10.

Even for $lag = 1000$ the correlation coefficient is ≈ 0.5, while it should be around zero for low autocorrelation. The problem with this large autocorrelation is that even sample instances which are 1000 steps apart are correlated. Hence, in order to get for example ten times to every position in the sampling function, one has to fulfil at least 10×1000 steps. However, the AR of $\approx$ 26.29 % in this calculation is close to the theoretical optimum AR of $\approx$ 23 % [1], which is why there is no way to reduce the autocorrelation without reducing the efficiency of the algorithm. Yet, this theoretical optimum AR is valid only for spherical normal problems, which does not apply for the sampling function in Chapter 3. Hence an AR of 23 % is not mandatory, yet one can assume that it is desirable to have an AR which is in the same order of

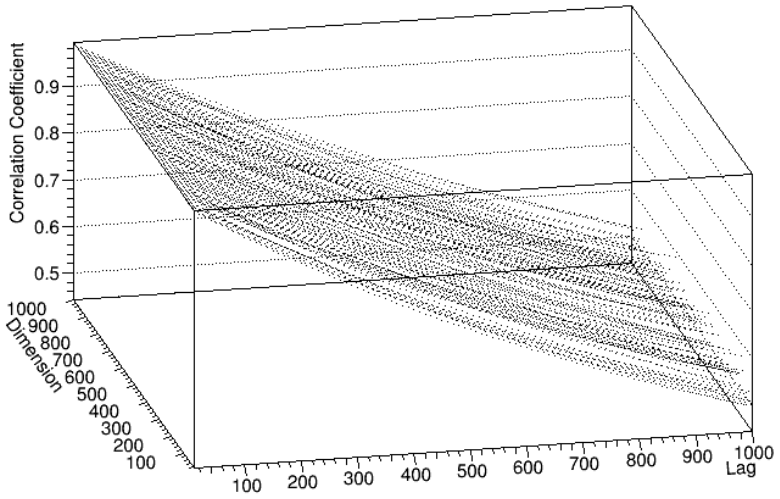


Figure 2.10: Autocorrelation function for several dimensions.

magnitude.

3 Applying The Metropolis Algorithm

In this chapter, the likelihood for the electron separation problem shall be implemented by calculating the exponential of Formula 1.20. The aim is to find the strength factors of the different sources such that their distributions constitute the total distribution as a superposition. This is done for a set of realistic Monte Carlo data, which is created through simulations of full events and their detector response. The simulation data represents the true impact parameter distributions of the electrons. The templates and the pseudo measurement data are created from this realistic Monte Carlo simulations through sampling. The likelihood is multiplied with the prior in Formula 1.8 to form a posterior distribution. Since neither the strength factors nor the entries in the histogram with the impact parameter distribution can be negative, using this Heaviside function as prior is appropriate. In detail, this is done by setting the logarithm of the likelihood to $-\infty$ if one of the A_{ij} or P_j is smaller than zero. Thus, the likelihood becomes zero and the step is rejected. The posterior is then sampled with the MA in order to get an estimate for the strength factors P_j. At last, the sample is investigated and tested for consistency with the true values.

3.1 Adjustment Of the MCMC fit

3.1.1 Forming The Posterior Distribution

The measurement data for this analysis are not real detector data but data created through Monte Carlo simulation because to assess the algorithm, a

sample with known true values is needed. The original, simulated measurement data are shown in Figure 3.1. In the three-dimensional histogram one can see the impact parameter and transverse momentum distribution of the electrons from different decay types.

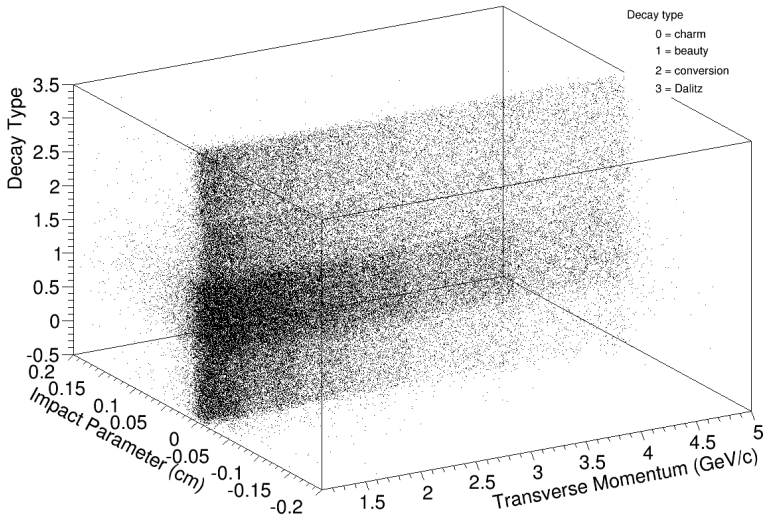


Figure 3.1: Original Monte Carlo data for MCMC fit. Data from [11].

Two cuts at 1.1 GeV/c and 5 GeV/c are applied to the data so that only electrons with a transversal momentum between these two values are taken into account. The lower boundary at 1.1 GeV/c is set due to the worse resolution at low p_T and due to stronger bending of the trajectories, which leads to a wider conversion electron distribution. The higher boundary is set to get higher statistics in this analysis. Then, the histogram is divided into 4 histograms, one for each decay origin group, and every single one of them is projected on the impact parameter axis by integrating over the transverse momentum. The several decay types are charm electrons, the beauty electrons, which one is interested in, conversion electrons produced through pair creation of photons in the detector material, and Dalitz electrons, which are primary electrons that are mainly created through Dalitz decays of pions [21]. Thus, for every decay type a one-dimensional histogram representing the impact parameter total distribution of the electrons is obtained. The histograms are normalized

by the number of entries. From these normalized histograms, the templates are built by sampling these. The sampling method does not matter here, hence the MA is used. Additionally, the normalized histograms are multiplied with arbitrary strength factors P_j, which are self-invented and should sum up to unity, and added up such that a superposition of the histograms is created. By sampling this superposition the pseudo measurement data is obtained. In Figure 3.2 the superposition representing the 'real physics' in this example and the normalized histograms, which are used to build the templates, multiplied with the self-invented, true strength factors P_j are shown.

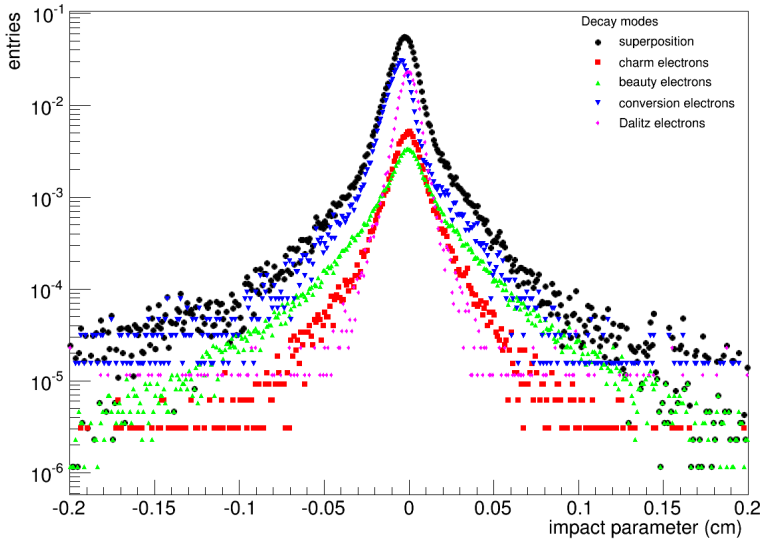


Figure 3.2: Superposition of normalized histograms and normalized histograms multiplied with the real strength factors. Data is taken for $1.1 < p_T < 5.0$ GeV/c.

Having created the pseudo measurement data and the templates, one can build the likelihood via calculation of the exponential its logarithm which is expressed in Formula 1.20 and multiply it with the Heaviside function from Formula 1.8 serving as prior in order to get the posterior distribution. Since in this analysis, there are $N = 200$ bins and $M = 4$ sources, the posterior distribution is a 804-dimensional function, whereby the 4 parameters p_j are the ones, we are really interested in (see section 1.3.5). As in this analysis, the

number of steps for sampling the pseudo measurement data N_D is the same as for sampling the templates $N_j = N_D = 10^6$, due to $p_j = P_j N_D / N_j$ one has $p_j = P_j$. Hence, in the following the strength factors are referred to as p_j.

3.1.2 Adjusting The Proposal Width

Since the expansion of the posterior distribution is now very different for the various dimensions, one of the most important tasks for successful MCMC sampling is the correct adjustment of the proposal widths. While for the p_j, which sum up to unity, the major part of the posterior distribution is at values of the order of 0.1, it differs tremendously for the A_{ij}. There are bins with an expected $A_{ij} \approx 0$ and some with many entries. As a result the optimum proposal widths differ strongly, too.

In order to ease this problem, one can consider the ansatz that a good proposal width should be proportional to the standard deviation of the posterior distribution in each dimension. Since this is unknown until the sampling of the posterior distribution is completed, another simplification has to be done. For a large number of sample instances N_j of the template sample, one can expect that $a_{ij} \approx A_{ij}$. Since the a_{ij} are Poisson distributed with the mean value A_{ij}, the standard deviation of the a_{ij} equals $\sqrt{A_{ij}}$. The simplification is now to invert this by considering the standard deviation of the A_{ij} as being equal to $\sqrt{a_{ij}}$. Even though this might not be accurate, it is sufficient to demand that the standard deviation of the A_{ij} is only proportional to $\sqrt{a_{ij}}$, since a proportional factor β is multiplied to the latter term anyway. Additionally, it is normally enough to have a proposal width of the correct order of magnitude even though it is not absolutely correct. Hence, for the p_j a standard deviation of the order 0.1 is assumed. For the proposal width, this is multiplied with the same factor β as for the A_{ij} such that the proposal widths scale equally.

One further difficulty with this method is that there may be bins with $a_{ij} = 0$ but $A_{ij} \neq 0$. This would result in a proposal width of 0 such that this value would not be fitted. Therefore a little offset of the order 1 is added to $\sqrt{a_{ij}}$ before multiplying these values with β. The final proposal widths are thus

given as

$$\sigma_{prop,ij} = \beta(\sqrt{a_{ij}} + 1) \text{ for the parameter } A_{ij} \tag{3.1}$$

$$\sigma_{prop,j} = \beta \cdot 0.08 \text{ for the parameter } p_j \tag{3.2}$$

whereby for $\sigma_{prop,j}$ the factor 0.08 has been chosen, because it yields a better AR than 0.1. Thus, the adjustment of the proposal widths is very simple, as there is only one variable β, of which they depend on. This parameter could now undergo adaptive proposal, however, the optimum AR is unknown, which is why β has been adjusted manually such that the AR is between $25 - 50\,\%$.

3.1.3 Choosing A Proper Starting Point

Although the Markov Chain should have forgotten its starting point after the burn in phase, it is desirable to start it in an area where the posterior distribution is not very low. Otherwise, the posterior distribution would be nearly flat for a large amount of steps, which would lead to a large amount of necessary steps until the Markov Chain "finds" the interesting area, where the expectation value is. Hence, it is reasonable to start at a position, where we expect the expectation value of the posterior. As mentioned in the previous subsection, for a large number of steps N_j in the template sample one can assume that $a_{ij} \approx A_{ij}$. Therefore, it is a good choice to set the a_{ij} as the starting point of the A_{ij}.

However, in this analysis, it was impossible to apply the MA without performing a maximum likelihood fit before. The reason for this is that the logarithm of the likelihood can easily reach high values. In this case, 'high' means, that the computer has not enough storage to operate with the absolute likelihood, which is the exponential of this already high logarithm. Hence, a maximum likelihood fit is used to calculate the value of the logarithm of the likelihood in the maximum and subtract this value from the logarithm. This is equivalent to dividing the total likelihood by a constant factor and hence does not change the expectation value or standard deviation of the likelihood. Since the maximum likelihood method is used anyway, it is reasonable to use

the fitted position of the maximum of the likelihood as the starting position of the MA because this yields the best results.

For the p_j, the true values are used as starting points, which is possible, because they are self-invented and hence known. In general, for the electron separation problems one could use a maximum likelihood fit to get the starting points for the strength factors.

3.2 Results

3.2.1 Summary Of The MCMC Fit

Having built the posterior and adjusted the proposal widths and starting points one can start the MA. In Figure 3.3, the created pseudo measurement data and the templates multiplied with the true strength factors are shown. Comparing them to Figure 3.2 yields that the templates are an appropriate replica of the true distributions. In Figure 3.4, the pseudo measurement data and the results

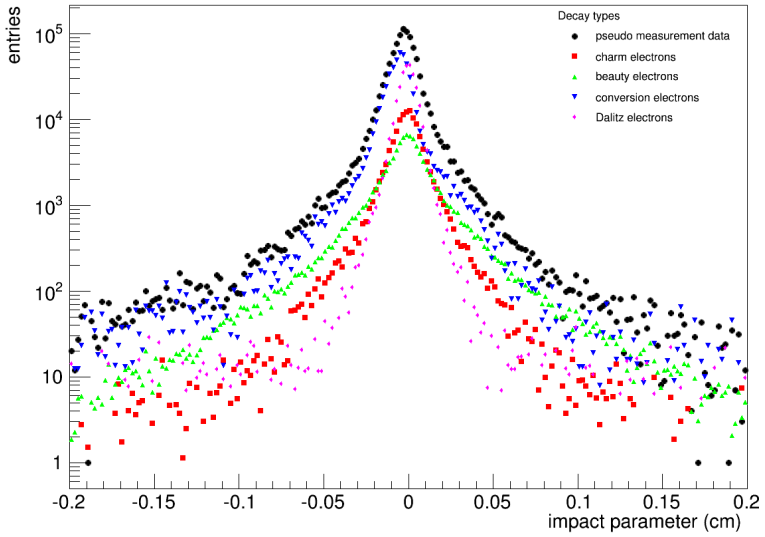


Figure 3.3: Templates for the MCMC fit.

of the MA as well as of the maximum likelihood fit are shown, whereby latter ones are fitted with the ROOT class TMinuit. The bin contents for the results

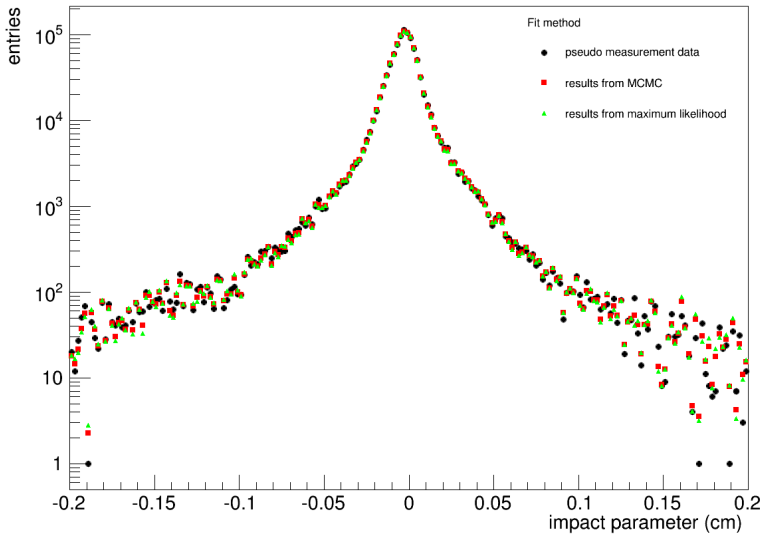


Figure 3.4: Results of the MCMC fit and maximum likelihood fit.

of the MA are calculated with the formula

$$f_i = \sum_{j=1}^{M} p_j A_{ij} \tag{3.3}$$

whereby p_j and A_{ij} are the values obtained from the MCMC fit. The histogram with the results is normalized by multiplying it with the factor $I_{data}/I_{results}$, whereby $I_{data} = N_D = 10^6$ is the number of entries in the histogram with the pseudo measurement data and $I_{results}$ is the number of entries in the histogram with the results. Due to visualization reasons, the uncertainties are not shown in here but will be investigated in the later sections of this work. Judging by eye, the results fit well to the pseudo measurement data, although there are some points with large relative deviation at the edge of the histogram. A summary of the fit is given in Table 3.1 and 3.2. While the fitted values for p_2 and p_3 go well together with the true values – for p_2 even within the 1σ range – for p_1 and p_4 there is a significant deviation of more than 3σ. One possible reason might be that the fit has not converged yet but needs more steps to do so. A larger number of steps would decrease the deviation of the true value

quantity	value
number of steps	10^7
calculation time	17700 s
AR	25.70%
β (factor for proposal widths)	0.01

Table 3.1: Results of MCMC fit for the electron separation problem.

strength factor	true value	MCMC fit	σ_x
p_1	0.1	0.12499	0.00547
p_2	0.1	0.09840	0.00243
p_3	0.5	0.49697	0.00198
p_4	0.3	0.27948	0.00317

Table 3.2: Strength factors p_j obtained from fit.

while the standard deviation of the sample instances would converge to the standard deviation of the posterior distribution. For the calculation time, it is important to mention that the calculation was run on the server 'alice-serv9' again.

In order to express the goodness of the fit quantitatively, the mean deviation of the fitted values f_i from the pseudo measurement data d_i shall be calculated. Since the absolute values of the f_i and hence their errors vary tremendously, not the absolute deviations but the relative ones are used. The mean relative deviation is calculated by

$$Dev_{rel}(x) = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{d_i - f_i}{d_i} \right| . \tag{3.4}$$

In this analysis one has $Dev_{rel}(x) \approx 4.75\%$, which is quite good. However, the fitted values have to be compatible to the pseudo measurement data within their errors. If they are, this would be a sign that the fit has converged correctly. Hence the errors are investigated in the next subsection.

3.2.2 Convergence Of The Fit

For the results of the MA algorithm in Figure 3.4, the errors are estimated by using the rules for Gaussian propagation of uncertainty and the standard deviations of the p_j and A_{ij} from the MCMC fit. One has

$$\begin{aligned}\Delta(f_i) &= \Delta\left(\sum_j p_j A_{ij}\right) \\ &= \sqrt{\sum_j (\Delta(p_j A_{ij}))^2} \\ &= \left[\sum_j \left(p_j A_{ij}\sqrt{\left(\frac{\Delta p_j}{p_j}\right)^2 + \left(\frac{\Delta A_{ij}}{A_{ij}}\right)^2}\right)^2\right]^{1/2}\end{aligned} \tag{3.5}$$

where $\Delta p_j = \sigma_{x,i}$ and $\Delta A_{ij} = \sigma_{x,ij}$ are the standard deviations of the sample instances for the respective fit parameters p_j or A_{ij}. Since the values of the f_i have been rescaled, the errors also need to be multiplied with the factor $I_{data}/I_{results}$. To get a summary of the errors, the mean of the relative errors is calculated by

$$Err_{rel}(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{\Delta f_i}{f_i}. \tag{3.6}$$

Its value is $Err_{rel}(x) \approx 1.30\%$, which is not bad. Yet, one has $Dev_{rel}(x) > 3{\cdot}Err_{rel}(x)$, which means that on average the fitted values deviate significantly from the pseudo measurement data. Indeed, this does not mean that most of the fitted values deviate significantly, yet the result indicates that the fit may not have converged. In the calculation of the mean relative error the statistical fluctuations of the pseudo measurement data Δf_i have not been considered, which increases $Err_{rel}(x)$.

In Figure 3.5 the absolute value $|f_i - d_i|$ of the deviation of the fitted values for the f_i from the pseudo measurement data divided by the total errors of the deviation $\Delta(|f_i - d_i|) = \sqrt{(\Delta f_i)^2 + (\Delta d_i)^2} = \sqrt{(\Delta f_i)^2 + d_i}$ is shown, where $\Delta d_i = \sqrt{d_i}$ represents the Poissonian fluctuations of the pseudo measurement data. Some of the points have a larger deviation than 3σ but most of them do

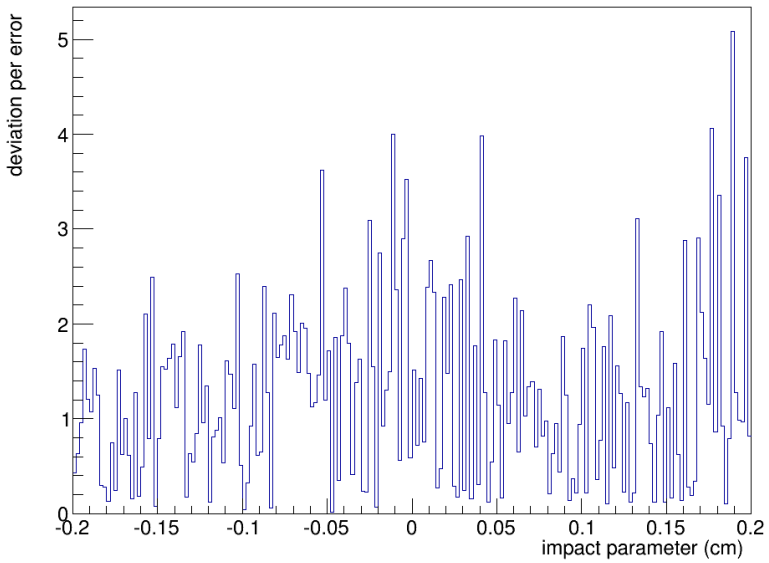


Figure 3.5: Deviation of the fitted values from the pseudo measurement data in units of Δf_i.

not deviate significantly.

3.2.3 Autocorrelation

In order to investigate how many steps one needs to get independent sample instances, the autocorrelation function $Auto(lag)$ according to Formula 1.13 is calculated for some values of dimensions and lag, i.e. the number of steps two sample instances are apart. It is shown in Figure 3.6. It is important to mention that due to computational reasons the autocorrelation function is calculated in another run than the fit in the previous subsection. Hereby, the number of steps is decreased to $N = 700000$. Even for $lag = 10000$ the autocorrelation is still at $Auto(10000) \approx 0.8$ for almost every dimension and hence quite high. For $lag = 10^5$ it is spread from ≈ -0.4 to ≈ 0.45 for the different dimensions and hence still not close to 0. Therefore, it is not surprising that the fit has not converged yet. The sample instances are simply not independent enough. There are no tendencies of the correlation behavior between the individual dimensions. Hence, the proposal widths have an approximately

equal scale for each dimension. If for example the proposal width was too large for one dimension but too small for another one, there would be a difference in the correlation coefficients. However, the autocorrelation function for the p_j is not shown in this plot. Therefore there is a good case to believe that the autocorrelation functions of the p_j look different than those for the A_{ij}.

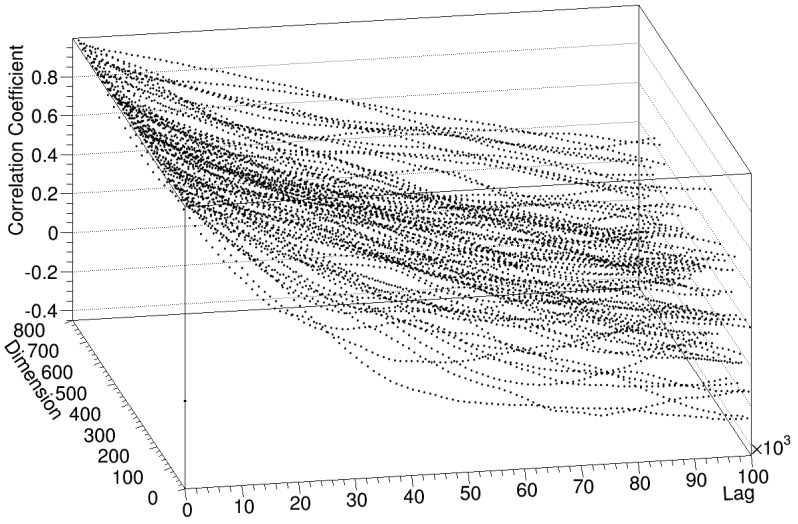


Figure 3.6: Autocorrelation function for MCMC sample.

For a one-dimensional posterior distribution one can assume an optimum for the proposal width for $\beta \approx 1$ (see Formula 3.2), because the factor multiplied to β to get the proposal width is of the order of the width of the posterior distribution. This would yield a quite independent sample. In this analysis one has $\beta = 0.01$, which means that the ratio of the standard deviation of the posterior distribution and the proposal width is $\sigma_{posterior}/\sigma_{prop} \approx 100$. Since the Markov chain behaves like a random walk, one can expect that after k_{acc} accepted steps it has 'travelled' the distance $l = \sigma_{prop}\sqrt{k_{acc}}$. Therefore, in order to have travelled the whole posterior distribution, $k_{acc} = (\sigma_{posterior}/\sigma_{prop})^2 \approx$ 10000 accepted steps are necessary [10]. Due to an AR of $\approx 25.70\,\%$ the number of accepted steps k_{acc} has to be divided by the AR to get the total number of steps necessary until the algorithm has travelled the whole posterior distribution once. In this case, one needs approximately 40000 steps for this.

With a total number of steps of 10^7, one can estimate, that the algorithm has travelled the posterior distribution for about $10^7/39000 = 250$ times [10]. This value is probably a little overestimated, since in Figure 3.6 one can see that at $lag = 40000$ there is still some autocorrelation. Therefore it is quite debatable whether this fit has converged or not.

3.3 Fit With Low Binning

3.3.1 Summary And Results

Since the fit with 200 bins has yielded an extremely high autocorrelation and some of the p_j deviated significantly from the true values, another MCMC fit is done with $N = 20$ bins. This aims at the reduction of the dimension such that the proposal widths can be increased and a lower autocorrelation is obtained. For the fit the same statistics $N_D = N_j = 10^6$ as before are used. However, the proposal widths for the p_j have been changed to

$$\sigma_{prop,j} = \beta \cdot 0.017 \tag{3.7}$$

such that they are smaller than in Formula 3.2. The factor 0.017 has been chosen by comparing the autocorrelation functions of the p_j to the ones of the A_{ij}. Since one wants the proposal widths of the several dimensions to have approximately the same scale respective to the posterior distributions, it is reasonable to choose the proposal widths such that the autocorrelation functions are similar for each dimension. This is the case for the proposal width in Formula 3.7. The results are presented in the Figures 3.7 to 3.10, which are analogue to the Figures 3.3 to 3.6, and in the Tables 3.3 and 3.4, which are analogue to the Tables 3.1 and 3.2. The templates multiplied with the true strength factors in Figure 3.7 fit well to the true distributions in Figure 3.2. The sampled pseudo measurement data represent the true distribution appropriately, too.

Again judging only by eye, in Figure 3.8, the results fit well to the pseudo measurement data. There are some fluctuations for the impact parameter

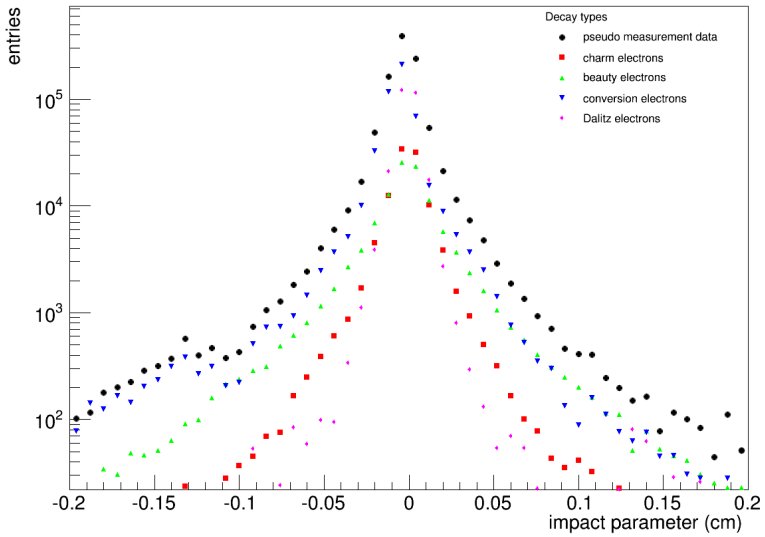


Figure 3.7: Templates for the MCMC fit with low binning.

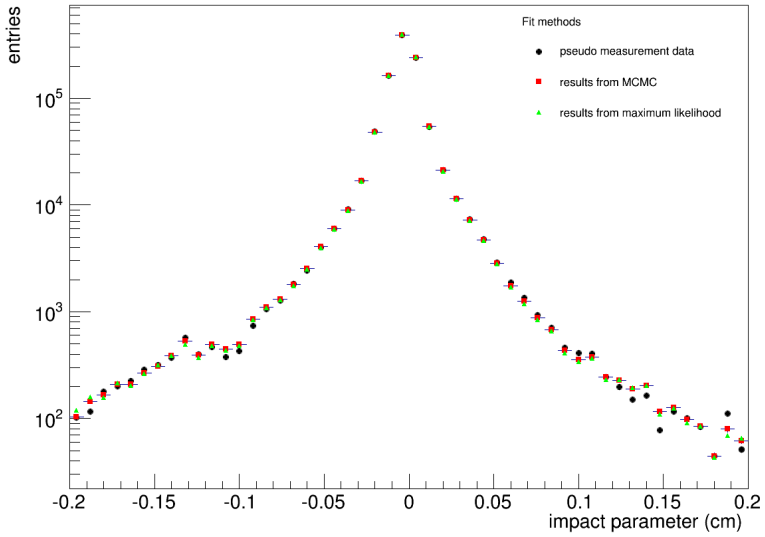


Figure 3.8: Results of the MCMC fit and maximum likelihood fit for low binning.

quantity	value
number of steps	10^7
calculation time	2700 s
AR	14.66 %
β (factor for proposal widths)	0.08

Table 3.3: Results of MCMC fit of electron separation problem with low binning.

strength factor	true value	MCMC fit	σ_x
p_1	0.1	0.10477	0.00640
p_2	0.1	0.11063	0.00283
p_3	0.5	0.49709	0.00213
p_4	0.3	0.28749	0.00373

Table 3.4: Fitted values for strength factors p_j for low binning.

values > 0.1 cm and around -0.1 cm, yet in Figure 3.9 one can see that the deviations of the fitted results from the pseudo measurement data are within $3\,\sigma$ for most bins. Only at ≈ -0.1 cm and at ≈ 0.15 cm, where one can see fluctuations in Figure 3.8, the deviations exceed $3\,\sigma$. However, the fluctuations result from the pseudo measurement data and since the Metropolis algorithm shows these fluctuations, too, the fit can still be considered to be good.

The fitted values for the p_j in Table 3.4 are close to the true values. The results for p_1 and p_3 are compatible with the true values within $3\,\sigma$, while the p_2 and p_4 show a larger deviation. A possible reason might be that the templates are also sampled with the Metropolis algorithm and might not have converged completely. This would shift the expectation values for the p_j.

Again the mean of the relative deviation is calculated as $Dev_{rel}(x) \approx 0.0184$ according to Formula 3.4. On the contrary to Formula 3.6, the mean of the relative errors is calculated considering the statistical fluctuations Δd_i of the pseudo measurement data d_i:

$$Err_{rel}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{\sqrt{(\Delta f_i)^2 + (\Delta d_i)^2}}{f_i}. \tag{3.8}$$

Thus the mean of the relative errors $Err_{rel}(x) = 0.0138$ is larger than without

considering them and one has $Dev_{rel}(x) < 3 \cdot Err_{rel}(x)$. This indicates that the fit has converged.

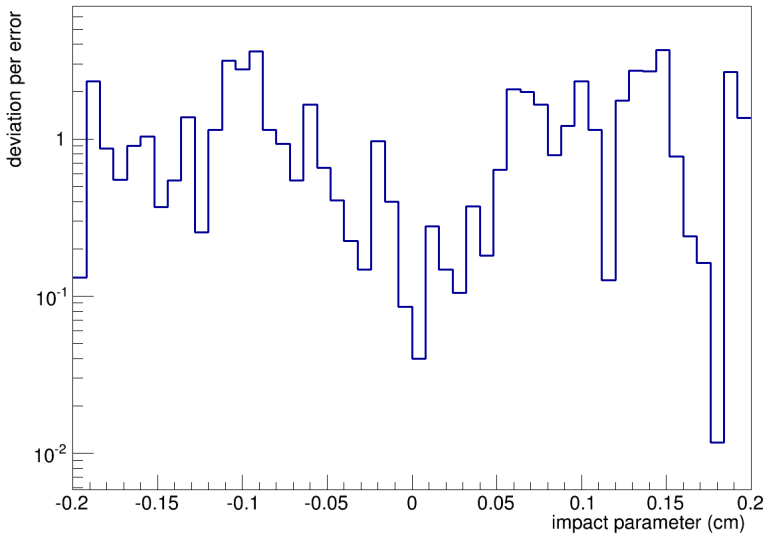


Figure 3.9: Deviation of the fitted values from the pseudo measurement data in units of Δf_i for low binning.

3.3.2 Autocorrelation

To test the convergence of the fit more precisely, again the autocorrelation function is shown in Figure 3.10. It is very important to mention that due to computational reasons the following plots were created in another run than the results before, whereby the number of steps is now only 700000. One can see that the absolute value of the correlation coefficient is below 0.2 for $lag > 5000$. Hence the autocorrelation is much smaller than for the fit with a binning of 200. However, it is important to mention that this is the result of the lower dimension as well as of the smaller proposal widths $\sigma_{prop,j}$. With smaller $\sigma_{prop,j}$ the AR increases and hence one can increase β, which results in a smaller autocorrelation in general. Again one can estimate the number of steps until the Markov chain has travelled the whole posterior distribution once. With $\beta = 0.08$ one has $\sigma_{posterior}/\sigma_{prop} = 1/\beta = 12.5$ The necessary

number of accepted steps is $k_{acc} = (\sigma_{posterior}/\sigma_{prop})^2 = 156.25$. Divided by the AR one has a necessary number of total steps of $k = 156.25/14.66\,\% \approx 1065$. In this estimation, the smaller proposal widths of the p_j are not considered, which is why the real total number should be even larger. However, this estimation yields a smaller number than the autocorrelation function. According to the latter one, independent sample instances should occur after 3000 - 5000 steps. With a total number of 10^7 steps in this fit one can expect that the Markov chain should have travelled the posterior distribution $\approx 10^7/5000 = 2000$ times, which should be enough for the fit to converge.

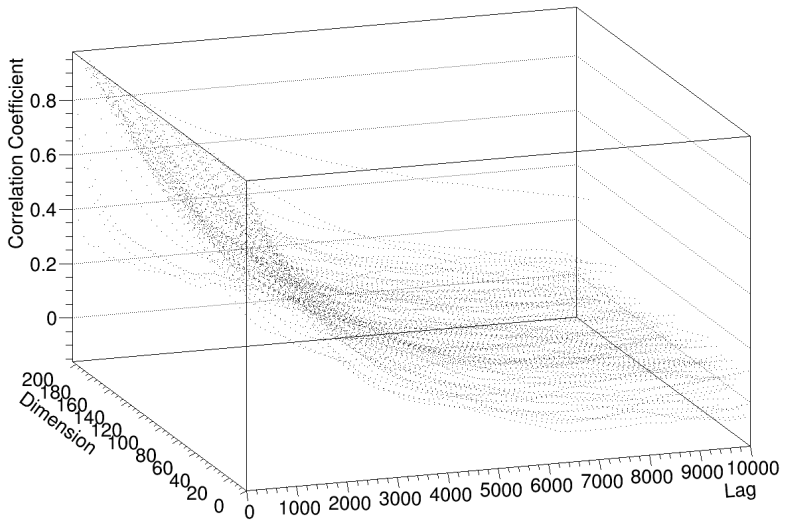


Figure 3.10: Autocorrelation function for MCMC sample with low binning.

3.3.3 Marginals Of The Strength Factors

At last the posterior distributions of the p_j shall be shown by projecting the sample instances on the dimensions of the p_j. In other words, the number of sample instances with a certain value for p_j is plotted in a histogram. This is called *marginalization* and is shown in Figure 3.11. One can see that the widths of the posterior distributions fit the values for the standard deviations σ_x in Table 3.4. However, the mean values are shifted so that they do not

coincide with the true values. Since the posterior distributions do not show any larger fluctuations but are quite smooth and well-shaped, the problem does not seem to be the MCMC fit but either the pseudo measurement data or the templates. As mentioned before the templates are created using the Metropolis algorithm and may have not converged yet, which could lead to this shift.

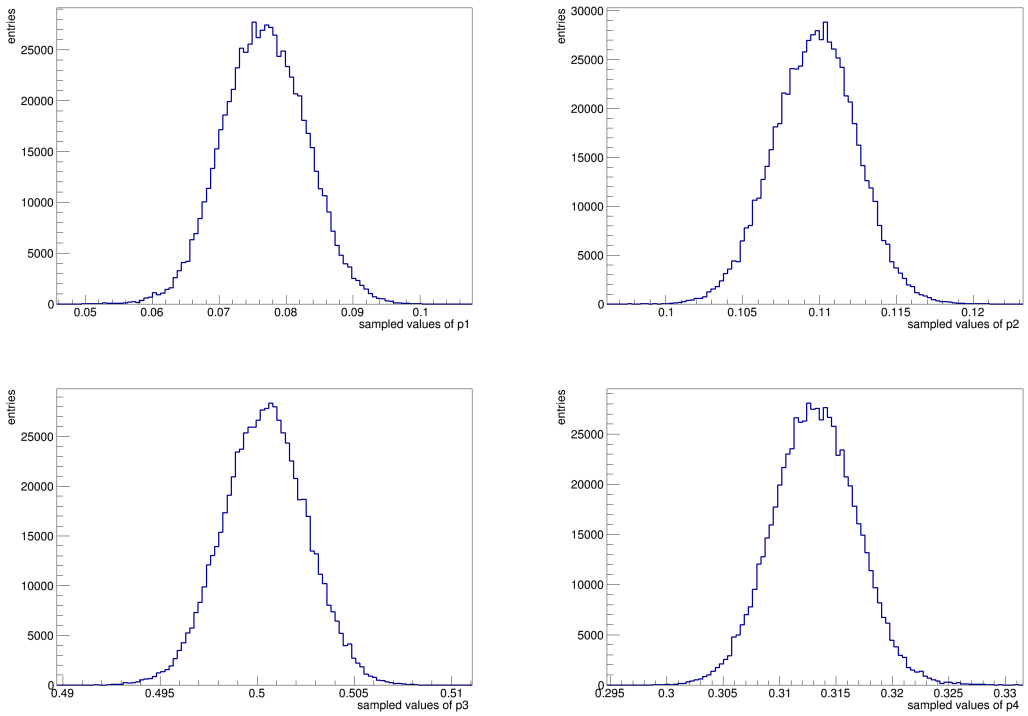


Figure 3.11: One-dimensional marginals of the p_j.

In the Figure 3.12 the marginals of two different strength factors are shown. These are the projections of the sample instances on two different p_j or in other words the number of sample instances with a certain value for p_j and p_k with $j \neq k$. First, one can see the well-formed extensions of the posterior distributions, whereby the extension of p_1 is the largest, as in Table 3.4 p_1 has the largest standard deviation σ_x. Furthermore, one can see the correlations of the p_j for the different sample instances to each other. The values of p_1 (charm electrons) and p_2 (beauty electrons), just as the values of p_1 and p_4

(Dalitz electrons), are anti-correlated with each other. The p_2 and p_4 values are positively correlated with each other. The values of p_3 (conversion electrons) are rather uncorrelated to the other strength factors, although a slight anti-correlation to p_1 is recognizable.

If one had only two strength factors, one would expect them to be anti-correlated, because they have to sum up to unity and hence the increase of one would lead to a decrease of the other. With four strength factors however, the correlation depends strongly on the particular shape of the distributions of the individual electron sources. Considering Figure 3.2, one can see that the distribution of the conversion electrons is not completely symmetric but a little shifted to the left, while the other distributions are quite symmetric. This makes the distribution of the conversion electrons very distinguishable from the other distributions such that the optimum value for p_3 is rather independent on the other ones. This is why the values of p_3 are not much correlated with the other p_j and why the fitted values for p_3 have always been good. The distributions of the other p_j are all quite symmetric and differ basically in their width. While beauty electrons show a rather large width – due to the long lifetime of the beauty quark – the Dalitz electrons are distributed sharply. The charm electrons are in between these two. Hence, the distribution of the charm electrons can partly be compensated in the fit by a superposition of the beauty and Dalitz electrons. This explains the correlation behavior. A low p_1 can be compensated by high p_2 and p_4 and vice versa. Hence, p_1 is anti correlated to p_2 and p_4. The latter ones however, have to be positively correlated, because they have to conserve the shape of the total distribution, which is neither too sharp nor too wide. Hence, if the broadly distributed beauty electrons increase, the sharply distributed Dalitz electrons must do so, too.

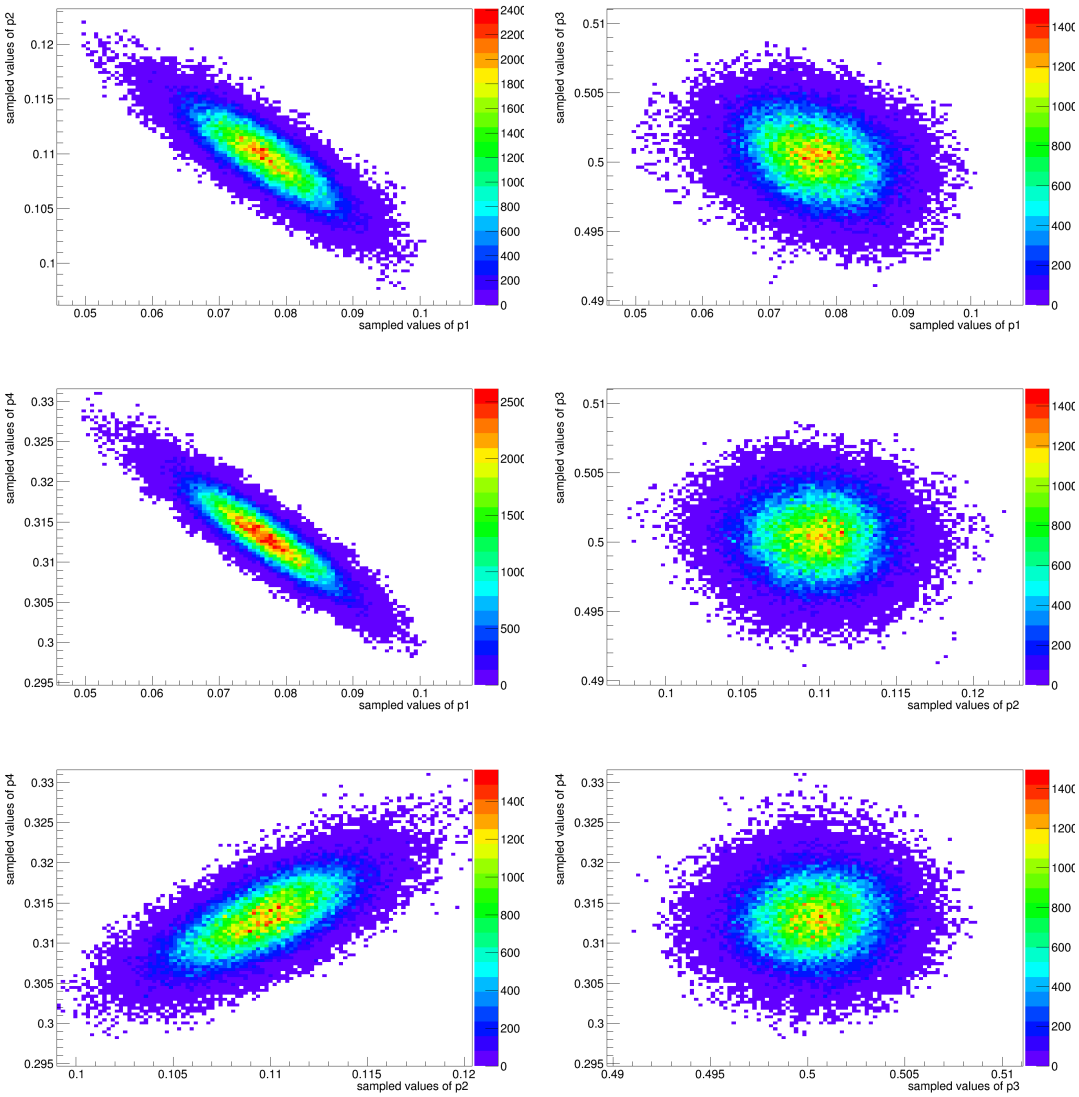


Figure 3.12: Two-dimensional marginals of the various strength factors p_j.

4 Summary, Discussion And Outlook

In this thesis, a Markov Chain Monte Carlo sampling algorithm, namely the Metropolis algorithm, was successfully implemented and applied to a posterior distribution which describes the probability distribution of the strength factors of different electron sources. These strength factors are the coefficients of impact parameter model distributions of electrons for different decay types, which build a superposition that corresponds to the impact parameter total distribution. The aim was to fit the strength factors and analyse the applicability of this method with respect to convergence of the fit in reasonable calculation time, proper error estimation of the fitted values, and profitability of further research on using MCMC methods for the separation of heavy flavor electrons.

Considering the adjustment of the fit, it would be interesting to test if the burn in is big enough by adopting the proposal of [13] and systematically investigating whether the fit results depend on the starting location. However, since the estimated number on which the algorithm has travelled the whole posterior distribution is estimated to be large, namely ≈ 250 for 200 bins and ≈ 2000 for 50 bins, one can assume that the results are independent on the starting point.

The main task in the MCMC fit is definitely finding proper proposal widths and distributions. Since the choice done in the fit with binning 50 works, one can consider it as appropriate. Yet, in the autocorrelation plot in Figure 3.10 one can see, that the correlation functions for the p_j do not completely concur, because they all have the same proposal width. A different one for each p_j could yield an even lower autocorrelation with the same AR and hence improve the

efficiency. Additionally, it would be interesting to have theoretically calculated optimal proposal widths. Alternatively, one could try to implement another kind of adaptive proposal, which repeatedly adjusts the proposal widths in the different dimensions looking for better efficiency.

One remaining issue are the shifts of the expectation values of the p_j from the true values. From the plots of the marginals one knows, that these shifts are not caused by the MCMC sampling of the posterior distribution but either by the pseudo measurement data or the templates. The algorithm seems to converge to the wrong expectation value and therefore increasing the number of steps would not fix this problem. A possible reason is, that the templates are sampled with the MA, too, and therefore might not have converged because the sample instances are autocorrelated. Although the AR of the template sampling is forced to $\approx 30\,\%$ with adaptive proposal in the fit with binning 50, the optimum AR for this sampling function is unknown. Hence, it may be a better choice to use acceptance rejection sampling for the templates. Considering Table 3.2, one can see that p_1 is too high, while p_2 and especially p_4 are too low. This could be due to the anti-correlation of p_1 with p_2 and p_4 based on the similarity of these distributions. A stronger charm electron distribution is compensated by weaker beauty and Dalitz electron distributions and hence the expectation value is shifted. To remedy this problem, in general a high granularity, i.e. a large number of bins, is important such that also small structures of the source distributions can be used to distinguish them. Additionally the templates should have good statistics, which was maybe not given here, because they were created with the MA. However, templates with high statistics sharpen the posterior distributions and hence lead to a lower AR or higher autocorrelation in case the proposal widths are made smaller.

The fit applied in this analysis is not applicable for a serious source separation problem yet, because it is not precise enough. Additionally the fitted values for the p_j have significant deviations from the true values for this settings, though this is probably due to the shifts of the expectation values. While the sample instances in the fit with binning 200 have been highly autocorrelated and hence the fit should be considered sceptically, one can consider the algorithm to have converged for binning 50, at least after the shifting problem

is fixed. Therefore MCMC methods have shown their potential to be applicable for the electron separation.

For further investigations on this topic, it is advisable to decrease the autocorrelation for higher binning. First of all, one can align the autocorrelation functions of the p_j to those of the A_{ij}, which was partly done for binning 50 in here. Then, one might consider changing the prior such that it regards the fact that the electron distributions have a maximum. This would be accomplished by pitching an area, from which one knows for sure that the maximum of the distribution lies within there. Then, it is checked whether there are two A_{ij} for which one is farther away from the pitched area but still larger than the other one. Every sample instance for which this is true is rejected. At last one might consider to change the sampling algorithm. One modification of the Metropolis algorithm is the so called 'Gibbs sampling'. It is basically a slow random walk sampler as well. However, while in the MA a new sample instance is considered for each dimension at once, in the Gibbs sampling a step is always made in only one dimension. The acceptance is dependent on the conditional distribution of the sampling function for this dimension. Thus, steps for the several dimensions are made one after another. This has the advantage that the proposals for the individual dimensions do not need to be adjusted relatively to each other and maybe it is possible to implement some kind of high dimensional adaptive proposal with the Gibbs sampling [10].

In conclusion, there are still many possibilities to improve the Markov Chain Monte Carlo method used in this work and since the results have already been satisfying, further investigation in this topic is highly recommendable.

Bibliography

[1] A. Gelman, G.O. Roberts, W.R. Gilks. Efficient metropolis jumping rules. *Bayesian Statistics*, 5:599–608, 1996.

[2] The ALICE Collaboration. The ALICE Experiment. URL http://aliceinfo.cern.ch/Public/en/Chapter2/Chap2Experiment-en.html, 2008.

[3] F. Carminati, P. Foka, P. Giubellino, A. Morsch, G. Paic, JP. Revol, K. Safarik, Y. Schutz, UA. Wiedemann et alias. ALICE: Physics performance report, volume I. *Journal of Physics G: Nuclear and Particle Physics*, 30(11):1517, 2004.

[4] I. Narsky, F.C. Porter. *Statistical Analysis Techniques in Particle Physics*. WILEY-VCH, 2014.

[5] J. Alme et alias. The ALICE TPC, a large 3-dimensional tracking device with fast readout for ultra-high multiplicity events. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 622(1):316–367, 2010.

[6] J.E. Turner, D.J. Downing, J.S. Bogard. *Statistical Methods in Radiation Physics*. WILEY-VCH, 2012.

[7] K. Nakamura, Particle Data Group et alias. Review of Particle Physics. *Journal of Physics G: Nuclear and Particle Physics*, 37(7A):075021, 2010.

[8] J. Klein. *Commissioning of and Preparations for Physics with the Transition Radiation Detector in A Large Ion Collider Experiment at CERN*. PhD thesis, Diploma thesis, Universität Heidelberg, Physikalisches Institut, Heidelberg, 2008.

[9] M. Völkl, MinJung Kweon, Yvonne Pachmayer. Measurement of Electrons from Beauty Hadron Decays in Pb-Pb at $\sqrt{s_{NN}} = 2.76$ TeV. 2014. ALICE internal analysis note.

[10] D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2005.

[11] Martin Völkl. Private communication.

[12] P. Braun-Munzinger, J. Stachel. The quest for the quark–gluon plasma. *Nature*, 448(7151):302–309, 2007.

[13] R. Bardenet. *Chapter "Monte Carlo methods" in Proceedings of the 2012 IN2P3 School of Statistics*, chapter Monte Carlo methods. EDP Sciences, 2013.

[14] R. Barlow, C. Beeston. Fitting using finite Monte Carlo samples. *Computer Physics Communications*, 77(2):219–228, 1993.

[15] K.O. Schweda. Prompt production of D mesons with ALICE at the LHC. 2014.

[16] The ALICE Collaboration. Technical Design Report of the Inner Tracking System (ITS). Technical report, CERN, 1999.

[17] The ALICE Collaboration. Technical Design Report of the Time of Flight System (TOF). Technical report, CERN, 2000.

[18] The ALICE Collaboration. Technical Design Report of the Time Projection Chamber. Technical report, CERN, 2000.

[19] The ALICE Collaboration. Technical Design Report of the Transition Radiation Detector. Technical report, CERN, 2001.

[20] The ALICE Collaboration. Measurement of electrons from beauty hadron decays in pp collisions at $\sqrt{s} = 7$ TeV. Technical report, CERN, 2013.

[21] M. Völkl. Study of the Transverse Momentum Spectra of Semielectronic Heavy Flavor Decays in pp Collisions at $\sqrt{s} = 7$ TeV and Pb-Pb Collisions at $\sqrt{s_{NN}} = 2.76$ TeV with ALICE, 2012. Master thesis.

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 10.11.2014,

.....