Development of the Monte Carlo event generator tuning software package `Lagrange` and its application to tune the PYTHIA model to the LHCb data

Dissertation

zur Erlangung des Doktorgrades
des Fachbereichs Physik der Universität Dortmund
ausgeführt am
Max-Planck-Institut für Kernphysik, Heidelberg

vorgelegt von
Dmitry Popov
aus Nowosibirsk, Russland

Heidelberg
September 2013

To my dear parents.

"You are never given a dream without also
being given the power to make it true.
You may have to work for it, however."

Richard Bach

Abstract

Lectori benevolo salutem.

Lat.: To the kind reader, greeting.

One of the general problems of modern high energy physics is a problem of comparing experimental data, measurements of observables in high energy collisions, to theory, which is represented by Monte Carlo simulations.

This work is dedicated to further development of the tuning methodology and implementation of software tools for tuning of the PYTHIA Monte Carlo event generator for the LHCb experiment. The aim of this thesis is to create a fast analytical model of the Monte Carlo event generator and then fitting the model to the experimental data, recorded by the LHCb detector, considering statistical and computational uncertainties and estimating the best values for the tuned parameters, by simultaneous tuning of a group of phenomenological parameters in many-dimensional parameter-space. The fitting algorithm is interfaced to the LHCb software framework, which models the response of the LHCb detector. Typically, the tunings are done to the measurements which are corrected for detector effects. These corrections depend on the model which is input into the detector simulation, and the proposed approach allows to avoid this dependence.

This tuning methodology and the developed software tools were tested on a small subset of kinematical parameters, which describe the gross fraction of the hadronic final states in high energy *pp*-collisions.

Zusammenfassung

Eines der allgemeinen Probleme der modernen Hochenergiephysik besteht darin experimentelle Daten, also Messungen von Observablen in Hochenergie-Kollisionen, mit Ergebnissen der Theorie zu vergleichen, die aus Monte Carlo Simulationen hervorgehen.

Die vorliegende Arbeit widmet sich der Weiterentwicklung der Tuning-Methodik und dem Implementieren von Softwarewerkzeugen zum Tunen des PYTHIA Monte Carlo Event Generators für das LHCb-Experiment. Das Ziel ist ein schnelles analytisches Modell des Monte Carlo Event Generators zu erstellen und dann dieses Modell an die vom LHCb Detektor aufgenommenen Daten anzupassen, dabei statistische wie numerische Fehler zu berücksichtigen, und die besten Schätzwerte für die zu tunenden Parameter durch ein simultanes Tuning einer Gruppe von phänomenologischen Parameter im vieldimensionalen Parameterraum zu erhalten. Der Anpassungsalgorithmus hat eine Schnittstelle zum LHCb-Software-Framework, welches das Ansprechen des Detektors modelliert. Üblicherweise werden für die Tunings Messungen verwendet, die auf Detektoreffekte korrigiert sind. Diese Korrekturen sind aber abhängig von den in den Detektorsimulationen verwendeten Modellen, was der vorgeschlagene Ansatz umgeht.

Diese Tuningmethode und die entwickelten Softwarewerkzeuge wurden mit einer kleinen Anzahl von kinematischen Parametern getestet, welche den größten Teil der hadronischen Endzustände von Hochenergie-Kollisionen beschreibt.

Contents

List of Figures

List of Tables

Acknowledgments

Summa cum pietate.

Lat.: With the greatest respect.

I am grateful to professor Michael Schmelling and professor Werner Hofmann for the great opportunity to work on my Ph.D. in one of the most famous German institutes for physics, in Max Planck Institute for Nuclear Physics (MPIK), which is situated in the romantic and picturesque cityscape of Heidelberg.

In particular, I am indebted to professor Schmelling for his constant interest in my research, for his useful comments, discussions and help in my studies, for our traditional Friday group meetings with debates on the work in progress, during which I have learned a lot. It is not an exaggeration to say that this research would have been impossible without his support and mentoring. Even after these few years of working together with Michael, I constantly find myself surprised by his amazing intelligence, inexhaustible energy, and genuine friendliness.

Through my graduate studies in the Max Planck Institute I had an opportunity to participate in one of the most ambitious experiments in physics of the XXI century and even attend the memorable historic press conference on the 4th of July, 2012, during which ATLAS and CMS have presented their first results on the search of the Higgs boson, and to see Peter Higgs wiping a tear from his eye as the findings were announced. I hope that my work has made a useful contribution to the experimental data processing at the LHCb and that the LHC experiments will yet delight the world with new and interesting results and discoveries.

I am very grateful to professor Tasso Knoepfle and his lovely wife Anna for their help in setting up my life when moving to Heidelberg, for the kind attention to me, their willingness to help, their sweet Christmas presents and for all the holidays spent in the warm circle of their family, which all made me feel at home in Germany and a little less homesick.

I have had helpful discussions and received comments and suggestions from many colleagues, including: Markward Britsch, Johan Blouw, Osvaldo Aquines, Florin Maciuc, Mikhail Zavertiaev, Marianna Fontana, Dmytro Volyanskyy, Nikolay Gagunashvili, Thomas Kihm and Alexander Zaytsev.

This work was supported by the Max Planck Society (Max-Planck-Gesellschaft).

Chapter 1

Introduction

Usus efficacissimus rerum omnium magister...
Lat.: Custom is the best master of all things...

– Pliny, Naturalis Historiae

1.1 CERN

The rapid development of fundamental science in the 20th century made it global and international: today many countries around the world join forces and resources to solve scientific problems which form the base of the scientific and technical progress. Indeed, current studies require high material costs which are hard to upkeep even by the most economically developed countries. The complexity of these scientific problems requires a focus of intellectual potential of many scientists, which was made possible by the rapid development of the information environment at the end of the last century. Foremost, all of this refers to the world of high-energy physics, which nowadays requires enormous technological effort. And one of such modern focal points of the state of the art technology and intelligence – is CERN.

CERN (European Organization for Nuclear Research) – is the world's largest center for physics research and a unique experimental ground in the field of elementary particle physics. It is located west of Geneva, on the Franco–Swiss border and at the foot of the Jura mountain range, where the geologic and seismic conditions are suitable for a large particle accelerator construction.

The proposal to establish CERN was first sounded by the French physicist, Nobel laureate Louis de Broglie in 1949 at the European Conference on Culture in Lausanne. He proposed the creation of an international organization for fundamental research of such a scale, which wouldn't be feasible by a single national institution alone. He was supported by European governments and in 1954 the European Centre for Nuclear Research was born. The initial number of 12 countries which signed the Convention on CERN's membership, today has grown up to 20 member states. About 7000 scientists from approximately 500 research centers and universities in 80 different countries from around the world use CERN as the experimental grounds.

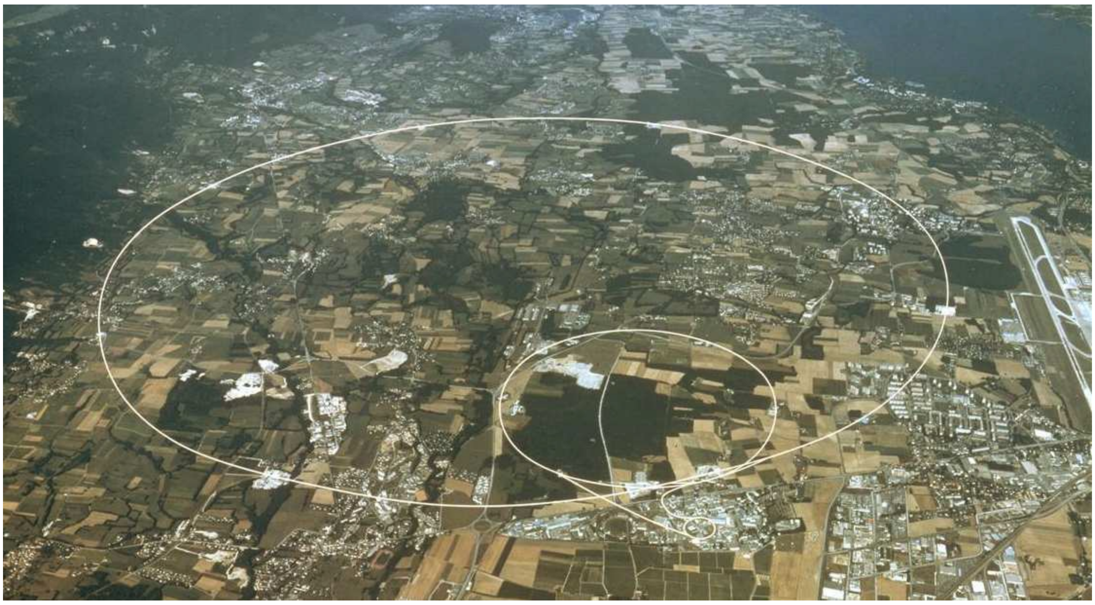


Figure 1.1: Aerial view of CERN [1].

Since it's foundation, CERN hosted many experiments and among the most famous achievements are the following ones:

1968: The invention of the multi-wire proportional and drift chambers.

1973: The experimental observation of the weak neutral currents at the Gargamelle bubble chamber, shortly after their theoretical prediction.

1983: The discovery of the weak bosons W and Z, which was an important milestone for the confirmation of the Standard Model.

1989: The determination of the number of light neutrino families at LEP.

1990: Birth of the World Wide Web.

1995: The first synthesis of atoms of antimatter (antihydrogen), which were produced in collisions of antiprotons and xenon atoms.

1999: The discovery of the phenomenon of direct CP violation (NA48 experiment).

2012: Discovery at LHC of a boson with mass around 125 GeV, which is consistent with long-sought Higgs boson.

1.1.1 The LHC

LHC, the Large Hadron Collider, is the particle accelerator with which physicists are able to penetrate as deeply into the matter as never before. Its construction was proposed in 1984 and it was planned that it would operate at a significantly increased, in respect to all previous accelerators, center-of-mass energy of $\sqrt{s} = 14$ TeV.

The name “Large” comes from the the fact that it is the largest and most complicated machine ever built by humanity. In order to save on building costs, the collider was installed in the 27 km underground tunnel at a depth of about 100 meters, which hosted the LEP accelerator before.

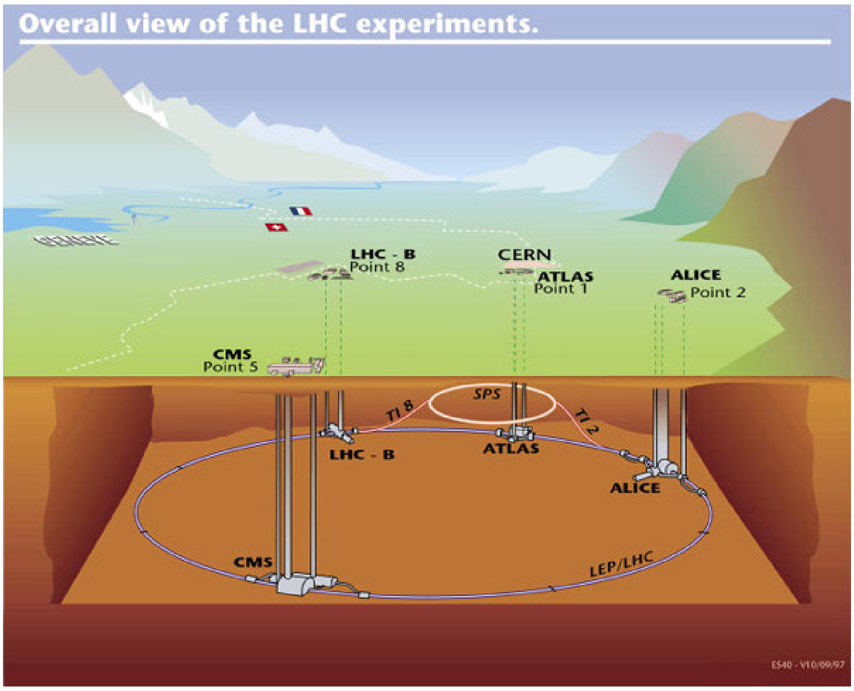


Figure 1.2: Schematic view of the LHC underground structure [1].

The “Hadron” part reflects the fact that the colliding particles are hadrons, i.e. protons and heavy nuclei. Unlike it’s predecessor, the LEP project, which accelerated electrons and positrons, LHC chose to work with proton beams. The reason for that was the aim to reach the maximum possible energy in a given accelerator ring, and the motion of charged particles in a circular orbit is accompanied by energy losses, due to the synchrotron radiation. One can estimate how much higher these losses are during acceleration of electrons in the same ring taking into account that the electron mass is almost 2000 times smaller than the proton’s mass, knowing that the intensity of such radiation is inversely proportional to the fourth power of the mass of the accelerated particle.

Finally, the “Collider” part of the name points out that it has two beams of particles circulating in opposite directions, which then intersect in special points for particles collisions. In comparison to fixed target experiments, the benefit of such a construction is the higher energy in the center of mass of the colliding particles.

The purpose of the experiments conducted at the LHC is to shed light on fundamental questions, such as, “What does matter consists of and where does it come from?”, “Why is the observable universe dominated by matter?”, “What are the forces acting between elementary particles and combine them into complex objects that surround us?”, “What is the origin of our universe and what are the laws that determine its development?”. Or if to be a bit more specific, to prove, or disprove, the existence of the Higgs boson, super-symmetric particles and search for any manifestation of new physics.

The LHC is indeed a very complex machine. Inside it, the two beams of protons circulate with opposite directions and their speed is close to the speed of light in vacuum. To keep the beam of these energetic particles on a circular track, the accelerator requires magnetic fields which are stronger than at any other accelerator. The technology for generating such magnetic fields, by means of superconducting magnets, has been used before, but there was never an attempt to construct a superconducting apparatus of such a great scale. There are more than 1200 superconducting dipole magnets which are placed along the ring and deflect the particle beams. The maximum value of the magnetic field created by these magnets reaches up to around 8 Tesla. There is yet another category of superconducting magnets: about 400 quadrupole magnets help to optimally focus the two beams along the accelerator ring and in the four collision points where the main experiments take place. All of these superconducting magnets operate at a temperature as low as 1.9 K and to cool them down to such a low temperature, nearly 96 tons of superfluid helium circle through the accelerator's ring.

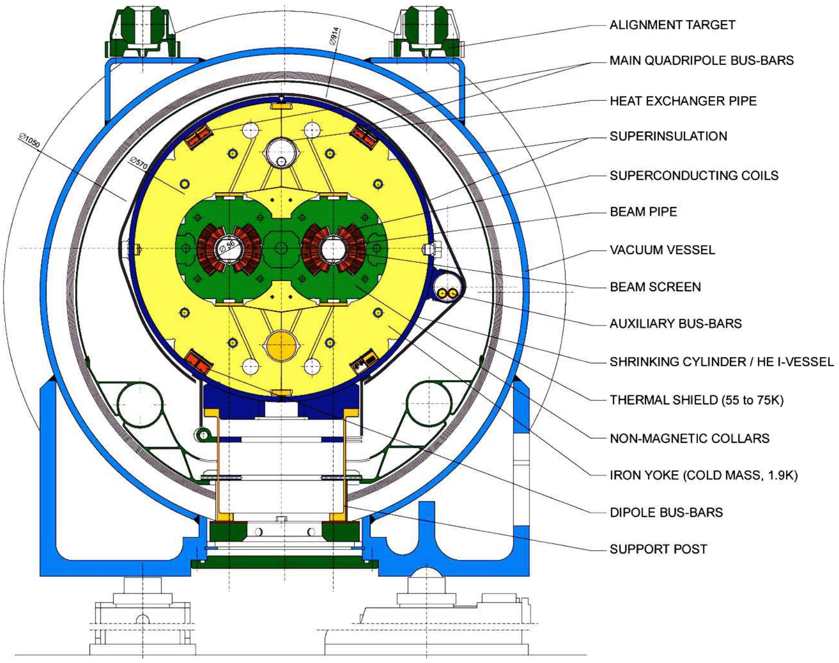


Figure 1.3: Diagram showing the cross-section of an LHC dipole magnet [1].

Since the LHC accelerates two beams of particles which move in opposite directions, in reality it consists of two accelerators in one, and before the beams are injected into the main ring, they are accelerated in several sequentially connected machines. Initially, protons reach the energy of 50 MeV in the linear accelerator (LINAC), from which they are rerouted into the Proton Synchrotron Booster (PBS), where they acquire an energy of 1 GeV. Afterwards they travel to the Proton Synchrotron (PS) and Super Proton Synchrotron (SPS) in which they are boosted to energies of 26 GeV and 450 GeV respectively. And only after all these intermediate steps the beams of protons reach the main ring of the LHC, where they are accelerated to the much higher energy of 4 TeV per beam, which was achieved in spring of 2012.

1.1.2 The four main experiments at the LHC

The beams in the LHC are not only the most energetic in the world, but also the most intensive. Each collision produces up to thousands of particles and it is a very challenging task to track down all of them. To record all the interactions and to help to understand all the processes which take place in them, four large detectors have been constructed at CERN: ATLAS, CMS, ALICE and LHCb.

The first two experiments, ATLAS (A Toroidal LHC ApparatuS) [2] and CMS (Compact Muon Solenoid) [3] are two general purpose experiments. They have similar physics programs and are very large international collaborations. Both of the experiments are 'general purpose' detectors and their main aim is discovery of the Higgs boson and super-symmetry particles, but they are also capable of solving other physics problems.

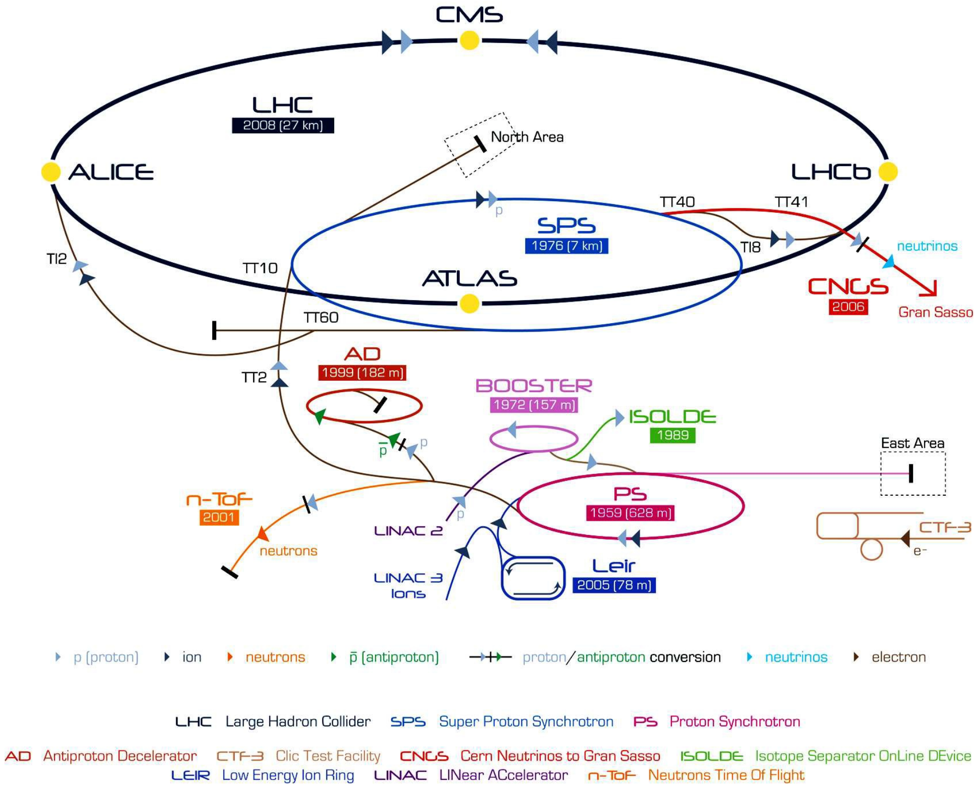


Figure 1.4: Layout of the accelerators [1].

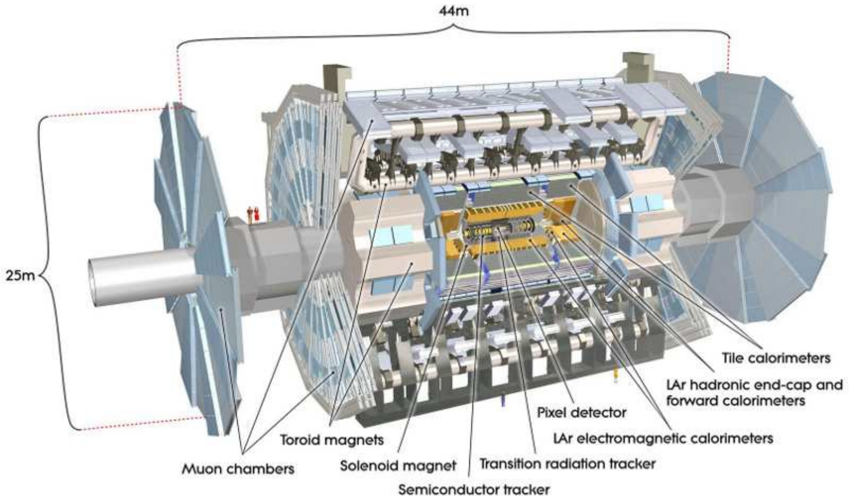

44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

Figure 1.5: ATLAS detector [1].

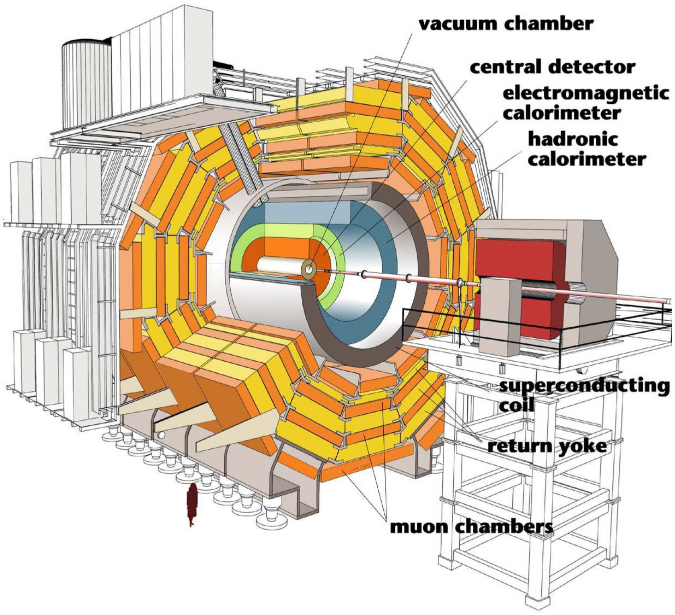


Figure 1.6: CMS detector [1].

The main goal which is pursued by the ALICE (A Large Ion Collider Experiment) [4] collaboration are the studies of quark-gluon plasma (QGP) physics in collisions of heavy ions, a state of matter which, theoretically, existed in the first moments after the birth of our universe.

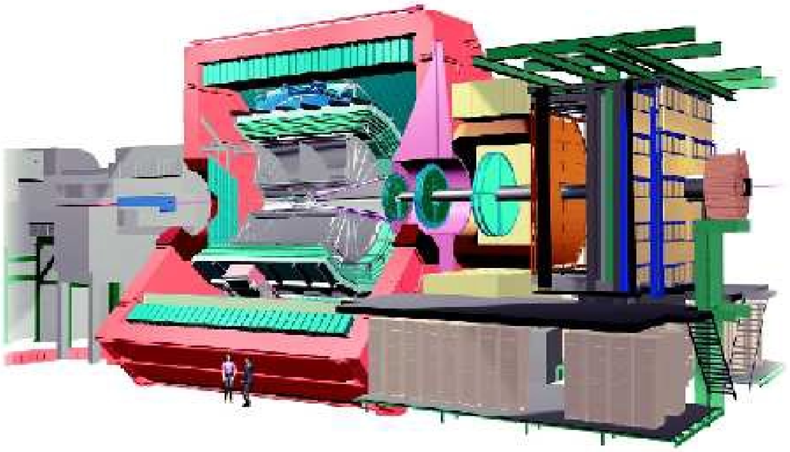


Figure 1.7: ALICE detector [1].

Finally, the LHCb (Large Hadron Collider beauty) detector is designed for B-physics studies, in particular the measurement of CP violation and the study of rare decays. It is described in more details in the next section.

The physics goals which are pursued by the collaborations of these four main LHC experiments do of course overlap, but the results will be achieved in different ways, which in turn helps to cross check them.

Besides these four large detectors there are also three smaller ones:

LHCf (Large Hadron Collider forward) consists of two detectors which are placed at 140 meters distance on both sides of ATLAS [5]. The LHCf collaboration studies forward particles created at the ATLAS interaction point to obtain a better understanding of cosmic ray interactions on the atmosphere.

TOTEM (Total Elastic and Diffractive cross-section) also studies the forward particles and focuses on the elastic scattering over a large range of momentum transfer and cross-sections of diffractive processes [6].

MoEDAL (Monopole and Exotics Detector at the LHC) is located in the same cavern as LHCb and focuses on the search for new physics beyond the Standard Model, in particular, the experiment aims to directly search for Magnetic Monopoles (Dyons) and other highly ionizing Stable (or pseudo-stable) Massive Particles (SMPs) at the LHC.

1.1.3 LHC's luminosity

Each of these detectors are situated in one of the four caverns along the LHC ring, where the accelerated beams of particles could cross and interact with each other. In these beams, particles are not distributed evenly along all the circumference, but are grouped in bunches. Each beam may consist of up to 2808 bunches with each of them containing around 10^{11} protons. Accelerated, they travel the full circle of the LHC ring in under 0.0001 seconds, making more than 10000 turns per second. Bunches stay in fixed positions with respect to each other so they always move synchronously along the accelerator's ring and collide in a specific order in the four nodal points.

One of the important parameters which characterize a collider is its luminosity, which is a measure of how many particles interact during a collision: the higher the luminosity, the more particles interact in the bunches of the colliding beams. Luminosity is defined as:

$$L = f \cdot \frac{n_1 n_2}{4\pi\sigma_x\sigma_y}.$$

This formula accounts for the information on the numbers of particles, n_1 and n_2, in the bunches of proton beams, the bunch profile (the area of the overlap between the two beams), the crossing rate f, the standard deviations of the particle densities in the bunches in the x and y directions σ_x, σ_y, presuming that the distributions are Gaussian.

The designed maximal luminosity for the LHC is $L_{\text{max}} = 10^{34}$ cm^{-2}s^{-1} at 25 ns bunch crossings, and the total number of inelastic collisions N_{pp} for a time period t is equal to:

$$N_{pp} = \sigma_{\text{inel}} \int_t L dt,$$

where $\sigma_{\text{inel}} = 80$ mb is the inelastic cross-section which is expected for proton-proton collision at a centre-of-mass energy $\sqrt{s} = 14$ TeV.

The number of proton–proton interactions per single crossing of two bunches is estimated by the Poissonian distribution:

$$\mathcal{P}(\mu, n) = \frac{\mu^n}{n!} e^{-\mu},$$

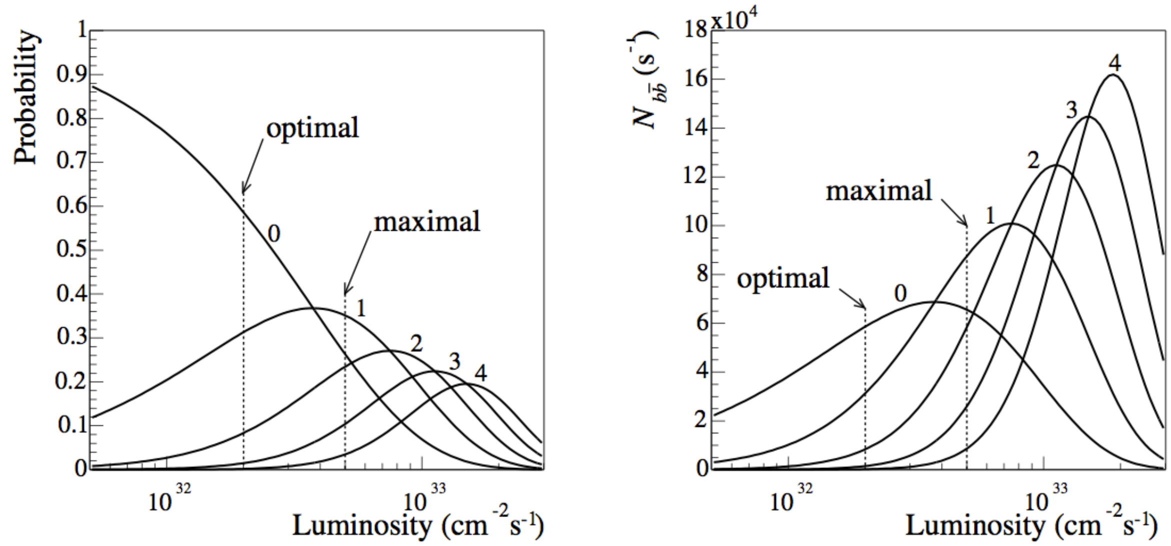


Figure 1.8: Probability for 0, 1, 2, 3, 4 inelastic collisions per bunch crossing versus the luminosity (left). The number of $b\bar{b}$ events per second with 0, 1, 2, 3, 4 pile-up (right) [7].

where μ defines the average number of inelastic interactions per bunch crossing and is directly-proportional to the cross-section of inelastic interactions $\sigma_{\text{inel}}^{pp}$. Probabilities for $n = 0, 1, 2, 3, 4$ of proton-proton interactions as a function of luminosity for $\sigma_{\text{inel}}^{pp}|_{\sqrt{s}=14TeV} = 80$ mb are presented in Fig. 1.8.

Events with a single proton-proton interaction per bunch crossing are the most convenient, clean and simple from the point of analysis. Though the portion of such events is maximal at luminosity $L \simeq 4 \times 10^{32}$ cm^{-2}s^{-1} (see Fig. 1.8), at such luminosity the number of events with $n >= 2$ is very high. On the other hand, higher luminosity means more interactions per event and a bigger chance to see a rare decay, but at the same time creating higher load on the readout, reconstruction system, and making the final analysis more complex. All this means that an optimal value of L varies for different experiments and goals. Currently, LHC is running at 50 ns between bunch crossings and the luminosity for LHCb is $L = 4 \times 10^{32}$ cm^{-2}s^{-1}.

1.1.4 b-quarks at the LHC

At the LHC, in proton-proton interactions, at the energy of $\sqrt{s} = 14$ TeV, the beauty quarks are produced mostly in $b\bar{b}$ pairs. The dominant mechanism is gluon-gluon scattering in which a pair of virtual gluons fuse and form a $quark - antiquark$ pair. The cross-sections for the processes with a single beauty quark production is very small. Table (1.1) demonstrates the estimated cross-sections for the LHC [8].

cross-section	Value
Total	$\sigma_{\text{tot}} = 100$ mb
Inelastic	$\sigma_{\text{inel}} = 80$ mb
$c\bar{c}$	$\sigma_{c\bar{c}} = 3.5$ mb
$b\bar{b}$	$\sigma_{b\bar{b}} = 0.5$ mb

Table 1.1: The estimated cross-sections at $\sqrt{s} = 14$ TeV [8].

Figure 1.9 illustrates the angular distribution of the b and $\bar{b}$ quarks from a $pp \to b\bar{b}X$ process in one event obtained with the PYTHIA [9] simulation application. Since in most cases, the two colliding virtual gluons significantly differ in their momenta, as a result, the produced quarks are strongly boosted in either direction of the beams, so they predominantly propagate in a narrow cone in forward or backward direction. This distribution determines the basic geometry of the LHCb detector.

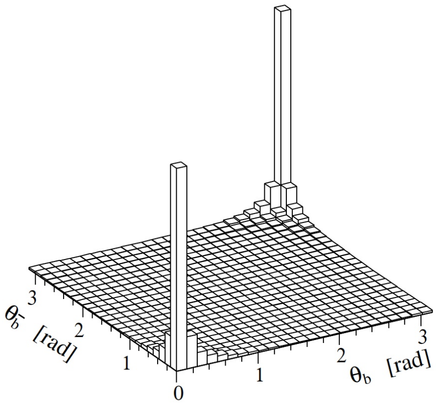


Figure 1.9: The angular distribution of the produced b-hadrons obtained with the PYTHIA simulation program [1].

1.2 The LHCb detector

The LHCb spectrometer, described in [10], is a large detector, optimized for studying B-meson decay, the decay of particles which contain a b-quark. This is indicated by the letter "b" in the name of the experiment, which means "Large Hadron Collider beauty" experiment.

One of the most interesting parts in studying B-mesons is that they exhibit a very important, but so far not fully studied phenomenon of CP-symmetry violation. The interesting fact behind this phenomenon is that its violation leads to slight differences in decays of particles and antiparticles, which directly relates to the origin of our universe and to the answer to the question "Why is it that in our universe matter dominates over antimatter?".

Amusingly, all of this is reflected on the experiment's logo: the symmetrically mirrored image is perturbed by a red line, which crosses the mirrored "Cp" part of the detector's name, hinting at CP violation.

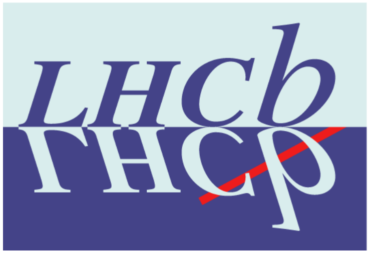


Figure 1.10: The logo of the LHCb detector, hinting on the main goal of the experiment – the study of CP-symmetry violation [1].

1.2.1 Setup of the LHCb experiment

In contrast to the large ATLAS and CMS detectors, which are general purpose detectors and feature barrel shapes, surrounding the interaction point of the colliding particles, the LHCb detector is a single-arm spectrometer and has the form of a cone with its tip exactly in the point of the colliding beams. It tracks particles, which propagate within the specific angular range with respect to the beam axis and its angular geometrical acceptance is approximately from 10 to 300 mrad in the bending plane of the magnetic field of its dipol magnet, and from 10 to 250 mrad in the perpendicular plane. The length of the LHCb detector is 21 meters and its mass is around 5600 tonnes.

The detector consists of multiple layers and its sequence is more or less canonical, however since it has a very specific objective (the study of B-mesons) its components are optimized for this specific task.

The overall layout is presented in Fig. 1.11. The vacuum pipe of the LHC accelerator traverses the detector and the accelerated particles collide in the left part of the apparatus inside the vertex locator, VELO. The collision products which happen to propagate within the detector's angular coverage travel further on to the right through one of the two Cherenkov detectors, RICH1, through the magnetic field of the large magnet which deflects trajectories of charged particles, then on their way they pass the three tracking stations, the second Cherenkov detector RICH2, arriving at the calorimeters and finally at the muon chambers.

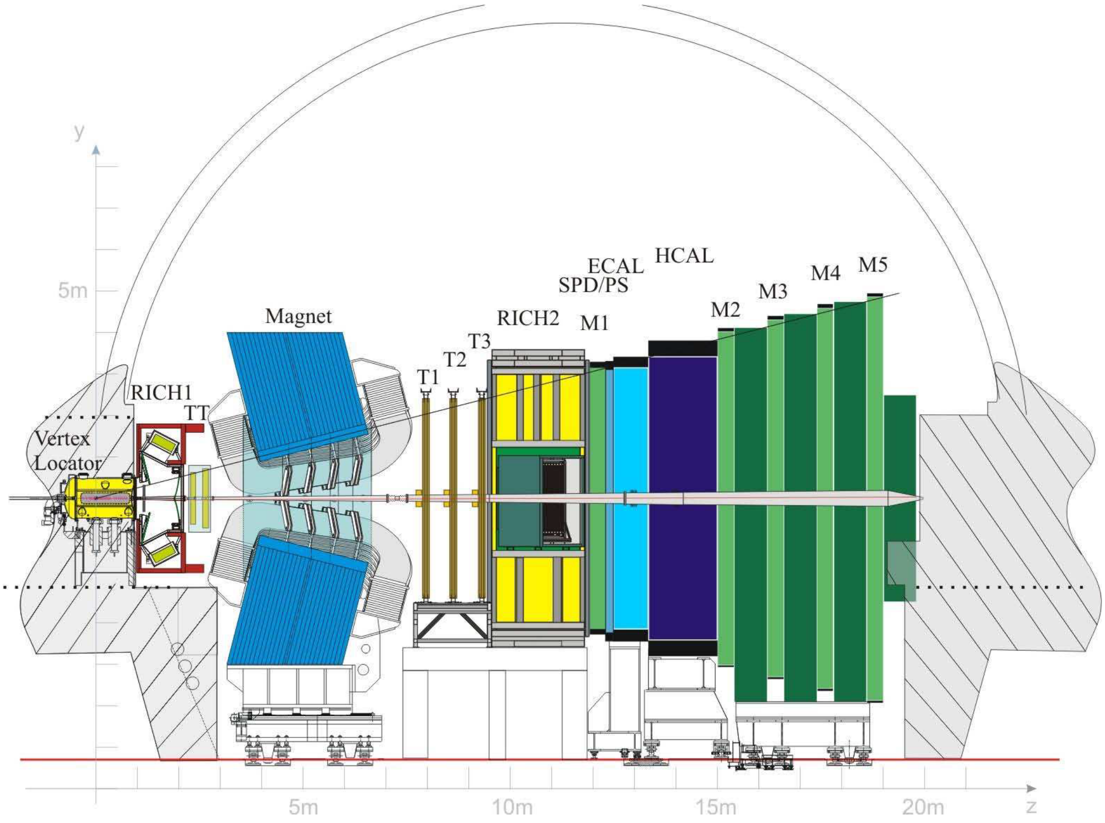


Figure 1.11: The LHCb detector layout, from left to right: the vertex locator (VELO), RICH1 & RICH2 – the two Ring Imaging Cherenkov detectors, the dipol magnet, TT, T1, T2, T3 – the four tracking stations, SPD/PS – the Scintillator-Pad Detector and PreShower, ECAL & HCAL – the Electromagnetic and Hadronic CALorimeters, M1-M5 – the Muon stations [11].

The subdetectors, the components of the LHCb detector which are described further on, can be divided into the following two categories:

- **Tracing detectors**: Vertex Locator (VELO), Tracker Turicensis (TT), Inner Tracer (IT), Outer Tracker (OT).
- **Particle Identification detectors (PIDs)**: the two Cherenkov detectors (RICH1, RICH2), Scintillating Pad Detector (SPD), PreShower (PS), Electromagnetic CALorimeter (ECAL), Hadronic CALorimeter (HCAL) and the muon stations (M1-M5).

Figure 1.11 shows the layout of the subdetectors in the LHCb machine in the $y--z$ plane, where the z-axis on the image is the axis of the particle beams, and the y-axis is perpendicular to the bending plane.

When the charged particles travel through the dipol magnet, their trajectories are bent by the magnetic field, which allows to measure their momentum [12]. The main component of the magnetic field B_y is shown as a function of z in the Fig. 1.12 below.

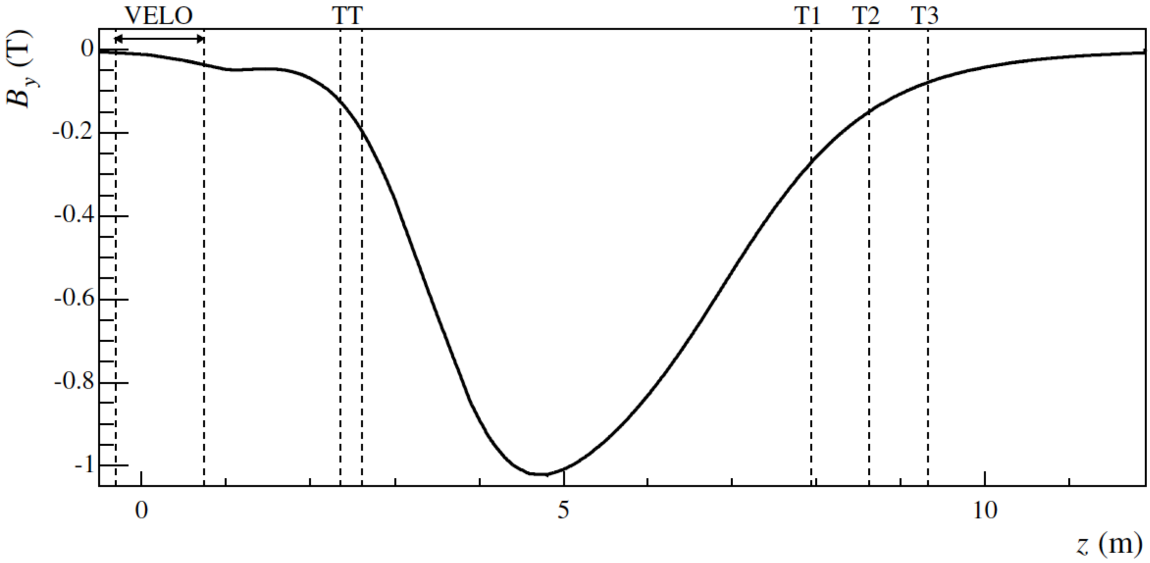


Figure 1.12: The main component of the magnetic field strenght B_y along the z-axis [11].

1.2.2 VELO

Of all the subdetectors of LHCb the VELO (VErtex LOcator [13]) detector is the one located closest to the point of collisions of the beams. It consists of two halves which are positioned symmetrically around the beam pipe and are retracted during injection, usually 30 mm, to keep the detector in the "opened" state during particle injections into the LHC, or closer to the beam at 7 mm for the "closed" state during data recording. Each VELO half contains 21 stations which are positioned along the beam pipe and at the same time perpendicular to the beam axis, and carry silicon modules with double-sensors which measure r and ϕ coordinates. The advantage of such a geometry is that the $r-z$ measurement allows a fast measurement of the impact parameter d_0 with respect to the primary vertex. The trigger system uses this approach to select only those tracks from the $r-z$ projection which have a large impact parameter, to reconstruct them in two dimensions. The impact parameter resolution σ_{d_0} is a very important characteristic of the detector and for the VELO, σ_{d_0} has been estimated for three momentum ranges which have been arbitrarily selected to separate particles with low, medium and high momentum [14]:

- The low momentum range ($p < 5$ GeV): $\sigma_{d_0} = 8.5 \pm 0.2\ \mu$m
- The medium momentum range (8 GeV < p < 15 GeV): $\sigma_{d_0} = 7.8 \pm 0.2\ \mu$m
- The high momentum range ($p > 18$ GeV): $\sigma_{d_0} = 8.6 \pm 0.2\ \mu$m

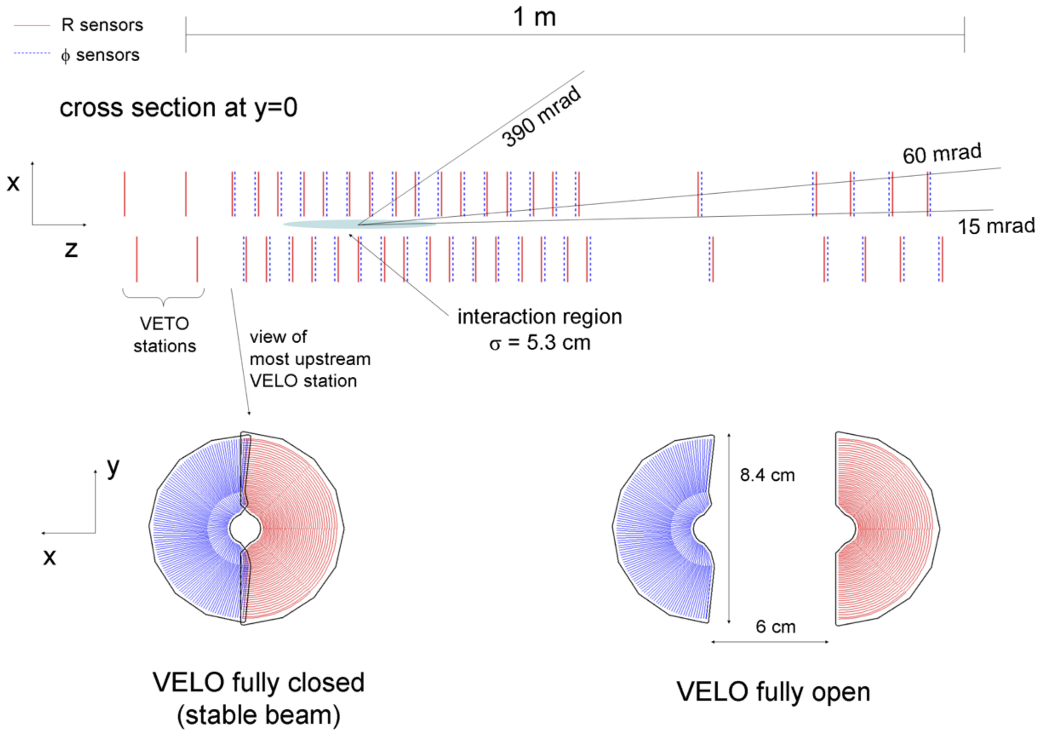


Figure 1.13: VELO detector crossection (view from above) and half-disc sensors [11].

There are also two half-disc stations which are placed in the backward part of the VELO detector and feature only r-sensors. They act as a part of the L0 trigger by allowing quick identification of collisions with several proton-proton interactions: since the magnetic field in the detector is weak, it is possible to use fast track-reconstruction algorithms in the trigger system, assuming that the tracks are not bent. The stations are grouped in two clusters: the inner one, which is closer to the interaction point and has a smaller pitch between the stations, thus increasing the density of the silicon modules, and the outer one with wider stations positioning. The cross-section of the VELO detector and it's half-disc sensors are shown on the Fig. 1.13.

1.2.3 Tracker Turicensis

The Tracker Turicensis (TT) [15] is the next tracking detector. It is positioned in front of the large dipol magnet, right after the first Cherenkov detector (RICH1). The detector is housed in a single large box which provides electrical and thermal insulation. It is constructed using

$500\mu m$ thick sensors with a pitch of $183\mu m$ and 512 strips. The detector consists of four layers with the strips oriented in stereo views, $x - u$ and $v - x$, corresponding to angles with the vertical y axis of 0°, 5°, $-5°$ and 0°. The stereo views allow a reconstruction of tracks in three dimensions. The layout for four planes is shown in Fig. 1.14. The silicon sensors are bonded together to form readout sectors of up to four sensor. In the figure the granularity of the readout is indicated with different shading, the highest around the beam pipe. The inner readout sectors are connected with up to 56 cm long cables to the respective front-end readout chips located outside the acceptance in the upper and lower parts of the detector.

The low integrated magnetic field between the VELO and the Tracker Turicensis makes it possible to assign a rough momentum estimate to the tracks. The TT also estimates momentum of slow particles, which are bent out of the LHCb acceptance before they reach the tracking stations on the other side of the dipol magnet, and for those particles which reach them, the TT improves their momentum estimate.

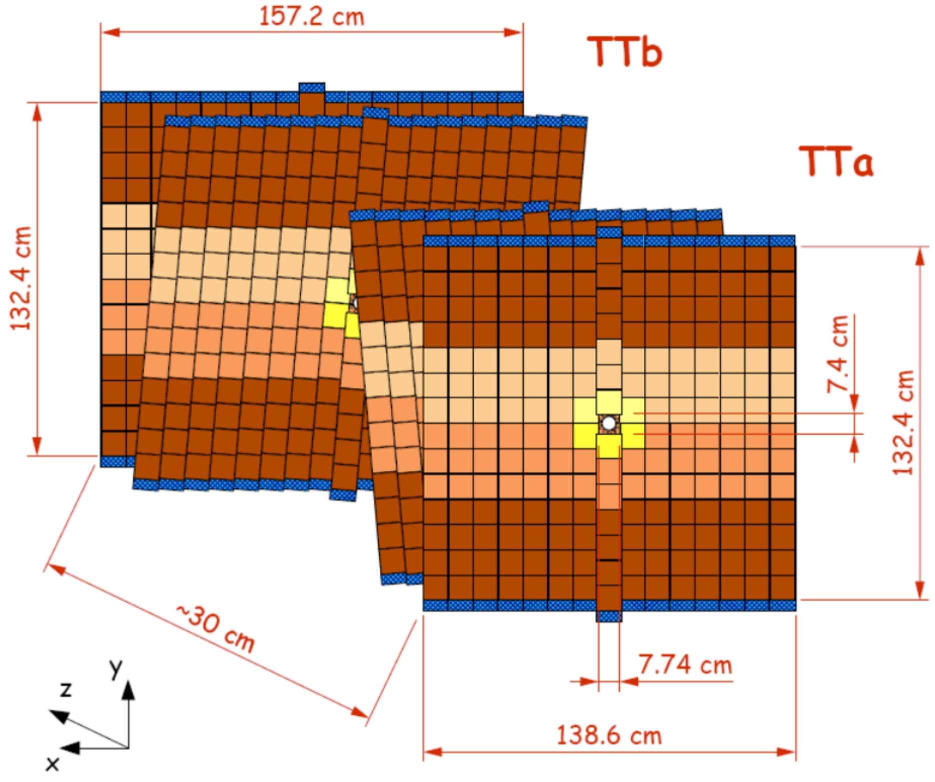


Figure 1.14: The Tracker Turicensis stations [15].

1.2.4 T stations

While the high-precision silicon Vertex Locator detector is used to determine the impact parameters of charged particle tracks and directions of their momenta, the task of measuring parameters of the charged particles rests with the three tracking stations T1-T3, which are positioned right behind the dipol magnet. Each of the stations is divided into two parts: Inner Tracker (IT) and Outer Tracker (OT).

The Inner Tracker [16] is constructed similar to the Tracker Turicensis, with each IT station consisting of four boxes of silicon sensors. The boxes are grouped around the beam pipe in a shape of a cross (Fig. 1.15). Each station box contains four layers in a topology similar to that in the TT.

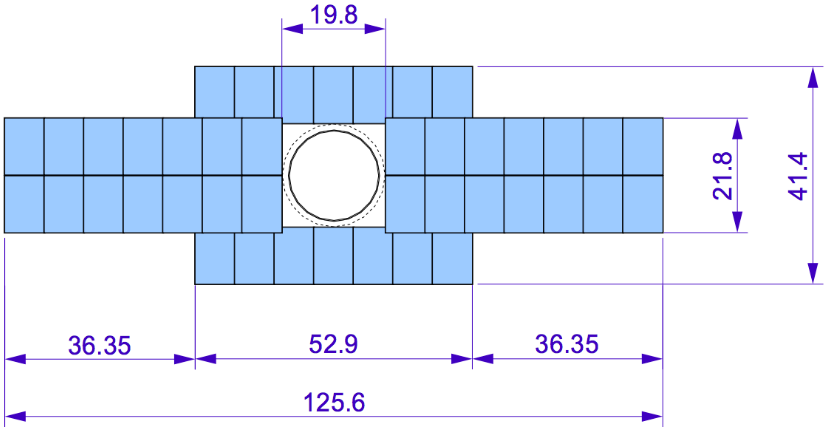


Figure 1.15: The layout of one of the layers of the Inner Tracker with the silicon sensores in the cross-shaped configuration, positioned around the beam pipe [16].

The Outer Tracker (OT) [17] stations cover a much larger region, beyond the acceptance of the Inner Tracker. Similar to the IT and the TT, each Outer Tracker station has four detection layers built of blocks of modules. As depicted in Fig. 1.16, there are two types of modules: eight shorter ones, named S1-S3, which fill up the area above and below the IT, and longer modules (L), seven of which are adjacent to each side of the IT station, covering the rest of the acceptance area of the OT.

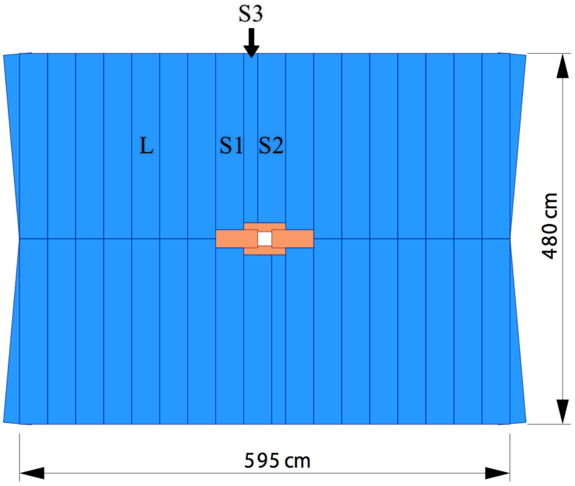


Figure 1.16: Layout of one of the Outer Tracker stations (front view), with the four modules of the Inner Tracker station pictured in the center.

Each module is composed of two rows of straw-tubes – cylindrical drift cells of 5 mm diameter (Fig. 1.17) with a drift-time of less than 50 ns, filled with a mixture of Argon and CO_2 gases. In total, there are 96 and 168 short and long modules and about 55000 single-straw channels in the Outer Tracker.

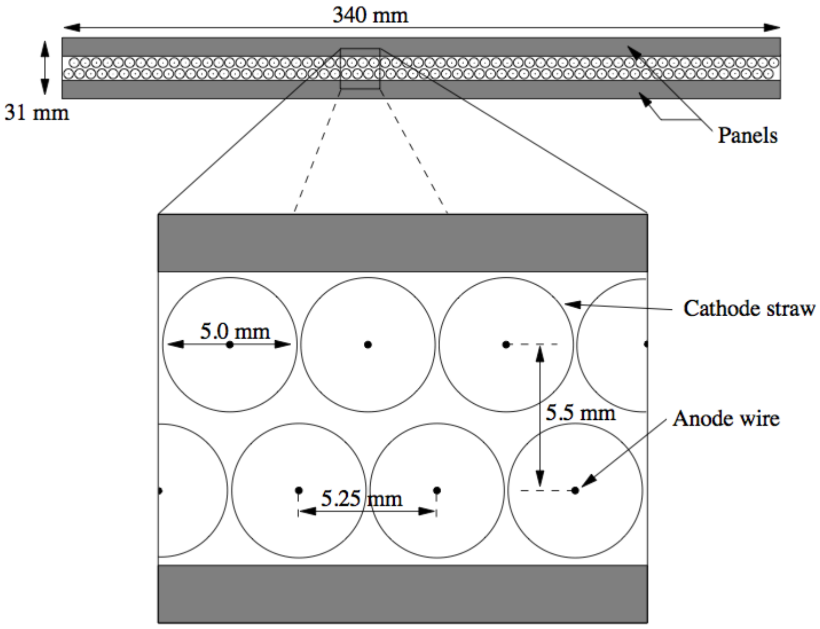


Figure 1.17: The cross-section of the Outer Tracker module with 128 single-straw channels [7].

To give an understanding of the performance of the tracking system, Fig. 1.19 presents the momentum and impact parameter resolutions of reconstructed long tracks. In the LHCb track classification, long tracks are those which have hits in most of the tracking detectors: in VELO and T stations, which make them the best kind of tracks for analysis. Different types of tracks are presented in Fig. 1.18. The second type of tracks usually used for analysis are the downstream tracks, which have no hits in the VELO but are registered in TT and T stations. Other kinds of tracks are often used for technical purposes. Such as for example T tracks are used for alignment and VELO tracks are used in vertexing.

The momentum resolution is plotted in Fig. 1.19 as a function of the track momentum. It varies from $\delta p/p = 0.35\%$ for low momentum tracks to $\delta p/p = 0.55\%$ for tracks with higher momentum. The impact parameter resolution is plotted as a function of inverse transverse momentum ($1/p_{\mathrm{T}}$) of the track and its linear dependence can be parametrized as $\sigma_{\mathrm{IP}} = 14\mu\mathrm{m}+35\mu\mathrm{m}/p_{\mathrm{T}}$ with p_{T} in GeV/c. The lower plots in the figure provide comparison to the momentum spectrum for B decay tracks and the $1/p_{\mathrm{T}}$ spectrum of B decay particles in the detector acceptance.

1.2.5 RICH detectors

The Ring Imaging Cherenkov detectors (RICH) [18] are designed for separation of $K^{\pm}$, $\pi^{\pm}$-mesons, and protons in a wide momentum range from about 3 GeV/c up to 100 GeV/c. This momentum interval includes kaons from cascade decays of charm-particles, which are used for beauty-particles flavor tagging, and charged hadrons from multi-particle decays of B-mesons.

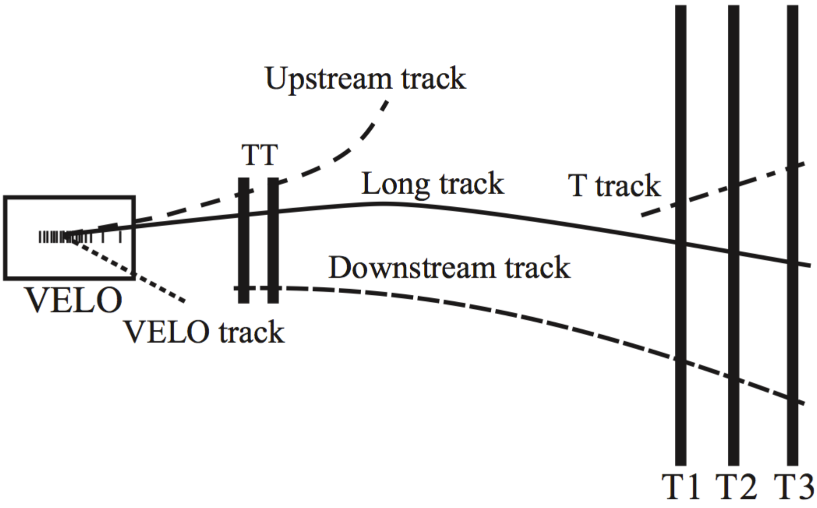


Figure 1.18: LHCb track types [11].

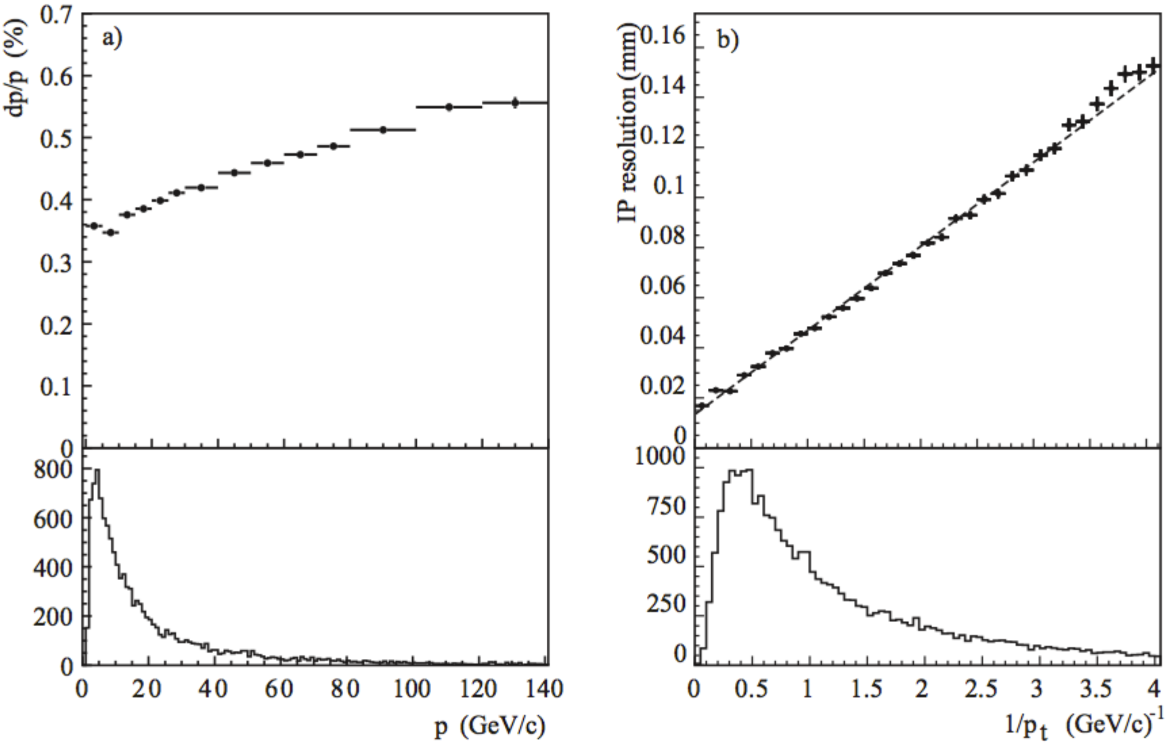


Figure 1.19: Resolution on the reconstructed track parameters at the production vertex of the track: (a) momentum resolution as a function of track momentum, (b) impact parameter resolution, calculated as the sum of the errors in the three projections added in quadrature, as a function of $1/p_T$. The p and $1/p_T$ spectra of B decay particles in the detector acceptance are shown in the lower part of the corresponding plots for comparison [11].

Particle identification in the LHCb detector is done by two such Cherenkov detectors (Fig. 1.20): RICH1, positioned behind the Vertex Locator, and RICH2, mounted between the last T station and the calorimeters.

The principle of the RICH detectors is based on measuring the velocity of the traversing charged particles. In the RICH detectors, the traversing charged particles travel in a medium, in which their velocities are greater than the speed of light in that medium, and emit Cherenkov photons in a cone in the direction of their flight. Those photons are redirected to photomultipliers via a system of mirrors and thus the particle's velocity is found by measuring the emission angle of Cherenkov photons with respect to the particle's trajectory. Fig. 1.21 presents an example of a typical simulated event as it is seen in the RICH1 and RICH2 detection planes.

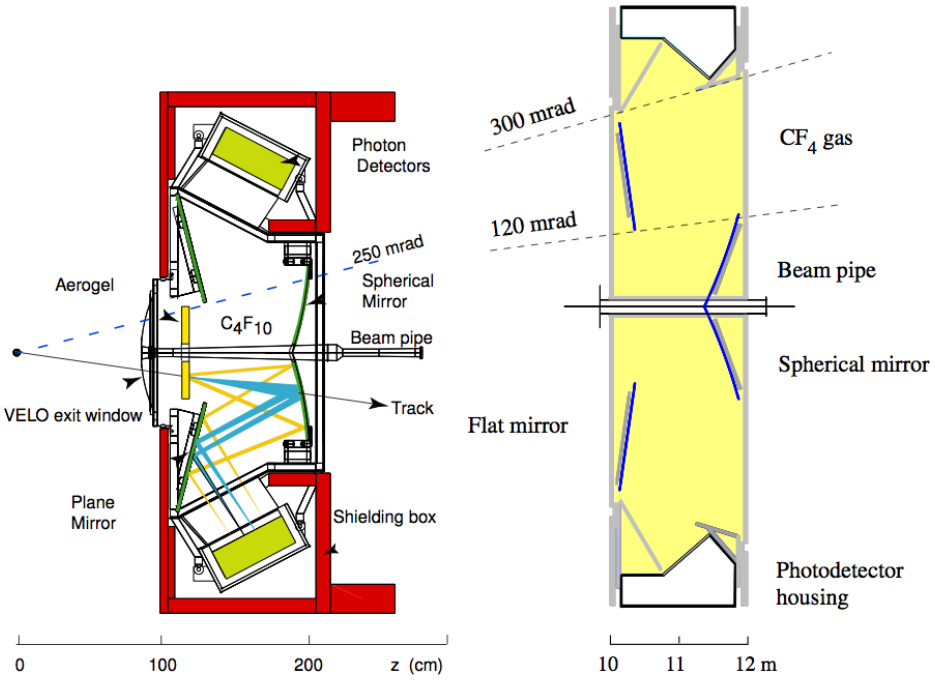


Figure 1.20: The vertical cross-section of the RICH1 detector (left) and the horizontal cross-section of the RICH2 detector (right) [7].

The RICH1 is filled with C_4F_{10} gas and uses silica aerogel, a solid and light material which is only two times more dense than air. Aerogel's refraction coefficient is between 1.01 to 1.10, which allows precise measurements of velocities in the range of 0.95 to 0.99 of the speed of light, i.e. to well identify particles with low energy of a few GeV. The gas' refraction coefficient is ≈ 1.001, which is better suited for measuring velocities of particles with higher energies, from dozens to a hundred GeV.

For the RICH system, the average efficiency for kaon identification for momenta between 2 and 100 GeV/c is $\epsilon(K \rightarrow K) \sim 95\%$, with a corresponding average pion misidentification rate $\epsilon(\pi \rightarrow K) \sim 5\%$. Around 30 GeV/c the identification probability is $\sim 97\%$ and the misidentification probability $\sim 5\%$ [18].

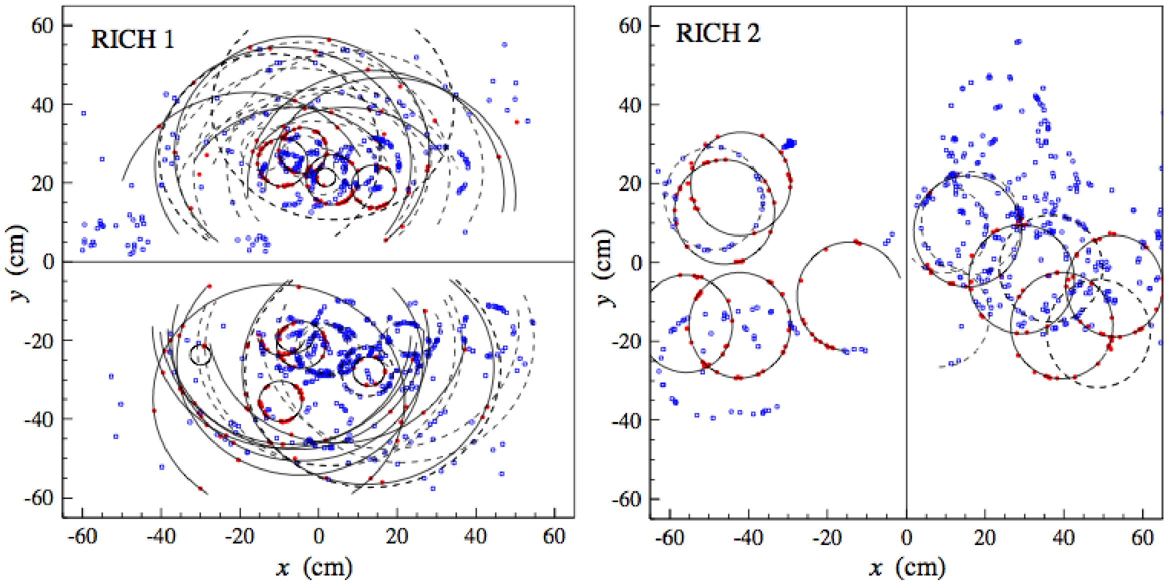


Figure 1.21: A typical simulated event in the detection planes of RICH1 (left) and RICH2 (right). The horizontal (vertical) line separates the two detection planes in RICH1 (RICH2). Superimposed are the reconstructed rings for tracks which extend from the VELO up to the last T station (solid) and all other tracks (dashed) [7].

1.2.6 LHCb's magnet

LHCb's large dipol magnet, which weights around 1600 tons and creates a magnetic field of about 4 Tm, is positioned between the Tracker Turicensis and the T stations. Its curved form, similar to a shark's open jaw, was chosen to achieve a more homogeneously distributed magnetic field on its sides. The opening angle of the magnet follows the polar angular coverage of the detector, which is from 10 to 300 milliradians (mrad) in the horizontal and 250 mrad in the vertical plane. The purpose of the magnet is to measure the momenta of the charged particles: as their tracks are deflected by the magnetic field, with help of the tracking detectors it is possible to estimate the particle's momentum with good precision.

Figure 1.22 shows the magnet's coil, which was constructed from 15 separate mono-layer pancakes. Each pancake was wound from a single length conductor of about 290 m length, extruded from pure aluminium in an annealed state, and feature channels for water cooling.

1.2.7 Calorimeters

The LHCb calorimeter system consists of several detectors: Scintillating Pad Detector, Pre-Shower detector, Electromagnetic Calorimeter and Hadronic Calorimeter [19]. The two main parts are the electromagnetic and hadronic calorimeters. The ECAL measures the electromagnetic showers of electrons and photons, and the HCAL is responsible for measuring the hadronic showers of pions, kaons and protons. The SPD and PS are positioned in front of the ECAL and act as additional detection layers, which provide information on the longitudinal evolution of the electromagnetic showers, improving the particle identification of the ECAL.

The functioning principles of the calorimeters are similar. All the particles which traverse the LHCb detector, except for the muons and neutrinos, are absorbed in the calorimeters. When the

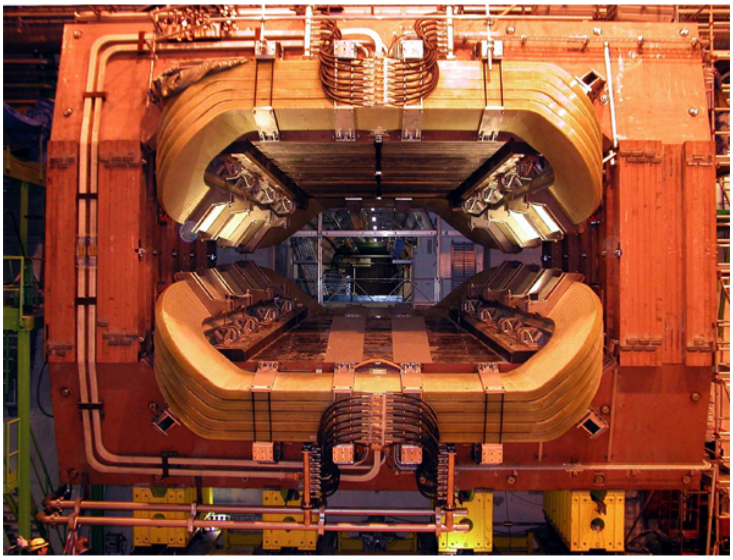


Figure 1.22: Frontal image of the LHCb's dipol magnet [1].

particles interact with the bulk matter of the calorimeters, they produce cascades of secondary particles, which excite atoms in the scintillating material, forcing them to emit scintillating light as they return to their ground state. This amount of scintillating light is proportional to the initial particle's energy.

The resolution of the measured energy E by the ECAL is given by

$$\frac{\sigma(E)}{E} = \frac{10\%}{\sqrt{E}} \oplus 1\%,$$

and the resulting resolution for the HCAL can be estimated as

$$\frac{\sigma(E)}{E} = \frac{(65 \pm 5)\%}{\sqrt{E}} \oplus (9 \pm 2)\%,$$

where E is in GeV and $\oplus$ is the quadratic sum.

1.2.8 Muon system

Detecting muons is very important for LHCb because B-decays of high interest, such as $B_d^0 \to J/\psi(\mu^+\mu^-)K_s^0$ and $B_s^0 \to J/\psi(\mu^+\mu^-)\phi$, contain a muon pair in their final states.

Since the muons are the only charged particles which traverse the whole of the LHCb detector, there is a separate system dedicated to detecting them – the muon detector. It consists of five stations, four of which, M2-M4, are located right behind the HCAL and are separated by thick iron plates, which act as filters for the hadronic background, and the first station M1 is placed in front of the SPD/PS system [20]. These muon stations are instrumented with Multi Wire Proportional Chambers (MWPCs) and in total there are about 1500 MWPCs, with a total active surface of 435 m^2.

The information of the muons with a high enough transverse momentum is used in the Level-0 trigger and based on its decision potentially interesting events are recorded for further analysis. Evidence of these interesting events with B-mesons decays will never surface clearly. Fig. 1.23 illustrates an example of a modeled $B_s^0 \to J/\psi(\mu^+\mu^-)\phi$ decay, in which traces of the decay are lost among the hadrons, produced in the same primary proton-proton interaction. However, the detected muons (the blue tracks on the image) signal that the event could be a good candidate for the particular analysis and trigger it for recording.

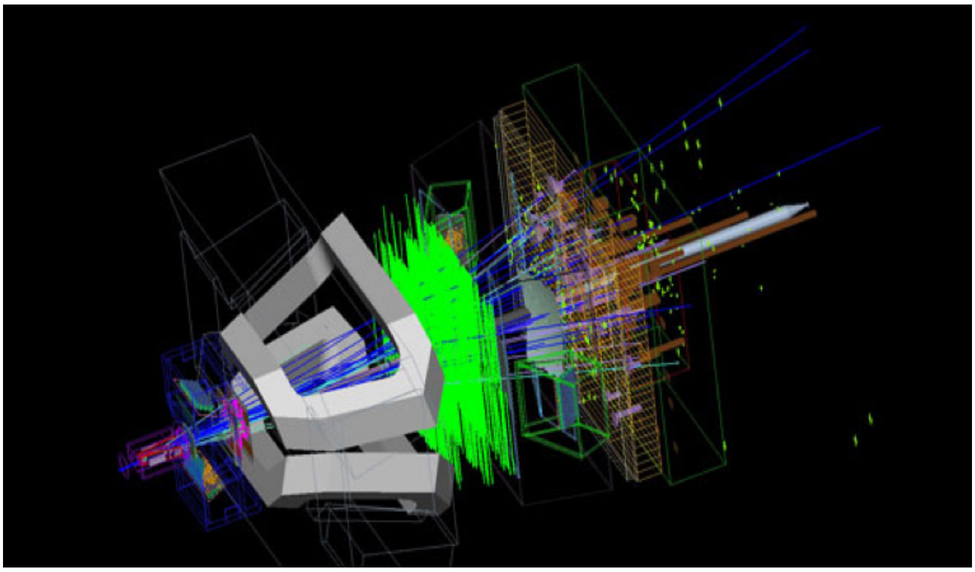


Figure 1.23: Modeled proton-proton collision event which among other decay products has a B_s-meson produced and decayed [1].

1.2.9 LHCb Trigger

The analog front-end system of the LHCb detector delivers data to the trigger system at a rate of up to 40 MHz. Even with modern cutting edge storage technologies, recording all this data would be quite a challenging task, not to mention – a very costly one, plus, a necessity not only to record, but also to fully reconstruct all these events at such a high input rate, would require serious investment into additional readout electronics and a powerful computational farm. The amount of recorded data can be significantly reduced since only a small fraction of the events is interesting from the perspective of the experiment. The production cross-section of b-quarks at the LHC energy is about ~ 0.5 mb, which is a small fraction of the total cross-section of inelastic interactions of estimated 80 mb (see Tab. 1.1). The LHCb trigger system allows to effectively separate the $pp \to (b\bar{b})X$ events from the rest of the inelastic interactions.

For the reason that there is not enough time to fully reconstruct all the data, the events, which are considered to be better candidates for further analysis, are selected in several successive stages. These stages, or levels, increase in complexity of criteria which the events have to pass: initially, rather loose cuts are imposed on just a few available reconstructed variables, in order to reduce the obvious background events, and later more sophisticated cuts are introduced to distinguish signal-like background from the real B events. Because of this staged design, the LHCb trigger consists of three separate subsections:

Level-0 (L0): the hardware level trigger, which reduces the readout rate from up to 40 MHz to below 1.1 MHz with a fixed latency of 4 μs.

HLT1: the software level trigger, which uses the data selected by L0 as the input and further reduces the data rate to ~ 43 kHz with about 1 ms processing time.

HLT2: the software level trigger, which uses the full event information and in 2011 had an output rate of 3 kHz with around 10 ms processing time [21].

The L0 itself consists of three trigger subsystems: the calorimeter, muon and pile-up triggers, and one additional decision unit, which collects information from the three trigger subsystems and makes the final L0 decision. The calorimeter trigger takes as input information the hadron, electron and photon clusters with the highest transverse energy E_{T} in calorimeters. The muon trigger requires data from the muon stations and looks for muon tracks which have a high p_{T}. The pile-up system consists of the two most upstream VELO stations, described in Sec. 1.2.2. The purpose of the system is to tag those bunch crossings which have multiple collisions. In the end, information from these subsystems are collected by the decision unit, which determines whether to accept or discard the event.

The Level-1 is one of the two software triggers which utilizes a computational farm (installed with consumer-level CPUs), shared with the HLT2 trigger. In the case when the L0 trigger accepts an event, the event data is passed from the subdetectors to the HLT1 trigger, which may buffer up to around 58000 events and in contrast with the L0 HLT1 has a variable latency. The HLT1 decision depends on several sources which account information provided by the L0, the Vertex Locator and the tracking subdetectors. This increased verbosity impacts the event data size footprint: if in the case of the L0 an event is described by a data structure of 5 kB, on HLT1 level the size grows up to 50 kB [7]. The algorithm behind the trigger also becomes more complex: the HLT1 aims to search for tracks of B decay which have a large transverse momentum combined with a high impact parameter.

The final stage of the trigger system is the HLT2, which is used at the last step of the data acquisition and decreases the event flow, down to 3.5 kHz, but it uses feedback from all of the LHCb subdetectors, thus the size of the events during this stage increases to 100 kB. The HLT2 uses information from the tracking stations to re-evaluate the decision carried out by the HLT1 trigger and during this step the event rate is already reduced by a factor of two without significant loss of signal events. In the next steps HLT2 requires even more reconstructed information such as a full 3D reconstruction of tracks registered by the VELO, their extrapolation to find hits in the Tracker Turicensis and the tracking stations, and information on the particle identity for those tracks which are considered as electron or muon candidates. In general, the procedure executed by the HLT2 trigger could be characterized as a common offline analysis with rather loose constraints on the events parameters.

Chapter 2

CP Violation and LHCb

De rerum natura.

Lat.: On the nature of things.

– Titus Lucretius Carus

This chapter presents a brief introduction into B-physics, the understanding of which is crucial for such experiments as LHCb. Section 2.1 presents a short explanation of the Standard Model, followed by Sec. 2.2 which describes the Cabibbo-Kobayashi-Maskawa matrix, CP violation and the unitarity triangles. Finally, Sec. 2.3 further discusses CP violation and possibilities for new physics at the LHCb.

2.1 The Standard Model

2.1.1 Theories in particle physics

The vast variety of physical phenomena, which arise in collisions of particles, are determined by only the following four types of interactions: electromagnetic, strong, weak and gravitational (Tab. 2.1). The most important in particle physics are the first three.

To describe the processes which occur in the world of particles, scientists use the relativistic quantum theory or quantum field theory. In this theory, the interactions are described in terms of exchange of quanta (bosons), which are associated with a particular type of interaction. The most well-known variant of this theory is used for the description of electromagnetic interactions and is called quantum electrodynamics (QED). QED is the most advanced of all physics theories. It provides record accurate predictions (to the scale of 10^{-9} to 10^{-10}) and so far there has been no sign of any facts which would be in conflict with it.

The weak interaction involves the exchange of intermediate bosons $W^{\pm}$, Z; it is responsible for the slow decay of particles. At present, there has been developed a theory which unified the electromagnetic and weak interactions into one, it is the electroweak theory. Glashow, Salam and Weinberg were awarded the Nobel prize in 1979 for its creation.

Another theory, called quantum chromodynamics (QCD) was proposed for the description of strong interactions. Strong interactions involve the exchange of gluons, i.e. color. QCD is the theory of color fields, in which the coupling strength is proportional to the colors of quarks (color charges) and equals to zero for colorless states.

In this way, electroweak theory and quantum chromodynamics together describe strong, electromagnetic and weak interactions of quarks and leptons, and form the Standard Model (SM).

The types of interactions are characterized by the coupling constant α_i (Tab. 2.1), which is a quantitative characteristic of the interaction strength acting between particles.

Interaction	Involved particles	Quantum field (gauge bosons)	Mass (mc^2) [GeV]	Relative strenght	Interaction radius [m]	Characteristic time of interaction [s]
Strong	hadrons	g (8 gluons)	0	$\alpha_s \approx 1$	$\approx 10^{-15}$	$10^{-21} - 10^{-23}$
Electromagnetic	charged particles	γ (photon)	0	$\alpha_e \approx 1/137$	∞	$\sim 10^{-18}$
Weak	leptons and quarks	$W^{\pm}$ Z	80 91	$\alpha_w \approx 10^{-6}$	$\approx 10^{-18}$	$\sim 10^{-10}$
Gravitational	all particles	graviton (hypothetical)	0	$\alpha_g \approx 10^{-38}$	∞	N/A

Table 2.1: Types and properties of the fundamental interactions

2.1.2 Systematics of particles

According to the summary tables of the Particle Data Group (PDG) Review of Particle Physics journal [22], there are about 500 known particles, but in fact all of this variety can be composed of only twelve fermions: six quarks and six leptons, which take part in different interactions and exchange the four bosons (photon γ, gluon g, W and Z). These 12 fermions, which have spin of 1/2, are naturally divided into three groups, which are usually referred to as generations. In each generation there are 2 quarks and 2 leptons (Tab. 2.2 and 2.3).

Generation I	electron (e)	electron neutrino (ν_e)
Generation II	muon (μ)	muon neutrino (ν_μ)
Generation III	tau (τ)	tau neutrino (ν_τ)

Table 2.2: Leptons

There are in total six quarks, i.e. six different flavors. Their names, electric charges and approximate masses are presented in Tab. 2.3. Some of the main properties of quarks are:

- Quarks have fractional electric charge values, $\pm\frac{2}{3}e$ or $\pm\frac{1}{3}e$ (where e is the elementary electric charge).
- The strong interaction between quarks is due to an exchange of gluons (in total there are 8 gluon states).
- The potential energy of two quarks (color force) increases with distance between them. This phenomenon leads to color confinement (often simply confinement), the phenomenon

that isolated quarks (or any color charged particles) can not be directly observed. When there is a large distance between two quarks, their interaction energy is large enough, such that new quark-antiquark pairs can be produced, which leads to the formation of new bound quark states.

Quark type (flavor)	Charge e	Mass [MeV/c^2]	Generation	Name
u	$+2/3$	$1.7 - 3.1$	I	up
d	$-1/3$	$4.1 - 5.7$		down
c	$+2/3$	1290^{+50}_{-110}	II	charm
s	$-1/3$	100^{+30}_{-20}		strange
t	$+2/3$	$172900 \pm 600 \pm 900$	III	top
b	$-1/3$	4190^{+180}_{-60}		bottom

Table 2.3: Quarks

Modern physics considers quarks and leptons point objects, meaning they have no structure. They are called the fundamental fermions and quarks are the building blocks of the hadron particles.

The four bosons (γ, g, W, Z) have spin 1 and are also considered to be structureless, fundamental particles. They are called fundamental or gauge bosons and are quanta of fundamental fields. Thus our world can be reduced to fundamental fermions which interact by means of gauge bosons.

Another class of particles, which are significantly larger in size ($\approx 10^{-13}$ cm), are called hadrons and consist of quarks. There are two types of hadrons:

Baryons are built of three quarks ($q_i q_j q_k$), not necessarily different in flavor. Baryons have a special characteristic called baryon quantum number (or charge) $B = 1$, and half-integer spin, which makes them fermions.

Mesons consist of a quark and an antiquark ($q_i \bar{q}_j$), their baryon charge is nil ($B = 0$) and their spin is integer, so mesons are bosons.

In this manner, protons consist of two u-quarks and one d-quark: $p = uud$, neutrons have two d-quarks and a single u-quark: $n = udd$, π-mesons $\pi^+ = u\bar{d}$ and $\pi^- = \bar{u}d$, and the most important for the LHCb experiment: $B^0_d = d\bar{b}$ and $B^0_s = s\bar{b}$.

2.1.3 Symmetry

The CPT theorem is a fundamental result of quantum field theory. The meaning of the theorem can be reduced to the following statement: our world and the world, which is created from our world via charge conjugation (C), parity transformation (P) and time reversal (T), are equivalent. In other words, our world and its mirror reflection, in which all particles are replaced with their respective antiparticle-counterparts and directions of movement inverted, have identical laws. According to this theorem, any conceivable Hamiltonian is invariant under the CPT transformation.

CP violation and the CPT theorem together lead to an interesting dilemma: either there is no T invariance (which should be violated if the CPT theorem is correct), or the CPT theorem is wrong.

The decay of neutral kaons is not the only known process in which CP violation is observed. The research of B^0-mesons showed that in case of decays of these B-mesons, CP violation is also present. Explaining the reason of the baryon asymmetry in our universe and CP violation are fundamental challenges of modern physics and one of the main goals of the LHCb experiment.

The next section of this chapter describes CP violation in the Standard Model and the CKM matrix in more detail, and since B-mesons are well suited for CP violation measurements, their decays are used as an example for explaining the CP violation phenomenon.

2.2 The Cabibbo-Kobayashi-Maskawa matrix and CP violation

The mixing of quarks in weak interactions is the source of CP violation. If (d', s', b') and (d, s, b) are the flavor and mass eigenstates, they are related by the Cabibbo-Kabayashi-Maskawa (CKM) mixing matrix V_{CKM} [23], according to

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = V_{\text{CKM}} \begin{pmatrix} d \\ s \\ b \end{pmatrix}$$

with a 3 × 3 unitary matrix

$$V_{\text{CKM}} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}. \tag{2.1}$$

The squared values of the matrix elements V_{ij}, $i = u, c, t$, $j = d, s, b$ correspond to the relative probability of transition of one of the quarks into another, i.e. $|V_{us}|^2$ is the relative probability for an $u \leftrightarrow s$ transition in weak charged quark interactions.

In the general case this matrix is complex. The CKM matrix can be parameterized with the help of tree angles and one phase, which is the unique source of CP violation in the Standard Model. There are several different parameterization variants of the CKM matrix. The most widely used is the parameterization, proposed by Lincoln Wolfenstein [24]. In his approach, Wolfenstein introduced the four parameters λ, A, ρ, η [22]:

$$\lambda = \frac{|V_{us}|}{\sqrt{|V_{ud}|^2 + |V_{us}|^2}}, \qquad A\lambda^2 = \lambda \left| \frac{V_{cb}}{V_{us}} \right|,$$

$$V_{ub}^* = A\lambda^3(\rho + i\eta) = \frac{A\lambda^3(\bar{\rho} + i\bar{\eta})\sqrt{1 - A^2\lambda^4}}{\sqrt{1 - \lambda^2}[1 - A^2\lambda^4(\bar{\rho} + i\bar{\eta})]}, \tag{2.2}$$

where

$$\bar{\rho} + i\bar{\eta} = -(V_{ud}V_{ub}^*)/(V_{cd}V_{cb}^*), \tag{2.3}$$

and

$$\bar{\rho} = \rho(1 - \lambda^2/2 + \cdots), \ \bar{\eta} = \eta(1 - \lambda^2/2 + \cdots).$$

The Wolfenstein parametrization expands each element of the CKM matrix as a power series in the parameter λ, which is $\lambda = |V_{us}| = 0.2229 \pm 0.0022$ [22]. By neglecting terms proportional to λ^n $(n > 5)$, the Cabibbo-Kabayashi-Maskawa matrix can be written in the following way:

$$V_{\text{CKM}} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} =$$

$$= \begin{pmatrix} 1-\frac{1}{2}\lambda^2-\frac{1}{8}\lambda^4 & \lambda & A\lambda^3(\rho-i\eta) \\ -\lambda-\frac{1}{2}A^2\lambda^5[1-2(\rho+i\eta)] & 1-\frac{1}{2}\lambda^2-\frac{1}{8}\lambda^4(1+4A^2) & A\lambda^2 \\ A\lambda^3[1-(1-\frac{1}{2}\lambda^2)(\rho+i\eta)] & -A\lambda^2+\frac{1}{2}A\lambda^4[1-2(\rho+i\eta)] & 1-\frac{1}{2}A^2\lambda^4 \end{pmatrix} + \mathcal{O}(\lambda^6). \quad (2.4)$$

The elements of the CKM matrix and the Wolfenstein parameters are related the following way (up to $\mathcal{O}(\lambda^5)$):

$$\begin{gathered} V_{us}=\lambda,\ \ V_{cb}=A\lambda^2,\ \ V_{ub}=A\lambda^3(\rho-i\eta),\ \ V_{td}=A\lambda^3(1-\bar{\rho}-i\bar{\eta}), \\ \mathcal{I}mV_{cd}=-A^2\lambda^5\eta,\quad \mathcal{I}mV_{ts}=-A\lambda^4\eta. \end{gathered} \quad (2.5)$$

2.2.1 The unitarity triangles

The unitarity condition $V^{\dagger}V=I$ of the Cabibbo-Kabayashi-Maskawa matrix means that rows and columns of the matrix are mutually orthogonal:

$$\sum_{i=u,c,t} V_{ij}V^*_{ik}=\delta_{jk}\quad j,k=d,s,b \qquad \text{and} \qquad \sum_{j=d,s,b} V_{ij}V^*_{nj}=\delta_{in}\quad i,n=u,c,t$$

Each of the six conditions, which are zero, are the sum of three complex numbers and can be geometrically presented in the complex plane as a triangle.

Let us review the following three relations: $\sum_i V_{ij}V^*_{ik}=0$ for rows $i=u,c,t$, columns $j,k=d,s,b$, $j\neq k$ [25,26]:

$$\underbrace{V_{ud}V^*_{us}}_{\mathcal{O}(\lambda)}+\underbrace{V_{cd}V^*_{cs}}_{\mathcal{O}(\lambda)}+\underbrace{V_{td}V^*_{ts}}_{\mathcal{O}(\lambda^5)}=0, \quad (2.6)$$

$$\underbrace{V_{ud}V^*_{ub}}_{\mathcal{O}(\lambda^3)}+\underbrace{V_{cd}V^*_{cb}}_{\mathcal{O}(\lambda^3)}+\underbrace{V_{td}V^*_{tb}}_{\mathcal{O}(\lambda^3)}=0, \quad (2.7)$$

$$\underbrace{V_{us}V^*_{ub}}_{\mathcal{O}(\lambda^4)}+\underbrace{V_{cs}V^*_{cb}}_{\mathcal{O}(\lambda^2)}+\underbrace{V_{ts}V^*_{tb}}_{\mathcal{O}(\lambda^2)}=0, \quad (2.8)$$

and three relations $\sum_j V_{ij}V^*_{nj}=0$ for columns $j=d,s,b$, rows $i,n=u,c,t$, $i\neq n$

$$\begin{gathered} \underbrace{V_{ud}V^*_{cd}}_{\mathcal{O}(\lambda)}+\underbrace{V_{us}V^*_{cs}}_{\mathcal{O}(\lambda)}+\underbrace{V_{ub}V^*_{cb}}_{\mathcal{O}(\lambda^5)}=0, \\ \underbrace{V_{ud}V^*_{td}}_{\mathcal{O}(\lambda^3)}+\underbrace{V_{us}V^*_{ts}}_{\mathcal{O}(\lambda^3)}+\underbrace{V_{ub}V^*_{tb}}_{\mathcal{O}(\lambda^3)}=0, \\ \underbrace{V_{cd}V^*_{td}}_{\mathcal{O}(\lambda^4)}+\underbrace{V_{cs}V^*_{ts}}_{\mathcal{O}(\lambda^2)}+\underbrace{V_{cb}V^*_{tb}}_{\mathcal{O}(\lambda^2)}=0. \end{gathered} \quad (2.9)$$

Based on current estimates [22] of the absolute values of the elements $|V_{ij}|$ of the Cabibbo-Kabayashi-Maskawa matrix, it is known, that the two triangles (2.7) and (2.9) have sides of a comparable length. The rest of the triangles have one of the sides significantly shorter than the lengths of the other two.

Figure 2.1 illustrates that the two triangles a) and b), which correspond to (2.6) and (2.8), almost collapse to a line. The third triangle c) is of a special interest, i.e. the relation (2.7) with $j = d$ and $k = b$ (db–"Unitary Triangle").

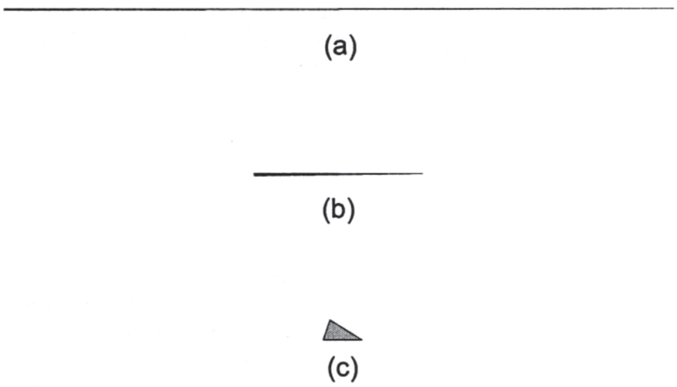


Figure 2.1: The three unitarity triangles (a) $V_{ud}V_{us}^* + V_{cd}V_{cs}^* + V_{td}V_{ts}^* = 0$ (2.6), (b) $V_{us}V_{ub}^* + V_{cs}V_{cb}^* + V_{ts}V_{tb}^* = 0$ (2.8), (c) $V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0$ (2.7) drawn to a common scale [27].

The unitary triangles (2.7) and (2.9) coincide up to $\mathcal{O}(\lambda^3)$, have corresponding interior angles and can be independently measured. Keeping $\mathcal{O}(\lambda^5)$ terms, we have [26]

$$V_{cd}V_{cb}^* = -A\lambda^3 + \mathcal{O}(\lambda^7)$$

and $V_{cd}V_{cb}^*$ is real with high accuracy, i.e. $|V_{cd}V_{cb}^*| = A\lambda^3$. Given this, the first unitary triangle can be obtained by dividing (2.7) by the well known $V_{cd}V_{cb}^*$

$$\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*} + 1 + \frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*} = 0.$$

Hence it follows, that the triangle side of the length 1 lies on the real axis. The form of the first unitary triangle is unchanged and the corners of this rescaled triangle are exactly $(0, 0)$, $(1, 0)$, $(\bar{\rho}, \bar{\eta})$, where $\bar{\rho} = \rho(1 - \lambda^2/2)$ and $\bar{\eta} = \eta(1 - \lambda^2/2)$ satisfy Eq. (2.3) (see Fig. 2.2).

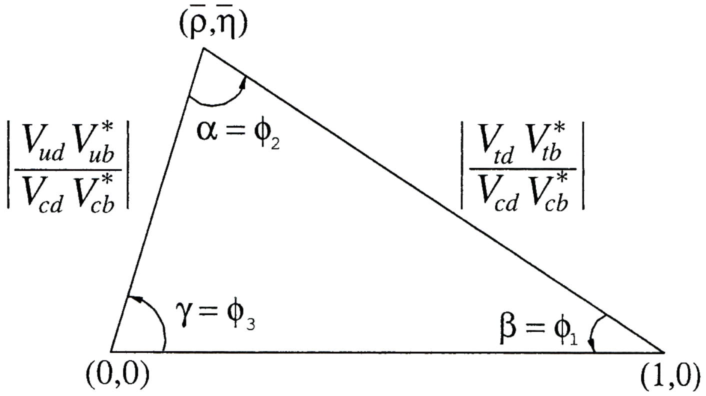

$(\bar{\rho},\bar{\eta})$
$\left|\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right|$
$\alpha = \phi_2$
$\left|\frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*}\right|$
$\gamma = \phi_3$
$\beta = \phi_1$
(0,0)
(1,0)

Figure 2.2: The rescaled unitarity triangle in the complex plane [22].

The lengths of the two sides of the triangle are

$$R_u = \frac{|V_{ud}V_{ub}^*|}{|V_{cd}V_{cb}^*|} = \sqrt{\bar{\rho}^2 + \bar{\eta}^2} = \frac{1-\lambda^2/2}{\lambda}\left|\frac{V_{ub}}{V_{cb}}\right|, \tag{2.10}$$

$$R_t = \frac{|V_{td}V_{tb}^*|}{|V_{cd}V_{cb}^*|} = \sqrt{(1-\bar{\rho})^2 + \bar{\eta}^2} = \frac{1}{\lambda}\left|\frac{V_{td}}{V_{cb}}\right|. \tag{2.11}$$

They can be depicted as circles around the vertices $(0,0)$ and $(1,0)$ in the $(\bar{\rho},\bar{\eta})$ plane, as illustrated in Fig. 2.3.

The angles of the unitarity triangle are denoted as α, β, γ. In terms of the CKM matrix, these angles are expressed in the following way

$$\begin{aligned} \beta &\equiv \arg\left(-\frac{V_{cd}V_{cb}^*}{V_{td}V_{tb}^*}\right) = \arctan\left(\frac{\bar{\eta}}{1-\bar{\rho}}\right), \\ \alpha &\equiv \arg\left(-\frac{V_{td}V_{tb}^*}{V_{ud}V_{ub}^*}\right) = \arctan\left(\frac{\bar{\eta}}{\bar{\eta}^2 + \bar{\rho}(\bar{\rho}-1)}\right), \\ \gamma &\equiv \arg\left(-\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right) = \arctan\left(\frac{\bar{\eta}}{\bar{\rho}}\right), \end{aligned} \tag{2.12}$$

where

$$\bar{\rho} = \rho\left(1-\frac{\lambda^2}{2}\right), \qquad \bar{\eta} = \eta\left(1-\frac{\lambda^2}{2}\right),$$

and the sum of the unitarity triangle angles must be equal to $\alpha + \beta + \gamma = \pi$.

The unitarity triangle angles can be extracted from special decay channels (see Table 2.4). For example, the value of $\sin 2\beta$ has been measured by BABAR [28] and BELLE [29] in the decay channel $B^0 \to J/\psi K_s^0$. In this case the asymmetry $a_{CP}(t)$ [30] is

$$a_{CP}(t) = \frac{\Gamma_{B^0\to J/\psi K_s^0}(t) - \Gamma_{\bar{B}^0\to J/\psi K_s^0}(t)}{\Gamma_{B^0\to J/\psi K_s^0}(t) + \Gamma_{\bar{B}^0\to J/\psi K_s^0}(t)} = -\sin(2\beta)\sin(\Delta m_d t),$$

where the Γs are the decay widths. The Δm_d, which determines the oscillation frequency between the B^0 and $\bar{B}^0$ mesons, is their mass difference. Thus, with known Δm_d the measurement of the asymmetry $a_{CP}(t)$ gives $\sin 2\beta$, where β is the angle of the unitarity triangle. The world average is $\sin 2\beta = 0.673 \pm 0.023$ [22].

All unitarity triangles have the same area that is equal to $J/2$, where $J = \mathcal{I}m[V_{ud}V_{cs}V_{us}^*V_{cd}^*]$ is the Jarlskog invariant, which can also be expressed via other combinations of the matrix elements $V_{\rm CKM}$ (see, e.g., [31]).

In the Wolfenstein parameterization $J \simeq \lambda^6 A^2 \eta$. The small quantity $|J| \sim \mathcal{O}(\lambda^6)$ sets the scale of *CP*-violation in the Standard Model. J can be viewed as the *CP* violation parameter, as if *CP* is violated, J is different from zero.

The precise determination of the CKM matrix elements is important because they are a part of the fundamental parameters of the Standard Model, which describe the weak interactions. Therefore, to be able to verify the predictions of the Standard Model, it is necessary to measure the angles and sides of the unitary triangles with a high accuracy. A new effect could be observed in measurements of the *CP* violating decays of *B*-mesons, and it is these studies that are the main goal of the LHCb experiment.

2.3 CP violation, New Physics and LHCb

It is now believed that the Standard Model, being the effective theory in the region of low energies, is a part of a more general, fundamental theory at higher energies. The LHC experiments aim at verifying the Standard Model predictions and at searching for New Physics (new particles, new symmetries) in the TeV region. New Physics could explain the matter-antimatter asymmetry of the universe. The LHCb experiment is designed specially for precise measurements of CP violation and research of rare B decays, and to test whether the CKM model is a suitable way to describe the experimental results.

Figure 2.3 illustrates the great progress, made in the past years, in the measurements of the parameters of the CKM matrix.

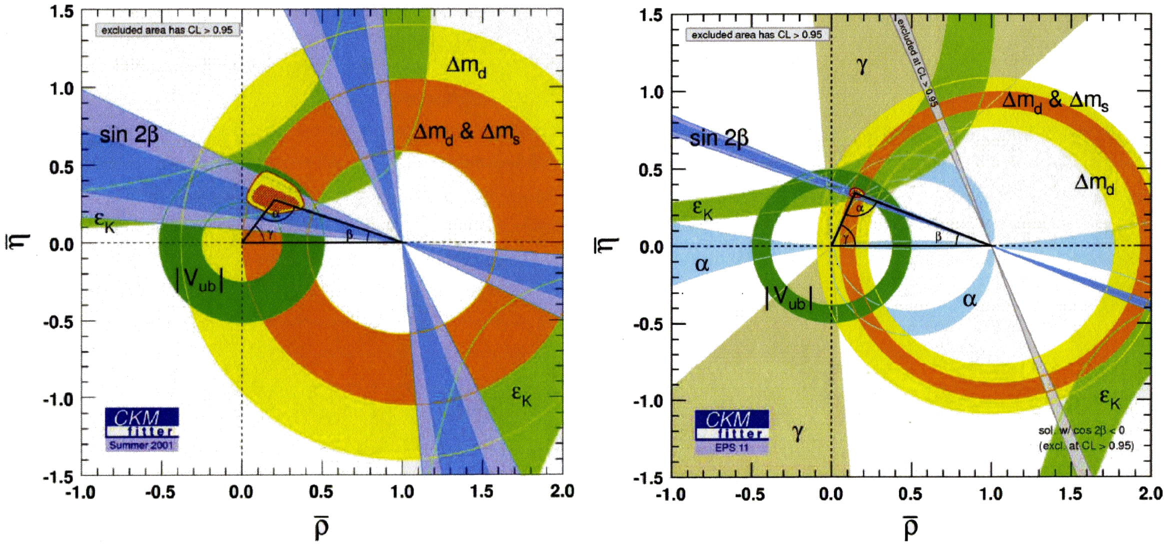


Figure 2.3: EPS 2001 on the left vs. EPS 2011 on the right [22].

The different colored bands show the experimental accuracy of different experiments and overlap in a small range with $\bar{\rho}, \bar{\eta} = (0.14, 0.34)$. A non-zero value of η can only be obtained if CP violation exists. It is obvious that this small overlap is far from zero ($\bar{\eta} \neq 0$), i.e. CP-violation is an experimentally proved fact.

The angles of the unitarity triangle can be extracted in the following ways:

- indirectly by measuring the lengths of the unitarity triangle sides;
- directly from the CP asymmetries in different final states of B-meson decays.

The theory and methods for measuring the angles of the unitarity triangles are well described in the literature [26, 27, 30]. Tab. 2.4 illustrates the different decays, from which the angles or their combinations could be defined at the LHCb experiment.

Here, the strategy is to perform a large number of measurements, results of which determine the values and phases of the elements of the CKM matrix. As was discussed previously in Sec. 2.2, in the case of Wolfenstein parametrization the elements of the CKM matrix are described by the set of parameters $\lambda, A, \bar{\rho}, \bar{\eta}$ and the unitarity triangle gives a very clear geometrical representation of the structure of the Cabibbo-Kobayashi-Maskawa matrix. Thus, in the $(\bar{\rho}, \bar{\eta})$ plane, the different curves, obtained from different experiments (and various decays), should intersect in a single point (Fig. 2.3), and any inconsistency of the obtained results can be considered as a sign for New Physics.

	Parameter	Decay Mode
1	$\beta+\gamma$ $(=\pi-\alpha)$	from B_d^0 and $\bar{B}_d^0 \to \pi^+\pi^-$
2	β	from B_d^0 and $\bar{B}_d^0 \to J/\psi K_s$
3	γ	from $B_d^0 \to \bar{D}^0 K^{*0}$, $D^0 K^{*0}$, $D_1 K^{*0}$ and $\bar{B}_d^0 \to \bar{D}^0 \bar{K}^{*0}$, $D^0 \bar{K}^{*0}$, $D_1 \bar{K}^{*0}$

Table 2.4: Decay Modes for measuring different CKM angles

Appearance of New Physics in the B-meson system can occur in the following ways [27]:

- the values of quantity in the SM, obtained from two different measurements, give incompatible results;
- the sum of the extracted angles is equal to $\alpha+\beta+\gamma=\pi$, but they are inconsistent with the measured triangle sides;
- the asymmetries are larger than predicted by the SM;
- the expected rare decays have enhanced rates or suppressed;
- mixing in B- or D-meson decays differ significantly from SM predictions.

Chapter 3

QCD

"Information gained in experiments should be revised into theoretical conclusions. ... Ideally, this is just one formula posing the whole tree of physics. But this ideal is still far away."

– Lev Okun. "Physics of elementary particles". Moscow: Nauka, 1988

This chapter presents a brief summary on Quantum Chromodynamics. A short introduction is followed by Section 3.1 which explains the quark structure of hadrons, and in Section 3.2 experiments and the quark parton model are described. Section 3.3 introduces the concept of color, and sections 3.4 and 3.5 are dedicated to the QCD Lagrangian and the running coupling, respectively. Section 3.6 describes interactions and processes which take place in a proton-proton scattering, and

Quantum chromodynamics (QCD) is a theory of interactions of quarks and gluons – the main components of strong interacting particles (hadrons).

Prior to Chadwick's discovery of the neutron in 1932, the proton was the only strongly interacting particle. Heisenberg and Ivanenko proposed the idea that atomic nuclei consist of protons and neutrons, but the theory of electromagnetic and gravitational forces could not explain the fact that protons and neutrons are held inside atomic nuclei. In 1935, Yukawa proposed that protons and neutrons interact through the exchange of an unknown strongly interacting particle - a pion. The nuclear (nucleon-nucleon) forces are much stronger than electromagnetic and gravitational forces, and are short-ranged. After that, pions were discovered in cosmic rays and in the processes of strong interactions of hadrons and nuclei at high energies where they are produced in large quantities.

In the 50's began an intensive theoretical work on the processes of multiple production of particles at high energies due to strong interactions. QCD emerged in the early 70's as a result of the synthesis of non-Abelian gauge fields theory (Yang, Mills 1954), parton model of deep inelastic interactions (Gell-Mann, Zweig, 1964), and the idea of the color of quarks.

Currently, the main theory for the description of strong interactions is quantum chromodynamics. In QCD strong interactions of hadrons are treated as color interactions between quarks and gluons. A detailed description of experimental results usually requires phenomenological assumptions in addition to the pure theory, in order to account for non-perturbative effects. Numerous analytical, computational, Monte Carlo simulations (which were based on ideas of QCD) were made to describe the experimental data of e^+e^-, ep, $pp(p\bar{p})$ interactions at high energies, and they proved very successful.

3.1 Quark structure of hadrons

Hadrons are particles that participate in strong interactions. The introduction of the quark model allowed "to restore order" in this large family of particles (there are about 450 of them). There are two types of hadrons: baryons and mesons.

The quark model was proposed by Gell-Mann and Zweig in 1964. The sensationalism of the idea lies in fractional electric charge and baryon charge attributed to the quarks. Currently there are six known quarks, which are strongly interacting fermions with spin 1/2 and, by convention, have positive parity. Antiquarks have negative parity. Table 3.1 gives the additive quantum numbers for the three generations of quarks.

Property Quark	d	u	s	c	b	t
B – baryon number	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
J – spin	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Q – electric charge	$-\frac{1}{3}e$	$+\frac{2}{3}e$	$-\frac{1}{3}e$	$+\frac{2}{3}e$	$-\frac{1}{3}e$	$+\frac{2}{3}e$
I – isospin	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
I_3 – isospin z-component	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
s – strangeness	0	0	-1	0	0	0
c – charm	0	0	0	$+1$	0	0
b – bottom	0	0	0	0	-1	0
t – top	0	0	0	0	0	$+1$

Table 3.1: Additive quantum numbers of the quarks (for all quarks $J^p = 1/2^+$ and baryon number $B = 1/3$)

The existence of quarks was demonstrated experimentally, but they were never detected in a free state. For all the quarks, their baryon number (or charge) B is equal to 1/3. Each of the six quarks has its own flavor, which is identical to its name u, d, s, c, b, t. The flavors for s, c, b, t are set by special additive numbers: *strangeness* (s) of s-quarks, *charm* (c) of c-quarks, quantum numbers *bottom* or *beauty* (b) and *top* or *truth* (t) of b- and t-quarks respectively. These quantum numbers are presented in Table 3.1 (it was agreed to consider the signs of flavors to coincide with the signs of the electric charges of the quarks).

Thus, for any hadron these quantum numbers are determined from its quark structure. Quantum numbers s, c, b, t divide hadrons into light hadrons ($p, n, \pi, ...$), strange particles ($K, \Lambda, \Sigma, ...$), charm ($D, \Lambda_c, \Sigma_c, ...$) and bottom- or beauty-particles ($B, \Lambda_b, \Xi_b, ...$). The t-quark, having a life time of $\tau \approx 10^{-25}$s, decays before forming a bound state. In the transition from a particle to an antiparticle signs of all additive quantum numbers (which mean different types of charges) are reversed. As it was already mentioned, the quark baryon charge is $+1/3$ ($-1/3$ in case of antiquarks), it is also an additive and a conserved quantum number. Hadrons consisting of three quarks have baryon charge $B = 1/3 + 1/3 + 1/3 = +1$ and are called baryons ($B = -1$ for antibaryons). Mesons and antimesons, which consist of a pair of a quark and an antiquark, have $B = (+1/3) + (-1/3) = 0$.

The Gell-Mann-Nishijima formula links the hadron's electrical charge Q (in units of the elementary charge e), its third projection of isospin I_3, and baryon charge B

$$Q = I_3 + \frac{Y}{2} \tag{3.1}$$

where $Y = B + s$ is hypercharge (or $Y = B + s + c + b + t$ called generalized hypercharge).

In order to identify the "aromatic" properties of the lightest quarks u and d, the isospin quantum number (I) is used, which is a characteristic of the *isospin symmetry* of strong interaction. Isospin space is a three-dimensional space. u and d quarks have identical isospin I, but different isospin projections I_3 on the 3-axis of the isospin space (the names of the quarks *up* and *down* specify the isospin direction). Isospin plays an important role in the classification of hadrons. The particles which have the same I and different projections I_3, but otherwise the same quantum numbers, form *isospin multiplets.* The number of particles n in an isospin multiplet is determined by the number of projections I_3, i.e. $n = 2I + 1$.

In this manner, proton and neutron, which consist of u and d quarks, have isospin $I = 1/2$, but differ by the sign of their I_3 projections, and form an isospin doublet:

$$p \equiv uud \;\; (I = 1/2, I_3 = +1/2),$$

$$n \equiv udd \;\; (I = 1/2, I_3 = -1/2),$$

i.e. they can be considered as two different states of a single particle – the nucleon.

π-mesons form an isospin triplet with $I = 1$:

$$\begin{aligned} \pi^+ &\equiv u\bar{d} \;\; (I = 1, I_3 = +1), \\ \pi^0 &\equiv u\bar{u} \text{ or } d\bar{d} \;\; (I = 1, I_3 = 0), \\ \pi^- &\equiv \bar{u}d \;\; (I = 1, I_3 = -1). \end{aligned}$$

Grouping of hadrons into isospin multiplets reflects the symmetry of the strong interactions, relative to the transformations (rotations) in isospin space.

Experience has shown that hadrons form wider unions of particles than isospin multiplets. Group theory, which is the basis for the description of symmetries, is used to study the symmetry of hadrons with respect to the replacement of their quarks with different ones.

As a simple example, let us consider hadrons which consist of the three lightest quarks u, d, s. The task is to study the symmetry of hadrons which include only three "building blocks". Such symmetries can be, e.g., the so called $SU(3)$ symmetries. The $SU(3)$ group (Special Unitary Group: $\det U = 1$, $UU^\dagger = I$) is formed by transformations which convert states into each other, constructed of three elements. Hadrons are grouped into multiplets on the base of the symmetry of flavors u, d, s, which are usually called $SU(3)$-multiplets. This symmetry implies the equality of the masses of u, d and s quarks and equality in pair interactions, such as uu, ud, us, dd, ss of quarks and antiquarks. Let us see which quark-antiquark (meson) and three-quark (baryon) combinations are possible. Mesons consist of $q_i\bar{q}_j$, where $i, j = u, d, s$ and in total $3^2 = 9$ combinations of the three quarks and antiquarks are possible, which are presented in Table 3.2.

	$\bar{u}$	$\bar{d}$	$\bar{s}$
u	$u\bar{u}$	$u\bar{d}$	$u\bar{s}$
d	$d\bar{u}$	$d\bar{d}$	$d\bar{s}$
s	$s\bar{u}$	$s\bar{d}$	$s\bar{s}$

Table 3.2: Quark-antiquark pair combinations.

This leads to nine "flavor" combinations which correspond to nine types of mesons. Meson groups, united by the same spin value J, parity p and are located in the same mass region, are examples of $SU(3)$ multiplets. The lightest meson group splits into two nonets: with $J^p = 0^-$ (pseudoscalar mesons) and with $J^p = 1^-$ (vector mesons). Considering that for mesons $B = 0$, from the Gell-Mann-Nishijima (3.1) formula it follows that out of Q, I_3 and s (or Y) only two quantities remain independent and all the mesons of the nonet can be obtained from any other of these states by changing only the I_3 and s (or Y) values. Therefore, it is convenient to represent these multiplets on the plane with I_3 and s (or Y) axis.

The quark structure of the lightest meson/antimesons nonet with $J^p = 0^-$ is illustrated in Fig. 3.1:

$$\pi^+, \pi^0, \pi^-, \eta, \eta', K^+, K^0, K^-, \bar{K}^0.$$

All meson multiplets contain both particles and antiparticles. Particles and antiparticles in Fig. 3.1 are arranged symmetrically about the center of the figure ($c = I_3 = 0$). For example, antiparticle $\bar{K}^0 = \bar{d}s$ is obtained from particle $K^0 = d\bar{s}$ by replacing its quarks with corresponding antiquarks: $d \to \bar{d}$, $\bar{s} \to s$.

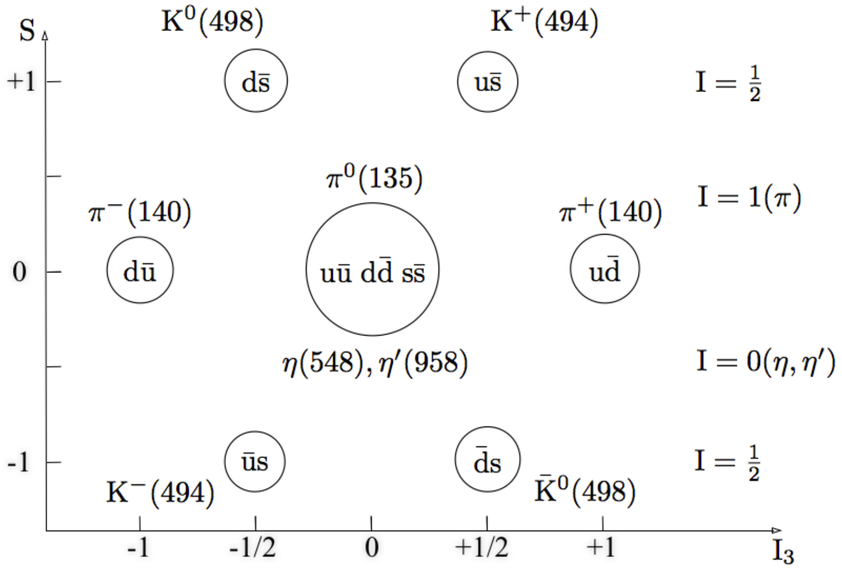


Figure 3.1: Quark structure of the lightest meson nonet.

The nonet can be split into two isodoublets ($I = 1/2$): K^0, K^+, and K^-, $\bar{K}^0$, one isotriplet ($I = 1$): π^-, π^0, π^+ and two isosinglets ($I = 0$): η and η'. There are three particles at the heart of the figure ($S = 0$ and $I_3 = 0$), since out of u, d, s quarks and their antiquarks it is possible to combine three pairs $u\bar{u}, d\bar{d}, s\bar{s}$ with $I_3 = 0$.

The quark structure of a baryon can be written as $q_i q_j q_k$, where the subscript corresponds to the quark's flavor $i, j, k = u, d, s$. All the combinations form several $SU(3)$ multiplets, which can be classified according to the symmetry under permutations of the quarks. Thus all the baryons which are composed of the three u, d, s quarks are grouped into a single decuplet, two octets, and one singlet. The octets differ by spin and parity of particles which form these octets. For example, the baryon octet with spin $J^p = 1/2^+$

$$p, n, \Lambda, \Sigma^+, \Sigma^0, \Sigma^-, \Xi^0, \Xi^-$$

is located in the mass region of 938 – 1321 MeV. Another octet groups baryons with $J^p = 3/2^-$ and their masses lie in the region of 1520 – 1820 MeV. Characteristics of the baryon octet with spin $J^p = 1/2^+$, together with their quark composition, are presented in Table 3.3.

Baryon	Quark composition	mc^2 [MeV]	$s(Y)$	I_3	I
n	uud	938	$0(+1)$	$+1/2$	$1/2$
p	udd	940	$0(+1)$	$-1/2$	$1/2$
Σ^+	uus	1189	$-1(0)$	$+1$	1
Σ^0	uds	1193	$-1(0)$	0	1
Σ^-	dds	1197	$-1(0)$	-1	1
Λ	uds	1116	$-1(0)$	0	0
Ξ^0	uss	1315	$-2(-1)$	$+1/2$	$1/2$
Ξ^-	dss	1321	$-2(-1)$	$-1/2$	$1/2$

Table 3.3: The lightest baryons octet ($J^p = 1/2^+$)

Table 3.3 also presents strangeness s, hypercharge $Y = B + s$, isospin I and isospin projection I_3 of the baryons. The isospin values of baryons are obtained by vector addition of isospins of quarks, and the isospin projection is calculated by algebraic addition. Baryons have strangeness which is caused by the presence of s quarks in their structure. Strangeness is determined as an algebraic sum of strangenesses of quarks which form the baryon.

The quark structure of the light baryon octet (with spin $J^p = 1/2^+$) is illustrated in Fig. 3.2:

It can be seen that the particles of the octet, lying on the horizontals, form the following isospin multiplets: one isosinglet Λ ($I = 0$), two isodoublets n, p and Ξ^0, Ξ^- ($I = 1/2$), and one isotriplet $\Sigma^-, \Sigma^0, \Sigma^+$ ($I = 1$). Inside the isomultiplets, particles differ only by their isospin projection and have similar properties with respect to the strong interaction. Their difference in masses, which reflects a high degree of isospin symmetry in strong interactions, is only on the percent level and is electromagnetic of nature. On the other hand, the difference in the masses of particles of different isomultiplets (vertically) is significant (6 – 40%).

The baryon octet ($J^p = 1/2^+$) is formed of three light quarks u, d, s and out of these seven different combinations one (uds) corresponds to two different particles – Σ^0 and Λ. The difference is that Σ^0 has isospin $I = 1$ and its projection $I_3 = 0$, i.e. this particle is a part of the isotriplet $\Sigma^+, \Sigma^0, \Sigma^-$. Whereas Λ is an isosinglet, i.e. a particle with $I = 0$ and $I_3 = 0$.

The baryon decuplet with $J^p = 3/2^+$ is formed in a similar way and is composed of the following particles:

$$\Delta^{++}, \Delta^+, \Delta^0, \Delta^-, \Sigma^{+*}, \Sigma^{0*}, \Sigma^{-*}, \Xi^{0*}, \Xi^{-*}, \Omega^-.$$

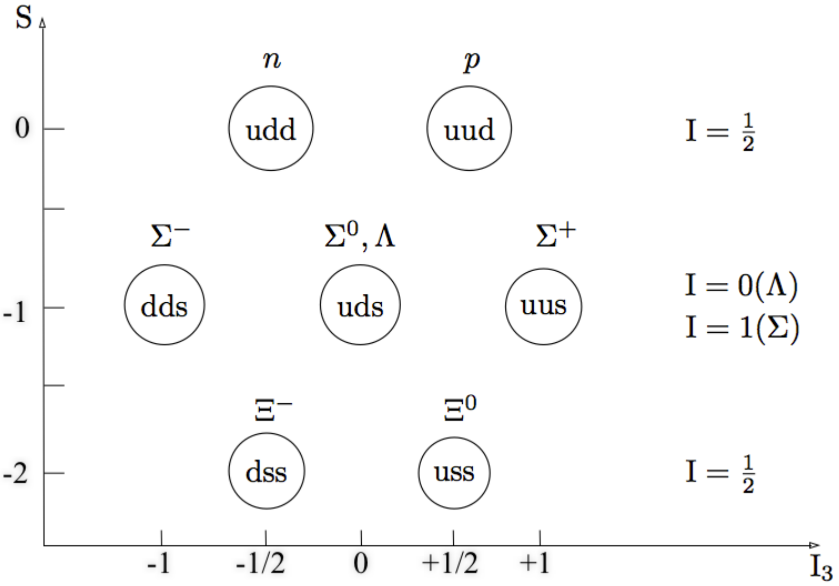
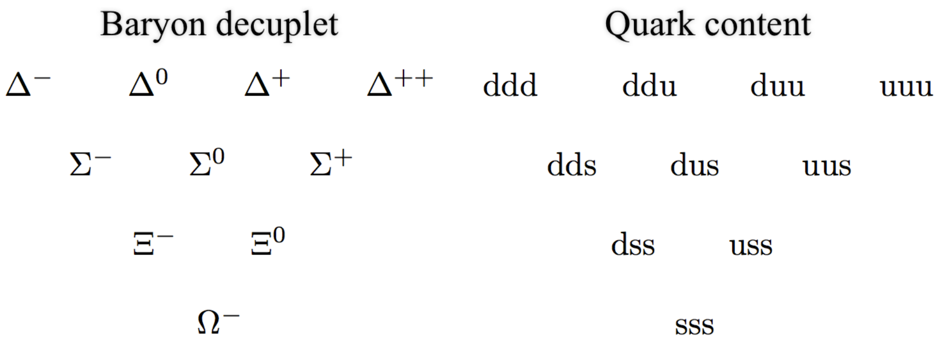


Figure 3.2: Quark structure of the lightest baryon octet.

The decuplet is presented in Fig. 3.3. In comparison to the baryon octet $J^P = 1/2^+$, the baryon decuplet $J^P = 3/2^+$ has member particles which are composed of quarks of only one flavor: ddd, uuu, sss.



Baryon decuplet				Quark content			
Δ^-	Δ^0	Δ^+	Δ^{++}	ddd	ddu	duu	uuu
Σ^-	Σ^0	Σ^+		dds	dus	uus	
Ξ^-	Ξ^0			dss	uss		
Ω^-				sss			

Figure 3.3: Baryon decuplet.

Experimental studies of the baryons of the decuplet show that the resulting orbital angular momentum of quarks of these particles is $L = 0$ and their spins are parallel ($\uparrow\uparrow\uparrow$). Thus the total momentum of the particles of the decuplet is $J = 3/2$, and parity of the particles is $P = +1$. Thus we obtain for the spin and parity of the particles of the decuplet $J^P = 3/2^+$.

The top row of the Fig. 3.3 is formed by the particles with strangeness $s = 0$. These particles are the four possible triple combinations of the two quarks u and d. They differ by their isospin projection I_3 and form the isospin quartet ($I = 3/2$). The next row ($s = -1$) consists of the three combinations of the two quarks, either u or d, and a single s quark, which form the isospin triplet ($I = 1$). The particles of the next row have two s-quarks ($s = -2$) in their composition: dss and uss, and form the isodublet ($I = 1/2$). The Ω^- particle of the bottom row consists of three s-quarks and forms the isosinglet with isospin $I = 0$.

Comparison of Fig. 3.3 and Fig. 3.2 shows that there are particles in the decuplet, which fully consist of quarks of the same flavor: ddd, uuu, sss (this is explained in Section 3.4). The rest of the quark combinations repeat combinations of the octet $J^p = 1/2^+$, but in the decuplet they have $J^p = 3/2^+$.

Figure 3.3 clearly demonstrates the remarkable symmetry. It is seen that in the horizontal direction from left to right, d quarks are replaced by u quarks (I spin symmetry). In the vertical direction, from top to bottom, along the left edge d quarks are substituted by s quarks and along the right edge the situation is vice versa - u quarks are exchanged for s quarks (the so-called U spin and V spin symmetry). Consequently, we can construct the known particles in a systematic way.

3.2 Experiments and the quark parton model

Classic experiments on the structure of the nucleon have been performed in 1964 at the Stanford National Laboratory (SLAC, USA). Experiments on the elastic scattering of electrons on protons $e+p \to e+p$ showed that protons and neutrons are not point-like objects but have an extended structure. Further study of the internal structure of the proton was made in experiments on deep inelastic scattering of electrons on protons. The difference between elastic and inelastic collisions is that there are no new particles produced in elastic scattering and the energy of the electron after scattering remains the same (only the direction of emission of the electron changes). Energy carried away by new particles can be determined from the electron energy after interaction, E'.

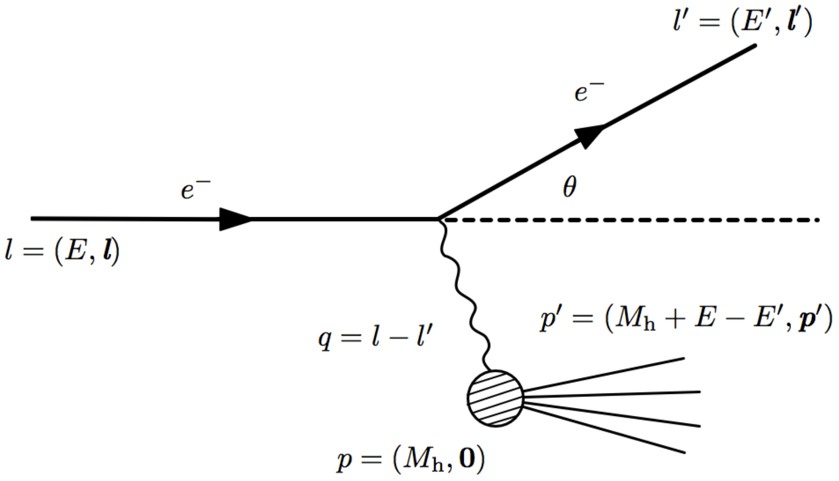

$l' = (E', \boldsymbol{l}')$
e^-
e^-
θ
$l = (E, \boldsymbol{l})$
$p' = (M_h + E - E', \boldsymbol{p}')$
$q = l - l'$
$p = (M_h, \boldsymbol{0})$

Figure 3.4: Kinematics of deep inelastic electron-nucleon scattering.

In Fig. 3.4 a high energy electron with initial energy E is scattered off a nucleon, which is at rest in the lab-system. The l and l' are the four-momenta of the incoming and outgoing electron and θ is the scattering angle of the electron. The electron transfers four-momentum $q = l - l'$

to the nucleon, and at large values of q^2 multiple hadron production takes place. We shall not concretize the resulting hadronic system, which makes this an inclusive interaction, i.e. the result of the interactions includes everything possible. Here p is the four-momentum of the nucleon with mass M_h, p' is the four-momentum of the hadronic system with the invariant mass W.

The process depicted in Fig. 3.4 is called deep ($Q^2 \gg M_h^2$) inelastic ($W^2 \gg M_h^2$) scattering (DIS). The mass of the initial and scattered electron can be *neglected*, and the nucleon filled circle represents the internal structure of the nucleon that can be expressed in terms of structure functions.

Usually in such an experiment one measures the angle of deflection of the electron θ, energy E' and momentum l' of the electron after its interaction with the nucleon. These data are sufficient to calculate the degree of excitation of the nucleon after collision with the electron, using the laws of conservation of energy and momentum, and the following formulas:

$$\nu = E - E' = \frac{q \cdot p}{M_h}$$

which describes the energy transfer from the electron to the hadronic system in the hadron's rest frame,

$$Q^2 = -q^2 = -(l - l')^2 = -(p' - p)^2 = 2p \cdot p' - p^2 - p'^2$$

which defines the squared momentum transfer Q^2 carried by the virtual photon,

$$W^2 = (p + q)^2$$

which corresponds to the squared mass of the hadronic system recoiling against the scattered electron.

From here, it follows that

$$Q^2 = 2M_h(E + M_h - E') - M_h^2 - W^2 = 2M_h\nu + M_h^2 - W^2 \leq 2M_h\nu,$$

where in the case of elastic scattering $M_h^2 = W^2$.

The Bjorken-variable x_B ($0 \leq x_B \leq 1$) is an experimental observable, which reflects the deviation from elastic scattering

$$x_B = \frac{Q^2}{2M_h\nu}. \tag{3.2}$$

For further understanding, one would need the fact that the double differential cross section with respect to Q^2, ν is derived from the cross section for elastic scattering of an electron with a spin-1/2 fermion of mass M_h and charge e_q and is equal to [32]

$$\frac{d^2\sigma}{dQ^2 d\nu} = \frac{4\pi\alpha_{\text{em}}^2 e_q^2}{Q^4}\frac{E'}{E}\left\{\cos^2\frac{\theta}{2} + \frac{Q^2}{2M_h^2}\sin^2\frac{\theta}{2}\right\}\delta\left(\nu - \frac{Q^2}{2M_h}\right). \tag{3.3}$$

In the case when the nucleon is considered to have some internal structure, one can assume that in such case the double differential cross section will be similar to (3.3):

$$\frac{d^2\sigma}{dQ^2 d\nu} = \frac{4\pi\alpha_{\text{em}}^2 e_q^2}{Q^4}\frac{E'}{E}\left\{W_2(Q^2,\nu)\cos^2\frac{\theta}{2} + 2W_1(Q^2,\nu)\sin^2\frac{\theta}{2}\right\}, \tag{3.4}$$

introducing structure functions $W_{1,2}(Q^2, \nu)$ which define the structure of the nucleon.

Hence, by comparing (3.3) and (3.4), the structure functions for the elastic scattering of pointlike particles can be written as:

$$W_1^{\mathrm{el}}(Q^2,\nu) = e_q^2 \frac{Q^2}{4M_h^2}\delta\left(\nu - \frac{Q^2}{2M_h}\right), \qquad W_2^{\mathrm{el}}(Q^2,\nu) = e_q^2 \delta\left(\nu - \frac{Q^2}{2M_h}\right). \tag{3.5}$$

Figure 3.5 illustrates the double differential cross section, depending on q^2 and $W = E'$ for an electron-proton scattering.

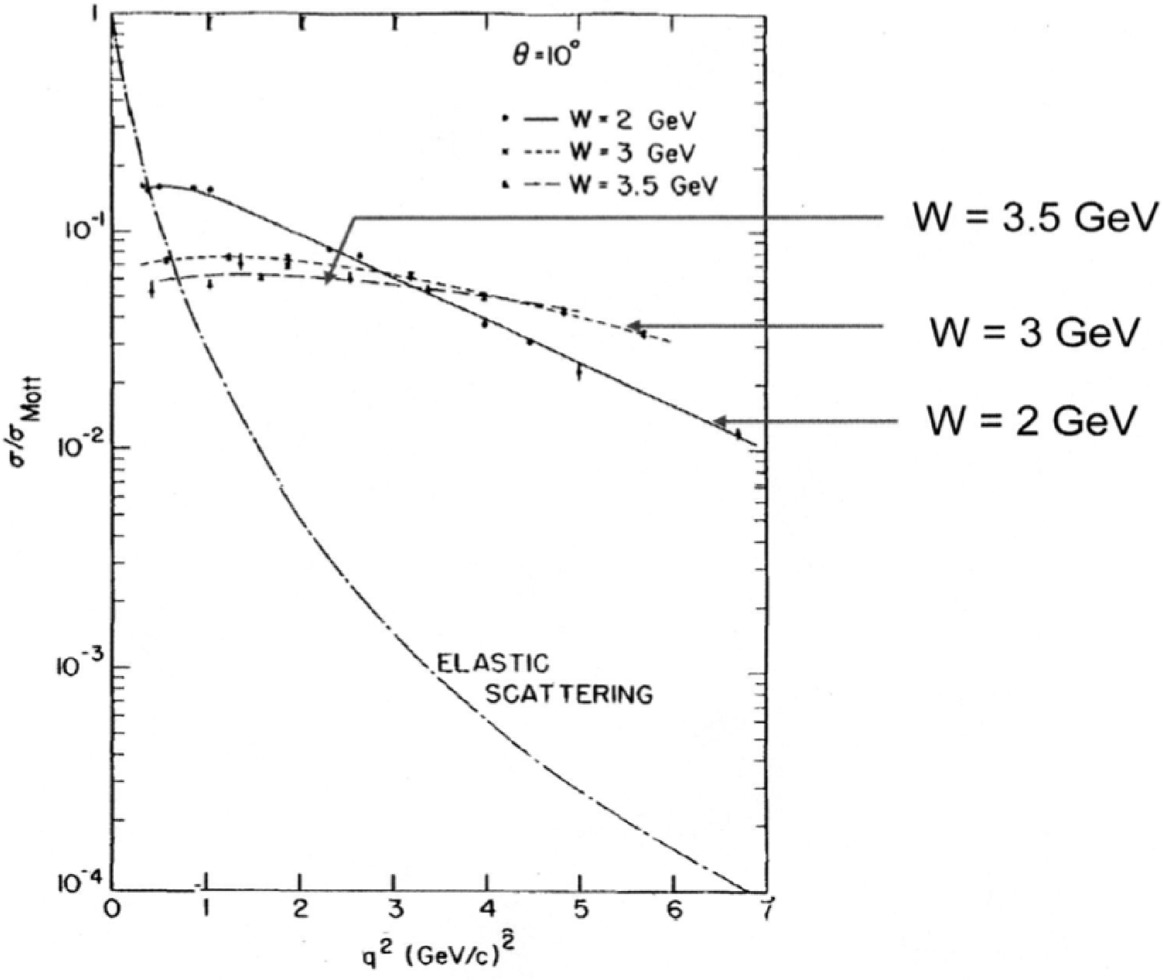


Figure 3.5: Double differential cross section $\sigma/\sigma_{\mathrm{Mott}}$ of an inelastic process, depending on q^2 and different energies $W = E'$ [33].

It is seen, that with the increase of W from 2 GeV to 3.5 GeV the dependence of the cross section on q^2 is weak in comparison to elastic scattering. This indicates that the electron is interacting with the pointlike objects contained inside the proton. Such analysis proved the existence of pointlike entities inside protons, which were called *partons*. Further study of the properties of partons (spin of which was found to be equal to $J = 1/2$) identified them as quarks, which were introduced by Gell-Mann and Zweig in order to explain the structure of hadrons.

In 1969, Feynman proposed the *parton model* , in which an extended nucleon consists of loosely coupled point particles (partons), on which the high-energy electron scatter incoherently, i.e. the electron elastically scatters on one of these particles, without affecting the others. Every nucleon participates in an inelastic interaction by a single parton i, which carries a fraction x_i of the nucleon's four-momentum p, with a probability density $f_i(x_i)$. The function $f_i(x)$ is called *the parton distribution function* (PDF), and $f_i(x)dx$ defines the probability for x_i to fall into the interval $[x, x+dx]$. Hence one can write down the structure functions W_1, W_2 as a sum of the elastic structure functions in (3.5) with weight f_i.

In the case when an electron scatters on a parton with momentum $x_i p$ and after the interaction the parton has the same mass as in the beginning of the reaction, we have $(x_i p)^2 = (x_i p + q)^2$, thus $Q^2 = 2x_i M_h \nu$. By replacing M_h by $x_i M_h$, summing and integrating one has the following

$$W_1(Q^2, \nu) = \sum_i \int_0^1 dx_i f_i(x_i) e_i^2 \frac{Q^2}{4x_i^2 M_h^2} \delta \left(\nu - \frac{Q^2}{2x_i M_h} \right) = \sum_i e_i^2 f_i(x_B) \frac{1}{2M_h}, \tag{3.6}$$

$$W_2(Q^2, \nu) = \sum_i \int_0^1 dx_i f_i(x_i) e_i^2 \delta \left(\nu - \frac{Q^2}{2x_i M_h} \right) = \sum_i e_i^2 f_i(x_B) \frac{x_B}{\nu}, \tag{3.7}$$

where the Bjorken-variable x_B is from (3.2), i.e. x_B can be considered as the four-momentum fraction x carried by the parton.

Thus we can redefine the structure functions for deep inelastic scattering of a charged unpolarized lepton on an unpolarized nucleon, in the form of

$$F_1(x) = M_h W_1 = \frac{1}{2} \sum_i e_i^2 f_i(x) \tag{3.8}$$

$$F_2(x) = \nu W_2 = \sum_i e_i^2 x f_i(x). \tag{3.9}$$

It is seen that $F_1(x, Q^2) = F_1(x)$ (the parton density), $F_2(x, Q^2) = F_2(x)$ (the momentum density) and the cross section depends only on x.

The independence of the dimensionless functions $F_1(x)$ and $F_2(x)$ of Q^2 means they are independent of the momentum transfer (scale invariance).

This phenomenon of scaling invariance of structure functions is called *Bjorken scaling.* In the base of Bjorken scaling lies the statement that the structure functions F_i *scale*, i.e. $F_i(x, Q^2) \to F_i(x)$ in the Bjorken limit, Q^2 and $\nu \to \infty$ with x fixed and F_i are functions of x only. Dimensionless structure functions are independent of any mass scale.

A demonstration of the fact that $F_2(Q^2)$ doesn't depend on Q^2 for fixed $x_B = 0.25$ in Fig. 3.6 is the experimental observation of scaling and confirmation of the existence of partons.

Above we assumed that partons are spin-1/2 particles. From (3.8) and (3.9) automatically follows the following relation for F_1 and F_2 (the *Callan-Gross relation*)

$$2xF_1(x) = F_2(x),$$

which is confirmed experimentally and proves that partons are spin-1/2 fermions.

Bjorken scaling claims that the structure of, for example, a proton is the same from the point of an electromagnetic probe (electron), regardless of how hard it hits the proton.

Figure 3.7 presents the proton structure functions F_2, in dependence of q^2, for different values of x.

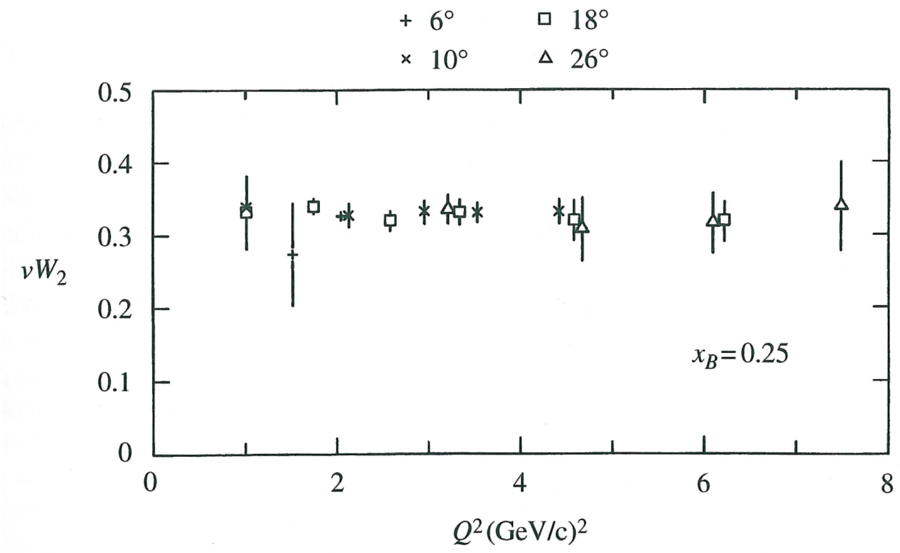


Figure 3.6: Value of $\nu W_2(Q^2) = F_2(Q^2)$ for fixed $x_B = 0.25$ (MIT-SLAC Collab, 1970).

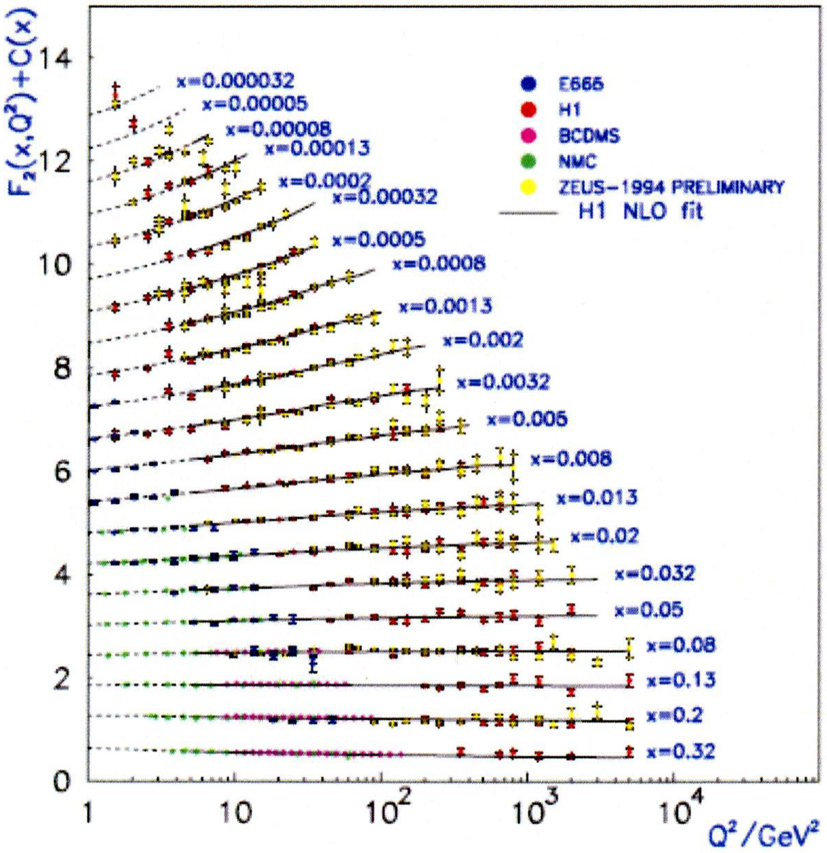


Figure 3.7: Proton structure function F_2^p measured in electromagnetic scattering of electorns and muons for $x > 0.00003$, $Q^2 = -q^2$. It is seen that for the values of x from 0.008 to 0.35, F_2 have only a weak dependence on Q^2 [22].

No dependence of the distributions of q^2 for $x > 0.02$ indicates the point-like nature of objects on which scattering occurs. The structure functions for partons inside nucleons (F_1, F_2) were estimated in numerous experiments on deep inelastic scattering of leptons on nucleons ($ep, \mu p, \nu p$). It is now believed that baryons consist of three quarks and mesons consist of a quark and an antiquark. The fine structure of hadrons appears only at very high momentum transfer at distances $r \sim \hbar/q < 10^{-14}$ cm. For small momentum transfer q, the lepton interacts not with an individual quark, but with a set of quark-antiquark pairs and gluons, which surround the pointlike quark, which is called valence quark. The quark-antiquark pairs which surround a valence quark are called sea quarks.

As a result of detailed measurements and comparison experiments on the scattering of electrons on various hadron targets, it was possible to separate the contribution of different sorts of partons and to determine their electric charges. It turned out that they coincide with the charges of quarks, namely 2/3 and −1/3.

The identification of charged partons with quarks leads to the quark-parton model. In analogy with partons (3.8), (3.9), the distribution functions of quarks and anti-quarks were introduced to describe their distribution in the nucleon: $f_q(x) \equiv q(x)$, $f_{\bar{q}}(x) \equiv \bar{q}(x)$ (where q and $\bar{q}$ correspond to a quark and an anti-quark respectively). In this notation, the expression (3.9) for the structur function $F_2(x)$ takes the following form:

$$F_2(x) = x \sum_q e_q^2 (q(x) + \bar{q}(x)), \tag{3.10}$$

where e_q^2 is the squared quark charge of type q.

The study of electron-nucleon scattering and neutrino-nucleon scattering made it possible to determine the fraction of the four-momentum of the quarks and anti-quarks in full nucleon momentum, when the latter is moving at high speed. Contribution of quarks and anti-quarks in the nucleon momentum is expressed in terms of the distribution function as follows:

$$\sum_q \int_0^1 dxx[q(x) + \bar{q}(x)] = 1 - \epsilon,$$

which takes into account the contribution of valence quarks and sea quarks. If we assume that the quarks and antiquarks are the only candidate for the role of partons, then ϵ must equal to zero. However, experiments on probing nucleons with electrons which were performed in SLAC in the end of 1960-s (and later experiments on probing nucleons with neutrinos in CERN) estimated the value of ϵ to be close to 0.5. These results show that around 50% of the momentum is carried by partons which are not involved in neither electromagnetic, nor weak interactions. In QCD, the role of such constituents of the nucleon could naturally be claimed by gluons – carriers of the strong interaction between quarks.

From (3.10) one can write down the structure functions F_2 for electron-proton scattering, taking into account the quark charges from Table 3.1:

$$F_2^{ep}(x) = x \left[\frac{4}{9}(u + \bar{u}) + \frac{1}{9}(d + \bar{d}) \right], \tag{3.11}$$

where

$$u = u_p(x) \quad d = d_p(x) \quad \bar{u} = \bar{u}_p(x) \quad \bar{d} = \bar{d}_p(x)$$

are the quark density functions in the proton, which characterize the density of quarks for a given x.

Figure 3.8 illustrates a set of functions $xf_i(x)$ for distributions of quarks, antiquarks and gluons in a proton. It is seen that u and d-quarks carry a significant fraction of the proton's momentum with higher probability, while antiquarks and gluons have smaller x. Each hadron has its own set of parton distribution functions.

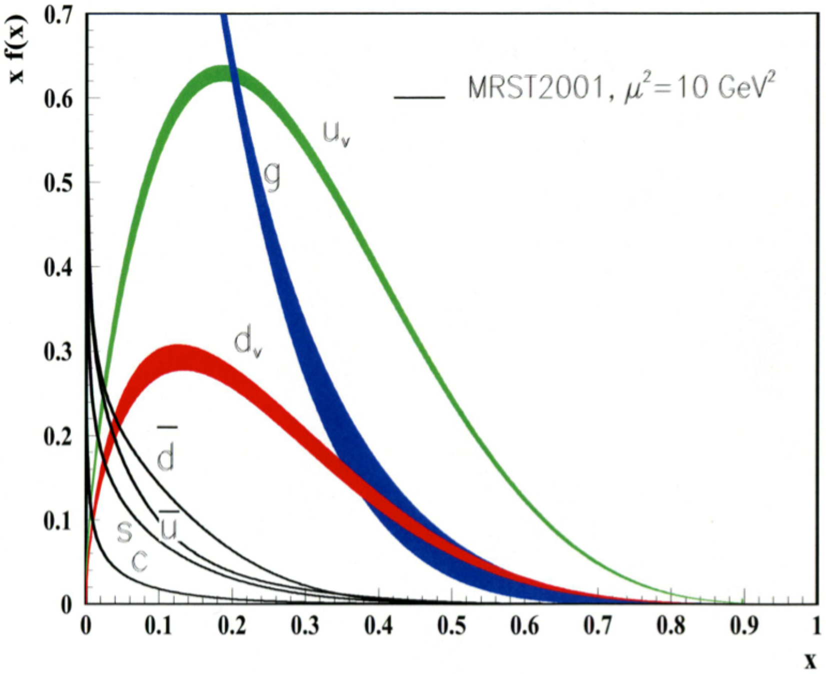


Figure 3.8: Behavior of the structure functions from the Q^2 evolution [22].

3.3 Color

With the evolution of the quark model [34], the diversity of hundreds of different hadrons was reduced to six pointlike particles – quarks, but in its turn this introduced new problems. One of such problems arises in the analysis of the quark composition of the baryon decuplet. The $\Delta^- = ddd$, $\Delta^{++} = uuu$, $\Omega^- = sss$ particles, which are arranged in the vertices of the "triangle" of the decuplet, are essentially the combinations of three identical quarks with the same quantum states, i.e. their full wave function are symmetric although they are fermions. This violation of the Pauli principle was eliminated by the introduction of an additional degree of freedom for quarks, the new quantum number which was called *color*, and the postulate that all the hadrons which are observed in nature are colorless. This color in reality is just an abstract property which has three values, analogized to the three primary colors of red (r), green (g) and blue (b). The introduction of color allows one to put three quarks of the same flavor in the same quantum-mechanical state inside a hadron. The Pauli principle is not violated, assuming that the wavefunctions are antisymmetric in this new degree of freedom (antisymmetric in color). Then, for example, Δ^{++} can be represented as a combination of three u-quarks in different color states: $\Delta^{++} = u_r u_g u_b$. Antiquarks have anticolors $(\bar{r}, \bar{g}, \bar{b})$. Each of the three possible states of color-anticolor $r\bar{r}$, $g\bar{g}$, $b\bar{b}$ is considered colorless (in fact, color-suppressed), and the combination of $r\bar{r} + g\bar{g} + b\bar{b}$, which does not change under rotations in the color space, is color singlet (white). A qq combination with any conjunction of colors of two quarks rr, rg, bg,... will be colored and therefore can not occur in nature by the postulate that only colorless bound quark states are observable in nature. This is called confinement. This excludes the possibility to observe separate, unbound quarks, since they are not colorless. In hadrons, quarks of different colors form a colorless combination, i.e. hadrons are *color singlets* with respect to rotations in the color space. With such a rotation a cyclic change of colors takes place, which mathematically means that there is a color $SU(3)$-symmetry.

Thus, mesons and baryons are $SU(3)$-singlets:

$$|\text{meson}\rangle_{\text{flavor}-\text{color}} = \frac{1}{\sqrt{3}} \sum_{\alpha=1}^{3} (q_i^{\alpha} \bar{q}_k^{\bar{\alpha}}),$$

$$|\text{baryon}\rangle_{\text{flavor}-\text{color}} = \frac{1}{\sqrt{6}} \sum_{\alpha,\beta,\gamma=1}^{3} (\varepsilon_{\alpha,\beta,\gamma} \cdot q_i^{\alpha} q_k^{\beta} q_j^{\gamma}),$$

where each color index α, β, γ takes values $1, 2, 3$ (or r, g, b), and $\varepsilon_{\alpha,\beta,\gamma}$ is an asymmetric tensor

$$\varepsilon_{123} = \varepsilon_{231} = \varepsilon_{312} = 1, \qquad \varepsilon_{213} = \varepsilon_{132} = \varepsilon_{321} = -1.$$

For example, the wavefunction of a π^--meson, taking into account flavor and color, is

$$|\pi^-\rangle = \sqrt{1/3}(d_r \bar{u}_{\bar{r}} + d_g \bar{u}_{\bar{g}} + d_b \bar{u}_{\bar{b}}).$$

Color is a natural candidate to act as the charge responsible for strong interactions, which is the exchange of a massless electrically neutral particle with spin 1, negative parity and zero isospin – *gluon*. In hadrons, this particle "glues" quarks together. A quark of a specific color may keep its color or it can change it while emitting or absorbing a gluon, while not violating the color (or color charge) conservation law. From here it follows, that every gluon carries a pair of color charges – colored and anticolored.

One can construct 9 pairwise combinations in a matrix:

	$\bar{r}$	$\bar{g}$	$\bar{b}$
r	$r\bar{r}$	$r\bar{g}$	$r\bar{b}$
g	$g\bar{r}$	$g\bar{g}$	$g\bar{b}$
b	$b\bar{r}$	$b\bar{g}$	$b\bar{b}$

Table 3.4: Color combinations.

It can be seen that the 6 non-diagonal combinations are colored and 3 diagonal have suppressed color. Because of the color charge conservation, the 6 non-diagonal colored pairs do not mix with each other, and the diagonal pairs mix (conservation of color charge does not prevent transitions $r\bar{r} \leftrightarrow g\bar{g} \leftrightarrow b\bar{b}$). As a result of these transitions, there are three linear combinations whose form can be obtained from the symmetry and the requirement of orthonormality of gluon states:

$$\frac{1}{\sqrt{2}}(r\bar{r} - g\bar{g}), \quad \frac{1}{\sqrt{6}}(r\bar{r} + g\bar{g} - 2b\bar{b}), \quad \frac{1}{\sqrt{3}}(r\bar{r} + g\bar{g} + b\bar{b}).$$

The latter combination is fully symmetric with respect to colors, i.e. it has no color charge (white, an absolutely colorless combination) and can not play the role of a gluon which participates in strong interaction (transporting color from one quark to another). Thus, only 8 gluons remain.

3.4 The QCD Lagrangian

Quantum Chromodynamics is a gauge theory of strong interactions of colored quarks and gluons, which at its base has the postulate that the color SU(3) is a local gauge symmetry. This requirement of local invariance leads to the conclusion of existence of the gluon fields octet.

The action S is an important quantity for the description of interactions. For fields, the action is defined as

$$S = \int \mathcal{L}(x) d^4x,$$

where $\mathcal{L}$ is a Lagrangian or Lagrange function density in $x(ct, \vec{r})$. The action contains all the dynamics of fields and particle interactions. All systems comply to the variational principle of Fermat: $\delta S = 0$. The equations of motion are obtained from S and $\mathcal{L}$. Therefore it is said that the construction of the theory of elementary particles is reduced to finding the fundamental Lagrangian, which describes the physical world, and to solving the equations of motion arising from it.

The general form of the QCD Lagrangian [32, 35] ($\hbar = c = 1$) is

$$\mathcal{L}_{\text{QCD}} = -\frac{1}{4} F^a_{\mu\nu} F^{a\mu\nu} + \sum_q \bar{q}_i (i\gamma^\mu D_\mu - m_q)_{ij} q_j \ , \tag{3.12}$$

where repeated indices are summed over.

The form of the QCD Lagrangian is dictated by the requirement of invariance under local gauge transformations of the non-abelian group $SU(N_c)$, $N_c = 3$:

$$q(x) \ \rightarrow \ U(x)q(x), \qquad \bar{q}(x) \ \rightarrow \ \bar{q}(x)U^\dagger(x),$$

where 3×3 $U(x) = \exp\{i\theta_a(x)T_a\}$ transformations are unitary and unimodular

$$UU^\dagger = 1, \quad U^\dagger_{ji} = U^*_{ij}, \quad \det U = 1,$$

$\theta_a(x)$ are parameters which depend on the coordinates of the world point x, $T_a = \lambda_a/2$ and λ_a are the Gell-Mann matrices (3.14).

The requirement of invariance is accomplished [32,35] by the introduction of $N_c^2 - 1 = 3^2 - 1 = 8$ gauge fields $A^a_\mu(x)$ $(a = 1, 2, \cdots, 8)$, by replacing the derivative ∂^μ by the covariant derivative D_μ (3.13) and the A^a_μ field transforms as [32]

$$A_\mu(x) \ \rightarrow \ U(x)A_\mu(x)U^\dagger(x) + \frac{i}{g_s}[\partial_\mu U(x)]U^\dagger(x).$$

In Eq. (3.12) the γ^μ are the Dirac γ-matrices, the field strength tensor $F^a_{\mu\nu}$ and the covariant derivative D_μ are

$$\begin{aligned} F^a_{\mu\nu} &= \partial_\mu A^a_\nu - \partial_\nu A^a_\mu - g_s f^{abc} A^b_\mu A^c_\nu, \\ (D_\mu)_{ij} &= \delta_{ij}\partial_\mu + i g_s T^a_{ij} A^a_\mu, \\ (m_q)_{ij} &= m_q \delta_{ij}, \end{aligned} \tag{3.13}$$

where g_s is the QCD coupling constant. The fundamental parameters of the QCD Lagrangian are the coupling g_s (or $\alpha_s = \frac{g_s^2}{4\pi}$) and the quark masses m_q.

The gauge fields of strong interactions A_μ^a are called gluons, and fields q_i are the quark fields of flavor i (q_i: $u, d, s, c, b, \cdots$), f^{abc} are the structure constants, $T_{ij}^a = \lambda_a/2$ correspond to eight 3×3 matrices and are the generators of the SU(3) group,

$$\lambda_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \lambda_2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\lambda_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \lambda_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

$$\lambda_5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix}, \qquad \lambda_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{3.14}$$

$$\lambda_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \qquad \lambda_8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.$$

The structure constants of the SU(3) group $f^{abc} = f_{abc}$ are real, antisymmetric in their indices and are defined by the commutation relations

$$[T^a, T^b] = if^{abc}T^c.$$

For $SU(3)$ the non-vanishing components of f^{abc} have the following values

$$\begin{cases} f_{123} = 1, \\ f_{147} = f_{165} = f_{246} = f_{345} = f_{376} = f_{257} = 1/2, \\ f_{458} = f_{678} = \sqrt{3}/2. \end{cases}$$

The term $-\frac{1}{4}F_{\mu\nu}^a F^{a\mu\nu}$ contains members of the from

$$\sim g_s A_\mu^a A_\nu^b A_\lambda^c \quad \text{and} \quad \sim g_s A_\mu^a A_\nu^b A_\lambda^c A_\sigma^d,$$

which correspond to the interactions between gluons. They arise due to the nonlinear terms in $F_{\mu\nu}^a$. As a result, QCD Feynman diagrams include gluons-only vertices (3-gluon and 4-gluon vertices), therefore it can be concluded that gluons have nonlinear self-interaction because they have color charge.

Figure 3.9 presents an easy to understand graphical representation of the QCD Lagrangian [32]. Here the gluon colors are indexed by $a, b, c, d, e = 1, \cdots, 8$ and the quark colors by $i, j = 1, 2, 3$.

It is seen in the first part of the Lagrangian that the field strength tensor (3.12) is the source of gluons (a free field term and two interaction terms with gluons, which couple to gluons).

The second part of the Lagrangian, a sum over all quark flavors, is a fermionic part (a free field term and a term of the quark-gluon coupling). From this scheme it is seen, for example, that the four-gluon coupling between gluons of color states a, b, c, d is proportional to $g_s^2 f^{abe} f^{cde}$, but the coupling between two quarks of colors i, j to a gluon of color state a is proportional to $g_s T_{ij}^a$.

Neither quarks nor gluons are observed as free particles, but the observed hadrons (color

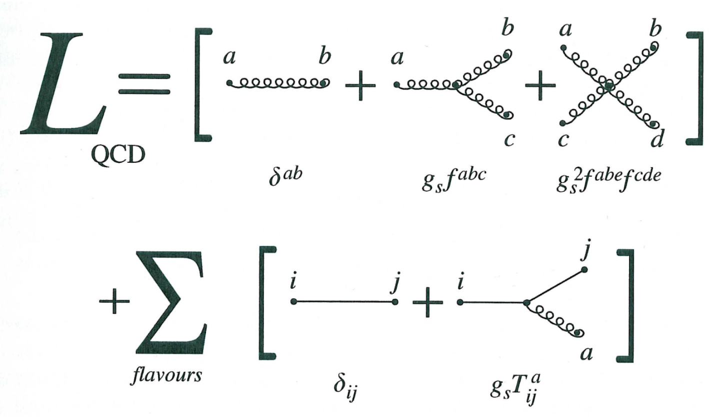


Figure 3.9: Scheme of the QCD Lagrangian [32].

singlet states) consist of quarks, antiquarks and gluons.

3.5 The running coupling

QCD is the current theory which describes the interaction of hadrons at short distances, and is based on the quark color model. It assumes that quarks interact by exchanging gluons, and the strength of the interaction is determined by the effective coupling constant α_s (effective charge), which depends on the square value of the transferred four-momentum Q^2 and is called the running coupling of strong interaction.

As it was previously discussed in Section 3.4, the requirement of gauge invariance in QCD introduced self-interaction of gluons. This in turn leads to an important property of QCD, called *asymptotic freedom*, i.e. with the increase of the transferred value of Q^2, the effective constant of strong interaction α_s grows smaller, logarithmically in leading order [32]:

$$\alpha_s(Q^2) \equiv \frac{g_s^2(Q^2)}{4\pi} = \frac{1}{\beta_0 \ln(Q^2/\Lambda^2_{\text{QCD}})}, \tag{3.15}$$

where $\beta_0 = 11 - 2/3n_f$, n_f – is the number of quark flavors. The Λ_{QCD} constant plays a fundamental role in QCD. Λ_{QCD} is a scale parameter of QCD – an energy scale at which non-perturbative effects become important (α_s grows large with the decrease of Q^2). The value of Λ_{QCD} is extracted from experimental data and it is $\mathcal{O}(200)$ MeV or $\Lambda_{\text{QCD}} \sim 1/R_0$, where $R_0 \sim 10^{-13}$ cm is a typical hadron size.

It is seen that at $Q^2 \to \infty$, the coupling constant grows smaller $\alpha_s(Q^2) \to 0$ and quarks stop interacting: the effective color charge vanishes as the colored object is closer approached, i.e. the object becomes asymptotically free (noninteracting). This phenomenon is a very important property of QCD. It helps to naturally explain the structure of hadrons, so that the predictions of the quark-parton model are valid in the asymptotic limit.

For example, the coupling constant becomes of the order of $\alpha_s \approx 0.1$ for momentum transfers in the range of 100 GeV - TeV [22]. Thus, the decrease of $\alpha_s(Q^2)$ with increasing Q^2 makes it possible to use perturbation theory for large enough values of Q^2. In this case QCD changes into free particles theory, up to logarithmic corrections which can be calculated perturbatively. By performing an expansion in a small parameter α_s of relations which describe the evolution of functions dependent on Q^2, it is possible to compare the predictions of the evolution to the experimental data. Thus, the property of asymptotic freedom allows the use of perturbation theory to describe high-energy processes, and for example multi-particle production processes. When the value of Q^2 approaches Λ_{QCD}, α_s increases and becomes infinite. As already described in Section 3.3, this phenomenon is called *confinement* and means that quarks cannot be found in a free state outside of hadrons: quarks and gluons cannot escape the region of strong interaction and be observed as single physical objects, which was proved experimentally - so far there is no evidence for free quarks in nature.

Growth of the coupling constant and the confinement phenomenon do not allow to obtain in QCD theoretical predictions for low-energy processes and to describe properties of hadrons. In this region, the perturbation theory does not work. Magnitudes and effects which cannot be described by the perturbation theory are called nonperturbative. There are ways to account for them semi-phenomenologically, or alternatively, lattice gauge theory can be used for calculating them.

The main experimental confirmations of QCD are related to its perturbative predictions. In particular, an important test of QCD is the match of $\alpha_s(Q^2)$ in the perturbative region against formulas (3.15). The reduction of the coupling strength with increasing energy has been confirmed experimentally and is illustrated in Fig. 3.10.

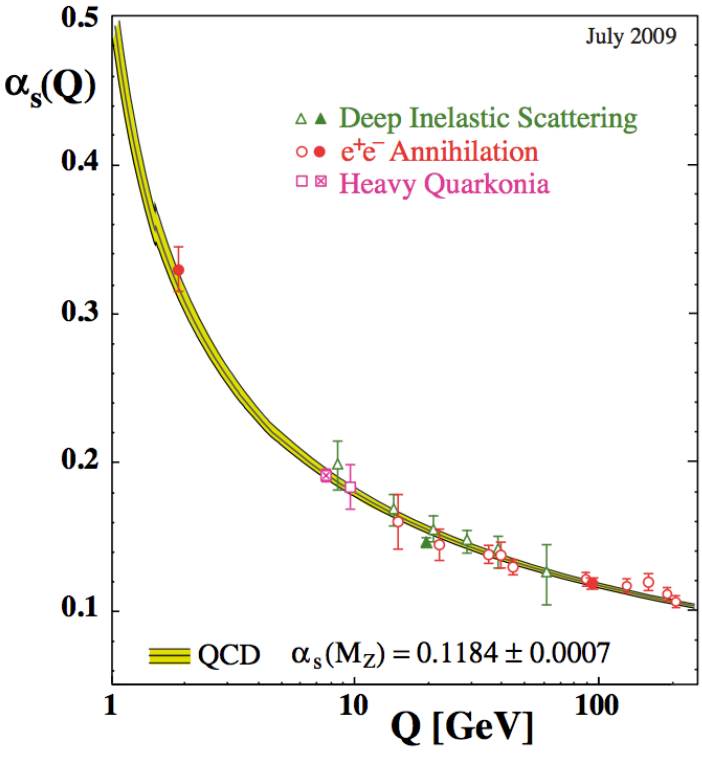


Figure 3.10: Summary of measurements of α_s as a function of the respective energy scale Q [22].

As follows from (3.15), calculations in QCD are divided into two main modes. One of them is in effect in the region of large Q^2 and short distances (perturbative), and the second one – in the region of small values of Q^2 and long distances (nonperturbative).

Thus, the asymptotic freedom makes it possible to use the perturbation theory through the separation of contribution of short and long distances into hadronic processes (the so-called factorization).

For example, due to the running coupling in processes with $Q \gg \Lambda_{\text{QCD}}$ ($\Lambda_{\text{QCD}} \sim 1/R_0$), in the parton model it is possible to consider hadrons as objects which consist of almost free partons – (anti)quarks and gluons, which are characterized by parton density functions (p.d.f.s).

3.6 Proton-proton scattering

The high-energy pp collisions which are studied at the LHC, are a special case of hadron-hadron scattering. They are characterized by a variety of reactions and can be divided into two main classes:

1. those which involve soft interactions, which are described by small momentum transfer (long-distance reactions) and in which hadrons interact with each other as whole, integral objects. The cross sections of these reactions are usually large and change slowly with the centre of mass (C.o.M). energy. Phenomenological approaches are used to describe these reactions (the total, elastic, single/double diffractive cross sections).

2. interactions which involve hard scattering, have large momentum transfer (short-distance reactions) and thus can probe the internal structure of the interacting hadrons. In this case, perturbative QCD is applied to describe these reactions (the high transverse energy jet, heavy quark and high mass lepton pair production).

General properties of hadron-hadron total cross sections. Figure 3.11 presents the general behavior of the total cross sections in hadron-hadron collisions as a function of the C.o.M. energy.

It can be seen that the total cross sections of hadron interactions slowly decrease with increasing energy (with minimum around $\sqrt{s} \sim 20$ GeV) and start to increase at energies of a few tens of GeV. A resonance structure is evident with $\sqrt{s} \leq 3$ GeV. Above the resonance the total hadron-hadron cross section is proportional to the product of the number of (anti)quarks in the colliding particles. Therefore, e.g. the cross section of the meson-baryon interaction is about 2/3 of the cross section of baryon-baryon interaction: $\sigma_{tot}(\pi^- p) \approx \sigma_{tot}(K^- p) \approx 2/3 \times [\sigma_{tot}(pp) \approx \sigma_{tot}(p\bar{p})]$. With further increase of energy, total cross sections also increase. Experiments show that the increase in cross sections is universal for hadronic interactions. This indicates that at high energies new additional mechanisms of interaction appear.

Soft proton-proton scattering. Proton-proton scattering can be divided into elastic scattering, diffractive scattering and inelastic scattering.

Elastic scattering corresponds to the process in which the participating particles slightly deviate from their initial trajectories after the interaction, i.e. $p + p \to p + p$. Elastic scattering is characterized by space-like momentum transfer $t = (p_{\text{in}} - p_{\text{out}})^2 < 0$.

The t-range can be separated into the following regions in which either electromagnetic or strong forces dominate:
$|t| < 0.001$ GeV^2 – the well understood Coulomb region;
$0.001 < |t| < 0.01$ GeV^2 – the interference region;
$0.01 < |t| < 0.15$ GeV^2 – the diffraction region.

The differential cross section has the following typical behavior as a function of $|t|$: the strong peak at $|t| < 0.01$ GeV^2 decreases to a minimum at $|t| \sim 1.4$ GeV^2, which is followed by the wide peak at $|t| \sim 2$ GeV^2. At $|t| > 2$ GeV^2 the decrease of the differential cross section is described as t^{-n} (the diffractive region), where $n = 10$ for baryon-baryon scattering. All this is well seen in Fig. 3.12 which demonstrates the t-dependence of pp and $p\bar{p}$ scattering.

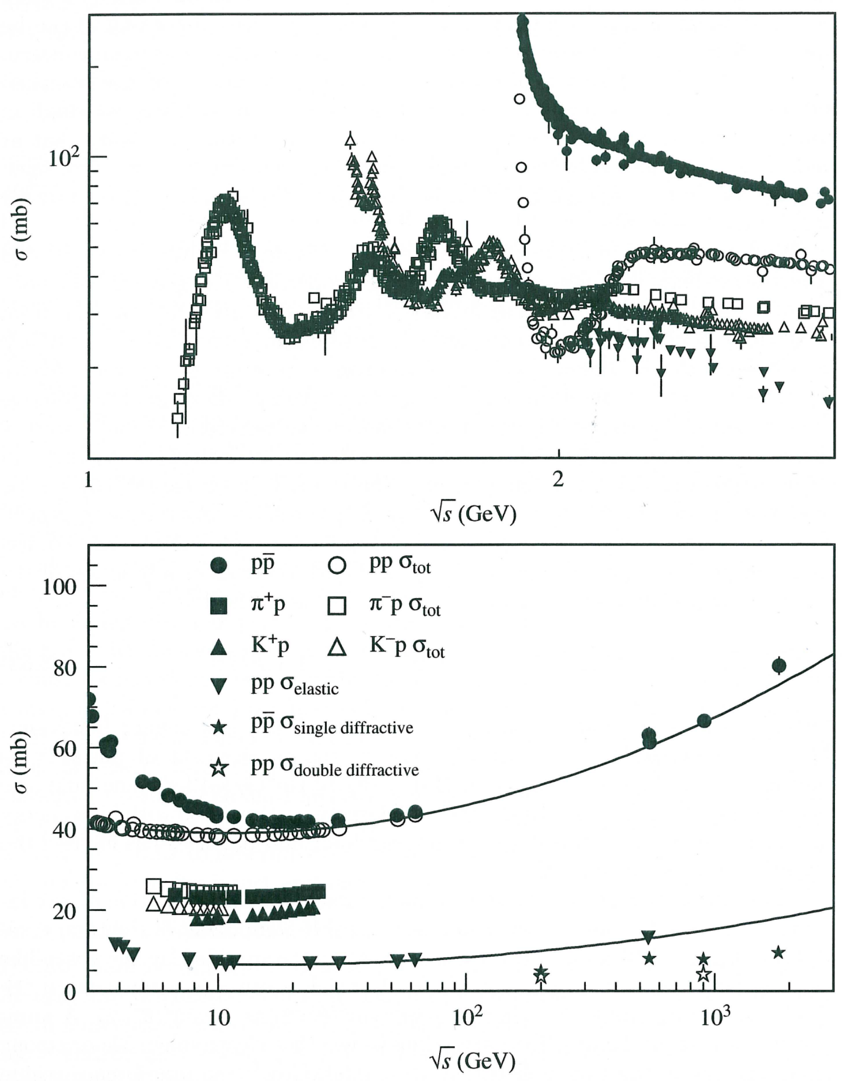


Figure 3.11: Measured cross sections in hadron-hadron collisions as a function of the C.o.M. energy [32].

In the diffractive region the differential cross section is approximated by [32]

$$\frac{d\sigma_{el}}{dt} \propto \begin{cases} e^{At+Bt^2} & |t| \ < 0.4 \ \text{GeV}^2 \\ t^{-n} & |t| \ > 3.0 \ \text{GeV}^2 \end{cases} \tag{3.16}$$

The next important class of interactions are reactions which involve diffractive dissociation processes, in which, during elastic scattering, one of the colliding protons becomes excited without the change of the internal state of the second proton, or both of the interacting protons excitate (single or double diffractive dissociation). These processes produce relatively few particles, because they come from the decays of the excited protons but not from the initial interaction, thus their production consumes a small fraction of the energy.

Phenomenologically, diffractive dissociation processes include such inelastic and multiple processes which involve protons, characterized by the following properties:

1. differential cross sections are strongly extended forward, the main part of the process is

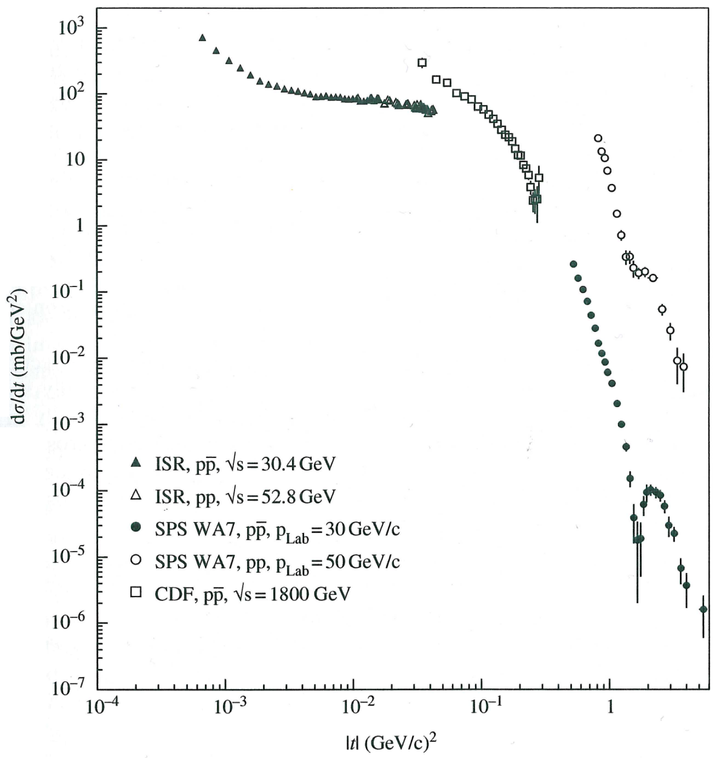


Figure 3.12: Measured differential cross section in $p\bar{p}$ and pp collisions as a function of the momentum transfer t for various C.o.M. energies [32].

associated with a small momentum transfer;

2. cross sections are almost independent of energy, increasing at very high energies ($>$ 100 GeV in the lab frame) proportionally to $\ln(s/s_0)$, where $\sqrt{s}$ is the energy in the center of mass frame, and s_0 is a constant of energy squared;

3. there is a large rapidity gap between the groups of produced particles;

4. cross sections involving particles and antiparticles are equal;

5. t-channel processes, dominated by Pomeron exchange.

Sketches of diffractive scattering processes are presented on of Fig. 3.13 (zig-zag lines mark Pomeron exchanges which are dominant at high energies). The left diagram corresponds to the single diffractive dissociation process with only one excited vertex. The double diffractive dissociation is represented by the middle diagram, which corresponds to the reaction when excitation occurs in both interaction vertices. The central diffractive process with double Pomeron exchange $p + p \to p + p + X$ (where X stands for a number of produced hadrons) is presented on the right diagram of Fig. 3.13. Description of this type of processes is possible with the help of the exchange of two Pomerons at higher energies than for the processes shown in the left diagram.

The following predictions of the cross sections for pp collisions were made for the LHC running with $\sqrt{s} = 14$ TeV [32] $\sigma_{\text{tot}} \approx 105$ mb, $\sigma_{\text{el}} \approx 25$ mb, $\sigma_{\text{SD}} \approx 15$ mb, $\sigma_{\text{DD}} \approx 10$ mb. The central diffraction cross section is estimated to be a few per cent of the total dissociation cross section.

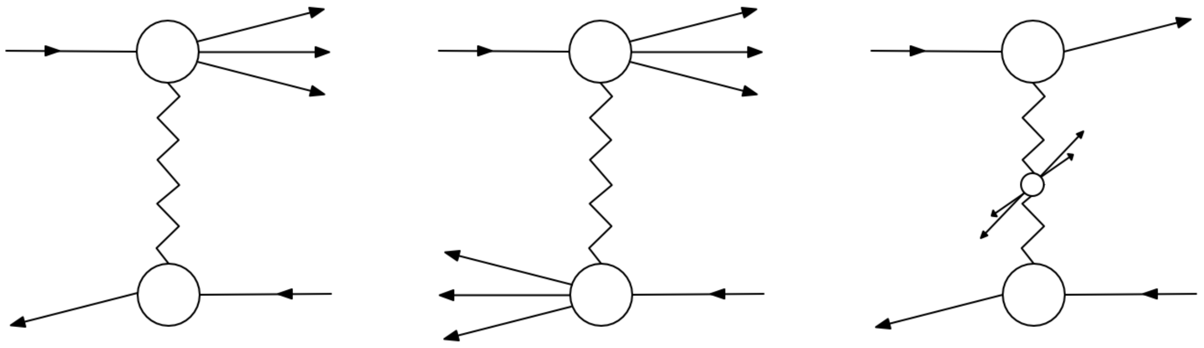


Figure 3.13: From left to right: single diffractive dissociation (SD), double diffractive dissociation (DD) and central diffractiv (CD) events.

Hard proton-proton scattering. The rare hard events are of a particular interest for the LHCb experiment. They can be classified as the events that contain large momentum scale and short-distance scale, and work with quark and gluon constituents in describing *pp* (hadron-hadron) interactions.

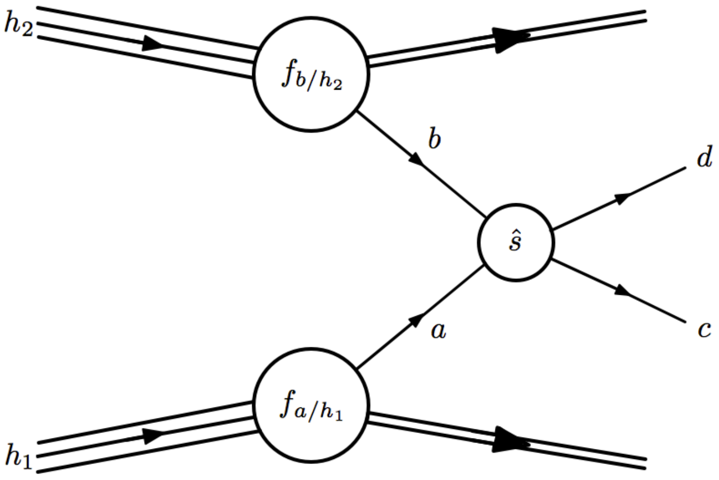


Figure 3.14: Schematic diagram for the production of final state particles c and d in a hard collision of hadrons h_1 and h_2.

In perturbative QCD, a factorization theorem is applied for a description of a differential cross section of an arbitrary hard process in a collision of two particles. For example, for the collision of hadrons h_1 and h_2, which produce particles c and d (see Fig. 3.14) the expression for the cross section can be written as

$$d\sigma_{h_1 h_2 \to cd} = \int_0^1 dx_1 dx_2 \sum_{a,b} f_{a/h_1}(x_1, \mu_F^2) f_{b/h_2}(x_2, \mu_F^2) d\hat{\sigma}_{ab\to cd}(\mu_F^2, \mu_R^2). \tag{3.17}$$

Here the partons $a, b \in \{q, \bar{q}, g\}$ are from the hadrons h_1, h_2;

f_{a/h_1}, f_{b/h_1} are the distributions of partons a, b in hadrons h_1, h_2 (are the same p.d.f.s as in DIS);

the parton momenta are $p_a^\mu = x_1 p_{h_1}^\mu$, $p_b^\mu = x_2 p_{h_2}^\mu$, where x_1, x_2 correspond to the fractions of the momenta of partons a, b;

μ_F is the factorization scale (hard interaction scale) parameter. The p.d.f.s depend on the factorization scale parameter μ_F, which is chosen and fixed and, is usually $\mathcal{O}(\mu_R)$. This parameter specifies the energy scale that separates (factorizes) physics at long distances and which can not be quantitatively represented in QCD;

$d\hat{\sigma}_{ab\to cd}$ is a differential cross section of the hard subprocess $ab \to cd$, which can be calculated in QCD perturbation theory in the form of a series in powers of the running coupling and depends on the renormalization scale μ_R for QCD running coupling.

The summation is done over all partonic subprocesses which contribute to the production of particles c and d.

Contributions of $2 \to 2$ processes into the hard subprocess cross section are the leading ones ($\mathcal{O}(\alpha_s^2)$, $\mathcal{O}(\alpha_s\alpha_{em})$). For example, the processes $q\bar{q} \to Q\bar{Q}$, $gg \to Q\bar{Q}$ and $qg \to q\gamma$, $q\bar{q} \to g\gamma$ contribute to the production of a pair of heavy quarks (Q) and to the prompt photon (γ) production respectively. To take into account the higher orders in the perturbative expansion $\hat{\sigma}$, it is necessary to consider $2 \to 3$ and more complex processes, in which singularities appear when the partons become collinear. This is the region of the factorization of singularities. The appearance of jets associated with each of the outgoing partons is a consequence of the near collinear radiation.

There are many *ad hoc* parameters in the QCD inspired models which need to be determined from the data. Chapter 4 provides some insight into how this is modeled in PYTHIA.

Chapter 4

Monte Carlo simulation

Utere et abutere.
Lat.: Use and abuse.

This chapter describes the LHCb software framework and its applications, which were used in the Monte Carlo tuning process. Section 4.1 begins this chapter with a short explanation on how the discussed theory is implemented in Monte Carlo events generators. Section 4.2 discusses the PYTHIA event generator and its process of event simulation, followed by Section 4.3 which briefly introduces detector simulation. Section 4.4 concludes this chapter with an overview of the LHCb software framework.

4.1 Using the Monte Carlo event generator

The software used at LHCb relies on the PYTHIA Monte Carlo event generator [9] to simulate hadronic events which occur in pp collisions. Since the description of simulated events is based on QCD, theoretically a generator requires only 7 free parameters: the strength of the coupling α_s and the six quark masses. However, in reality Monte Carlo event generators have many miscellaneous parameters and options due to the use of perturbative approximations and non-perturbative models (the wide range of hard subprocesses, models for soft physics, the well developed Lund string model [9] for the hadronization scheme and so on).

The values of many of these parameters are defined by tuning the Monte Carlo generators to the experimental data. Usually a $\chi^2(p_1, p_2, \cdots, p_n)$ function is calculated and the result of its minimization is considered the best set of parameter values within their corresponding one-sigma errors (for more details see Chapter 4).

The necessity to generate large samples of data, the multidimensionality of the parameter space, the large number of distributions which are used as comparison criteria, accounting of uncertainties and the problem of stability of numerical minimization - all of these problems make tuning procedures a complicated task, transforming it into art.

There are different tuning approaches, but usually the process is split into several stages and the initial tuning is done with a subset of 'major' parameters, e.g. Λ_{QCD}, the principle hadronization model parameters, shower model parameters, etc. Then with the values of these parameters fixed, the next tuning iteration is done for the next subset of parameters, and so on.

4.2 Event generation and PYTHIA

During the event generation phase, the Gauss application is used to simulate processes which take place during and right after the proton-proton collisions. In fact, to simulate the collisions it relies on the external event generators, such as: PYTHIA 6/8, HIJING, SHERPA, HERWIG/HERWIG++. Currently PYTHIA 6.4 is still used as the default one though the LHCb software framework is migrating to PYTHIA 8.

PYTHIA is a general purpose event generator which contains theory and models for a number of physics aspects, such as hard and soft interactions, multiple interactions, parton distributions, initial and final state parton showers, fragmentation (the Lund model) and decays. For description of these physics aspects, PYTHIA uses various models with phenomenological parameters. These parameters are included in the default parameter set of the Monte Carlo event generator.

The PYTHIA program work flow is divided in the three steps:

First step, definition of the studied process(es);

Second step, generation of the initial state showers and beam fragments;

Third step, generation of the final state showers, fragmentation (Lund model) and decays.

The PYTHIA event generator can simulate many physics aspects which are described by different models, some of which contain phenomenological parameters that control their behavior. These parameters are included in the default parameter set in PYTHIA and their values are usually determined from experimental data.

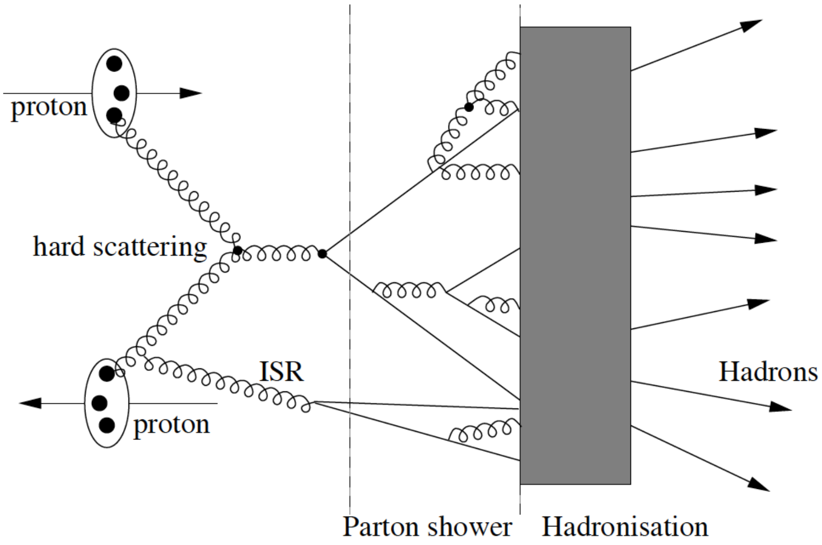


Figure 4.1: Scheme of a *pp* collision with modeled stages of the initial state radiation (ISR), the hard scattering process, the parton shower and the hadronisation [7].

The physics in the Monte Carlo event generator simulation can be divided into two parts: the perturbative and non-perturbative regimes which are separated by a *hadronization scale* $Q_0 \sim 1$ GeV. The perturbative regime is limited by the hard scattering region and is characterized by the large momentum transfers $Q^2 \gg \Lambda^2$. The non-perturbative models are used for the description of physics at low scale, i.e. for the simulation of the hadronic final-state. These phenomenological models contain parameters which must be *tuned* to experimental data. The parton model provides a factorized form for QCD physics description, with separation of the low and high scale physics. The parton shower evolves the partons from the hard scales (hard subprocess) down to the hadronization scale where hadronization models can be applied.

A typical *pp* event simulation can be divided into the following steps, which are time ordered:

- Two beams of particles (protons), moving in the opposite directions, collide, creating a hard interaction. Particles in the beams are characterized by structure functions, which describe partonic structure.

- An initial state parton shower is created as a sequence of branchings by the shower initiator partons from each of the beams.

- One incoming parton from each of the showers enters the hard process, where then a number of outgoing partons are produced. The outgoing partons may also branch, to build up final state showers.

- Due to confinement the outgoing colored quarks and gluons cannot leave the interaction region but instead they must fragment to the colorless hadrons (fragmentation). The process of transition of colored partons into colorless hadrons is called fragmentation (or hadronization).

- And the last step is a decay of the produced unstable hadrons.

The major parameters of the perturbative QCD treatment are Λ (or α_s) and the shower cut-off Q_0, the a and b parameters of the Lund symmetric fragmentation function, and the width of the transverse momentum distribution σ. Further we shall consider how some of the parameterizations and models which depend on these parameters are realized in PYTHIA.

The total hadron-hadron cross-section.

PYTHIA uses the Donnachie and Landshoff [36] parameterization to calculate the total hadronic cross-section σ_{tot}^{ab} in hadron-hadron interaction $ab \to$ anything and is the sum of two terms (Pomeron and Reggeon):

$$\sigma_{\text{tot}}^{ab}(s) = X^{ab}s^{\epsilon} + Y^{ab}s^{-\eta} = X^{ab}s^{\epsilon} + Y^{ab}\frac{1}{s^{\eta}}. \tag{4.1}$$

Here $s = E_{cm}^2$, the powers $\epsilon = 0.0808$ and $\eta = 0.4525$ are expected to be universal, (the data indicate that they vary very slowly with s, so they are almost constants), the coefficients X^{ab} and Y^{ab} are specific for each initial state, where at high energies $X^{ab} = X^{\bar{a}b}$, i.e. the Pomeron term is expected to be the same for the particle and antiparticle. From (4.1) it is seen that the total hadronic cross-section increases with an increase of s, herewith the first term dominates. For example, the parameterization (4.1) for *pp* looks like [36] (see Fig. 4.2)

$$\sigma_{\text{tot}}^{pp}(s) = 21.70s^{0.0808} + 56.08s^{-0.4525}$$

and for $\bar{p}p$

$$\sigma_{\text{tot}}^{\bar{p}p}(s) = 21.70s^{0.0808} + 98.39s^{-0.4525}.$$

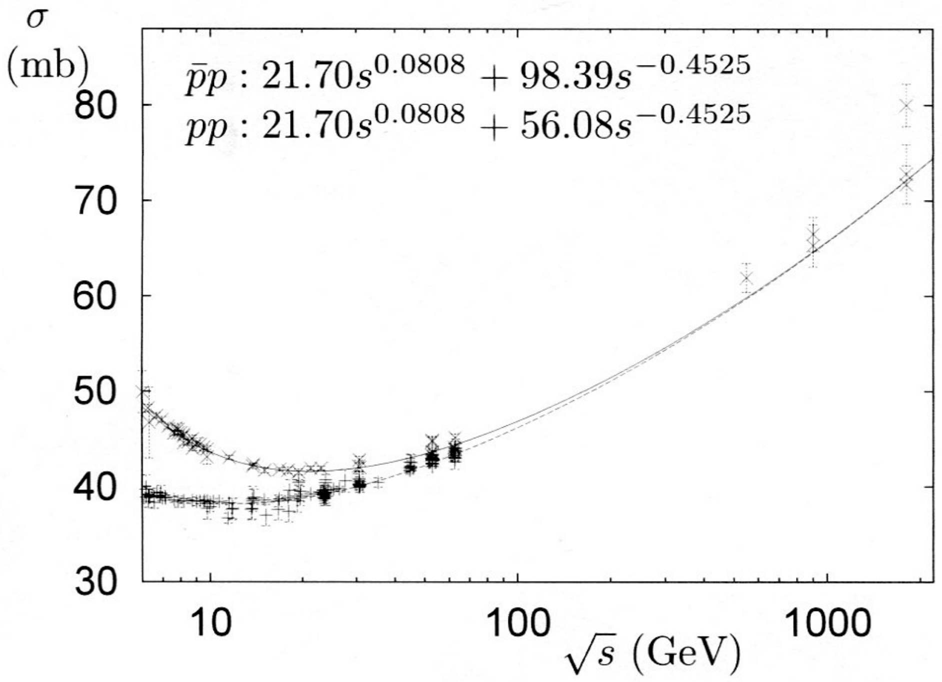


Figure 4.2: Data for total cross-sections with fits of type (4.1) [36].

Parameterizations not provided by [36] are calculated as

$$\sigma^{ab}_{\text{tot}}(s) = \sigma^{ab}_{\text{el}}(s) + \sigma^{ab}_{\text{sd}(Xb)}(s) + \sigma^{ab}_{\text{sd}(aX)}(s) + \sigma^{ab}_{\text{dd}}(s) + \sigma^{ab}_{\text{nd}}(s). \tag{4.2}$$

Here $'$el$'$ is the elastic process: $ab \to ab$; $'sd(Xb)'$ is the single diffractive dissociation: $ab \to X + gap + b$, with X denoting anything but the original beam particle and 'gap' is a rapidity region devoid of observed activity; $'sd(aX)'$ is the single diffractive dissociation: $ab \to a{+}gap{+}X$; $'dd(X_1X_2)$' is the double diffractive dissociation: $ab \to X_1 + gap + X_2$; $'nd'$ is the inelastic non-diffractive scattering: everything else ($\sigma^{ab}_{nd} \subset \sigma^{ab}_{hard}$). The central diffraction $ab \to a + gap + X + gap + b$ processes are usually neglected. The elastic, diffractive cross-sections are computed with the help of Regge theory [9].

The factorization formula for QCD cross-section. The cross-section for a scattering subprocess $ab \to n$ at hadron colliders is calculated with the formula

$$\sigma_{ab\to n} = \sum_{ab} \int dx_i \int dx_j f^{h_1}_a(x_a, Q^2) f^{h_2}_b(x_b, Q^2) \hat{\sigma}_{ab\to n},$$

where $f^h_a(x_a, Q^2)$ is the parton distribution function (PDF) that describes the probability to find a parton a inside a beam particle h, with parton a carrying a fraction x_a of the total h momentum, Q is the hard interaction scale.

Parton distribution functions contain nonperturbative information (dependence of x on a fixed scale Q), but their dependence on the Q scale can be predicted for large values of Q using the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi evolution equations (DGLAP [32].

The cross-section for the hard partonic process is $\hat{\sigma}_{ab\to n}$ (scattering cross-section of partons a and b), which in QCD is calculated as a series in powers of the running coupling $\alpha_s(Q^2)$.

One of the main sources of information on the parton distribution functions are the deep inelastic lepton-nucleon scattering processes. A set of PDFs is incorporated in a software package, in which the values of the PDFs are presented on a grid of (x, Q) values and are interpolated to any point in the space of the variables.

The parton showers. The structure of initial- and final-state showers is given in terms of branchings $1 \to 2$ or $a \to bc$, where the mother a decays into two daughters b, c. The following branchings are included in PYTHIA: $q \to qg$, $q \to q\gamma$, $g \to gg$, $g \to q\bar{q}$, $l \to l\gamma$. The branching kinematics is given in terms of Q^2 and z variables: Q^2 is related to the mass or transverse momentum scale of the branching, z gives the sharing of the a energy and momentum between the daughters (b takes a fraction z and c takes the remaining $1 - z$ of the a energy). The formed partons b and c can in turn branch, and so on.

Each process is described by a splitting kernel $P_{a\to bc}(z)$ which is interpreted as the probability for the parton a to branch into b, c and which is calculated in PYTHIA by the formula, corresponding to each process [9]. The branching rate is proportional to $\int P_{a\to bc}(z)dz$.

A "virtuality scale" Q^2, which characterizes each parton, gives a sense of time ordering to the cascade. Each parton initiates a virtual cascade, branching into a number of partons. The evolution variable Q^2 is associated with the m^2 of the branching parton. The final-state showers are time-like (i.e. partons have $m^2 = E^2 - \mathbf{p}^2 \geq 0$). Starting from some $Q^2_{\max}$, the parton shower evolution is based on calculating the Sudakov form factor [22], $\Delta(Q_{\max}, Q_1)$, which is computed for each parton and corresponds to the probability that the parton does not split into other partons $a \to bc$ between the hard scale $Q_{\max}$ and some smaller scale Q_1. The value Q_1 is determined from the expression $r = \Delta(Q_{\max}, Q_1)$, in which r is a random number in the interval $[0, 1]$. This selected Q_1^2 value defines the mass of the branching parton and the z value, which is determined from the splitting kernel, represents the division of the parton energy between its daughters. Each daughter may evolve downwards, i.e. the process is repeated in order to obtain the scale Q_2 of the next emission, and so on. The shower evolution is cut off at some lower scale $Q_0 = Q_{\text{shower cut−off}}$, usually around 1 GeV for QCD branchings that separates the perturbative and non-perturbative part of the simulation. In PYTHIA the shower cut-off parameter is one of many which are usually tuned to data.

As a next step, the Monte Carlo generator hadronizes the simulated partonic configuration, according to its hadronization model. PYTHIA uses the Lund model, which is described later in this chapter. Historically, PYTHIA handles the evolution of the initial-state showers itself and uses the JETSET packages for final-state showers [9].

The incoming charged particles can radiate before entering the hard subprocess, which results in obtaining a non-vanishing transverse momentum and the virtuality of such particles becoming negative (space-like). The initial-state showers are space-like, i.e. in the sequence of branchings $a \to bc$, obtained in the result of evolution of the process from the shower initiator to the hard interaction, particles a, b have $m^2 = E^2 - \mathbf{p}^2 < 0$. In this case, the evolution variable $Q^2 = -m^2$, which characterizes the evolution of the space-like 'main branch', increases along the shower towards the hard subprocess, i.e. $Q^2_b > Q^2_a$. Each branching $a \to bc$ produces a space-like parent parton b and a time-like daughter parton c (it does not participate in the hard scattering), which continue to evolve according to the backward initial-state and forward final-state shower algorithms.

In PYTHIA, the process with the initial-state radiation is implemented by the 'backwards evolution' scheme, when the hard scattering event is selected as the first step and then the parton shower is reconstructed. The reconstruction is a 'backwards in time' step by step motion which starts from the $Q^2_{\max}$ scale to the cut-off scale Q^2_0, back to the shower initiator (i.e. for a given parton b, the parton a which branched into b is defined). This scheme is possible if the evolved parton distributions are used for the choice of the hard scattering, because the functions $f_i(x, Q^2)$ contain the inclusive effects of initial-state radiation. The evolution in the Monte Carlo program is done in terms of a sequence of decreasing the space-like Q^2 and increasing the momentum fractions x.

Confinement, hadronization and the Lund string model. The mechanism of confinement is used to explain the negative results of experiments, designed to register single, unbound quarks. It is a model which describes the behavior of quarks inside hadrons.

The interaction between a pair of quarks is described by a potential of the following form

$$V(r) \approx -\frac{\alpha_s \hbar c}{r} + \kappa r. \tag{4.3}$$

In this formula, the first term, dominating at short distances ($<$ 0.2 fm), is similar to the Coulomb's potential, and the second term, which plays the major role at distances $r \sim 10^{-13}$ cm $= 1$ fm. Here, κ is the energy density per unit length of a color flux tube formed by gluon self-interaction between two colored partons (experimentally, $\kappa \approx 1$ GeV/fm).

The string fragmentation is a very popular model which is used in the PYTHIA Monte Carlo event generation application. The program uses the phenomenological Lund string model, in which the confinement is realized by stretching strings between the outgoing q and $\bar{q}$ partons, which move apart from each other. A one-dimensional string is represented as an axis of a color cylindrically symmetric flux tube, formed by gluon self-interaction. The tube is considered to be uniform along its length with the string tension $\kappa \approx 1$ GeV/fm, which leads to the confinement mechanism with a linearly rising potential (the short distance Coulomb term in (4.3) is neglected). Since the q and $\bar{q}$ move in the opposite directions, stretching the string, the potential energy stored in the string increases. When the energy has grown sufficiently, the string breaks up by creating a new pair $q_1\bar{q}_1$ from the vacuum. As a result, the system splits into two color-singlet systems $q\bar{q}_1$ and $q_1\bar{q}$. In the Lund string model, this break-up process proceeds as long as the invariant mass of the string pieces exceeds the on-shell mass of a hadron. Hence, each produced hadron corresponds to a small piece of the string with a quark and an antiquark at its ends. Thus, the fragmentation process is described in terms of branchings ("string ⇒ hadron + string remainder" and so on) and may be calculated iteratively.

The Lund model uses the idea of quantum tunneling in order to create a quark-antiquark pair $q_i\bar{q}_i$, which move apart along the z axis. The field energy between the quark-antiquark pair can be transformed into the sum $m_\perp$ of the two transverse masses (the transverse mass squared $m^2_{\perp q} = m^2_q + p^2_{\perp q}$). The tunneling mechanism is assumed to give a transverse momentum $p_\perp$ for each new created pair $q_i\bar{q}_i$. The transverse momentum $p_\perp$ is locally compensated between q_i and $\bar{q}_i$ (since the string is assumed to have no transverse excitations), and is given a Gaussian distribution in p_x and p_y separately. The $p_\perp$ of a produced meson (hadron) $q_{i-1}\bar{q}_i$ is obtained as the vector sum of the $p_\perp$'s of the q_{i-1} and $\bar{q}_i$.

For each breakup vertex the $q_i\bar{q}_i$ pairs are created according to the tunneling probability, leading to a Gaussian suppression

$$Prob(m^2_q, p^2_{\perp q}) \propto \exp\left(-\frac{\pi m^2_q}{\kappa}\right) \exp\left(-\frac{\pi p^2_{\perp q}}{\kappa}\right) = \exp\left(-\frac{\pi(m^2_q + p^2_{\perp q})}{\kappa}\right),$$

where m_q is the mass of the produced quark flavor, $p^2_{\perp q} = p^2_x + p^2_y$, $p_{\perp q}$ is the transverse momentum imparted by the breakup process (the $\bar{q}$ has the same mass and opposite $p_\perp$), and $\kappa \approx 1$ GeV/fm is the string tension.

With $p_\perp^2$ and m^2 fixed, the final step is to select the fraction z of the quark's longitudinal momentum that is carried by the created meson or hadron. The fragmentation function $f(z)$ is the probability that a given z is picked. PYTHIA uses the following expression for the Lund symmetric string fragmentation function [9, 22]

$$f(z) \propto z^{-1}(1-z)^a \exp\left(-\frac{b(m_h^2 + p_{\perp h}^2)}{z}\right). \tag{4.4}$$

This function has two phenomenological free parameters a, b, which need to be consistent to lead the fragmentation into accordance with the experimental data. In PYTHIA, the parameters PARJ(41) and PARJ(42) correspond to a, b, and their default values are set to PARJ(41)$= 0.3$ and PARJ(42)$= 0.58$ GeV^{-2}.

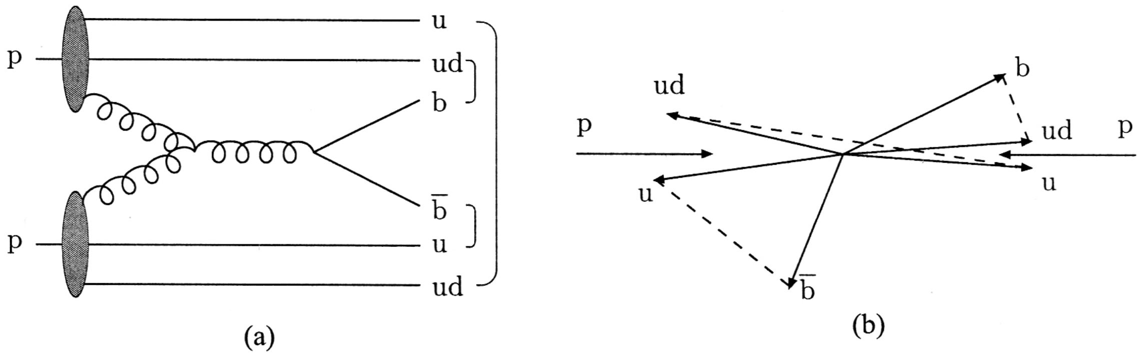


Figure 4.3: Example of a string configuration in a pp collision.(a) Graph of the process, with brackets denoting the final colour singlet subsystems. (b) Corresponding momentum space picture, with dashed lines denoting the strings [37].

It is assumed, that baryons can be produced in a similar way (though this is still a poorly studied area). A diquark is considered as a color anti-triplet, behaving like an ordinary antiquark, and the string can break by producing a quark-antiquark or a diquark-antidiquark pair. Thus, an event can contain several separate strings. Fig. 4.3 presents an example of a string configuration for a proton-proton collision. Representing a remnant of the proton as a ud diquark and a u quark (or uu and d) leads to three strings $ud-b$, $u-\bar{b}$ and $ud-u$. Also, because of the symmetry, we can expect that the b quark could couple with the ud diquark of the other proton.

The next step is the assignment of a spin S and an internal orbital angular momentum L to the produced quarks from which the hadron multiplets are formed. This sector contains the largest amount of free parameters [9] which are determined from data.

The production mechanism of heavy quark pairs ($b\bar{b}$, $c\bar{c}$) in proton-proton collisions in PYTHIA is influenced by the three following processes:

- Pair creation by quark-antiquark annihilation ($q\bar{q} \to b\bar{b}$) and gluon fusion ($gg \to b\bar{b}$).
- Flavor excitation. This case is described as $bq \to bq$ or $bg \to bg$, when a virtual b quark from one beam proton is put on mass shell by scattering with a parton from the other beam proton.
- Gluon splitting. In this case the $b\bar{b}$ pair is created when a $g \to b\bar{b}$ branching takes place in the initial- or final-state shower, instead of the hard scattering.

it is possible to influence the simulation process by changing the generator's parameters. These three processes give contributions to the total cross-section as shown in Table 4.1.

Production process	2 TeV	7 TeV	14 TeV
1. Pair creation	2.7	15.0	34.3
2. Flavour excitation	7.7	44.7	104.6
3. Gluon splitting	1.5	19.4	71.2
Total	11.9	79.1	210.1

Table 4.1: Results for B production cross-sections (μb) for b-quarks simulated with PYTHIA 6.4 ($p_T > 10$ GeV/c, $|\eta| < 2.5$) [38].

After a pair of protons collided, the produced B mesons are emitted in the forward or backward cones, in the directions of the colliding particle beams. The boost of the $b\bar{b}$ pairs comes from the difference of momenta of the partons which are involved in the scattering process: since each of the partons carries a fraction of the total proton momentum, combinations with higher momentum ratio become possible with the growth of the centre-of-mass energy, resulting in a large boost of the $b\bar{b}$ pair in the detector frame towards higher $\sqrt{s}$ values.

PYTHIA also simulates multiple parton interactions: the other partons of the two colliding protons may cause additional (softer) parton-parton interactions and such underlying events contribute to the observed particle multiplicity.

The final stage of the event simulation is the hadronization of the colored quarks and gluons to mesons and baryons.

Decay simulation. Many particles produced by fragmentation are unstable and decay. The new B physics experiments require a reasonable description of the decay of heavy mesons. The LHCb software framework relies on the EvtGen particle decay simulation package for the simulation of physics processes, related to decays of B mesons. The package uses a set of tables where branching ratios and decay modes are stored, and correctly simulates the angle and time-dependent distributions in sequential B decays using the actual decay amplitudes [39].

4.3 Detector simulation

After an event has been generated, it is necessary to model the propagation of the produced particles through the detector. In the case of the LHCb detector this modeling handles the following three major physical processes:

- Propagation of particles through matter and simulation of secondary interactions.
- Bending of the charged particles in magnetic field of the detector's magnet.
- Decay of the remaining unstable particles.
- Simulation of detector response.

These tasks are implemented by a set of algorithms, incorporated in the GEANT software package [40], execution of which is controlled by the GAUSS application. The algorithms of GEANT describe a large number of physical processes and cover a wide energy range of many types of particles. The output of the GEANT simulation are sets of hits which represent the interactions of particles with the sensitive parts of the detector. This output is then used by the BOOLE application during the digitization phase.

GEANT also simulates the propagation of charged particles in the magnetic field. In the most simple case in a uniform magnetic field in vacuum, the trajectories of charged particles are represented by helixes, but in the case of LHCb, the magnetic field is not uniform and GEANT uses the Runge-Kutta [41] method to properly describe the particles' movement. Figure 4.4 illustrates a GEANT simulation event with Monte Carlo particles propagating the LHCb tracking system. It is clearly visible that the trajectories of the particles are bent in the region of the LHCb magnet and the cross-shaped markers represent hits in the sensitive material of the tracking detectors.

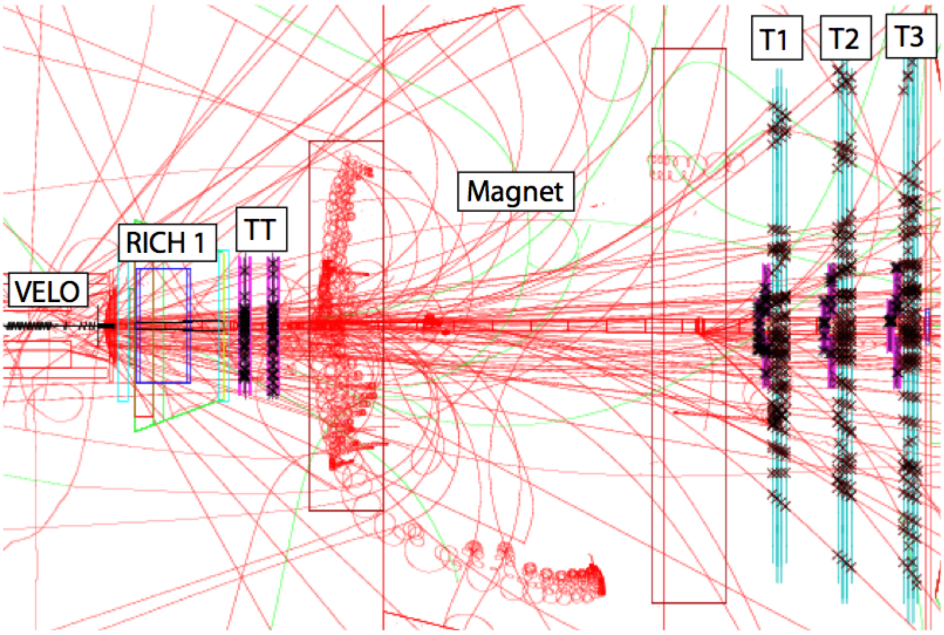


Figure 4.4: GEANT graphical representation of the trajectories of charged particles in the LHCb tracking system [1].

Another important aspect of GEANT operation is the modeling of propagation of particles through material of the detector and their energy loss. When particles traverse through matter, they interact with its atoms. Most of these interactions are electromagnetic and affect only the charged particles. GEANT is able to simulate different phenomena, depending on how close the given particle passes the atom of the matter it is traversing, such as atomic ionization and excitation, interaction of the particle with an individual electron of the atom, Coulomb scattering and multiple scattering, nuclear interaction.

4.4 Overview of the LHCb software

The LHCb software framework is used to work with the data collected by the detector and to simulate the detector's response. The main part of the object oriented framework, called GAUDI [42], was developed in C++, though some of the applications on which it relies were coded in Fortran and the most recent interfaces are developed in Python. GAUDI provides basic functionality, such as data access and analysis, computational job steering, logging, etc. Separate applications, which form the framework, are responsible for specific tasks, such as event generation and visualization, simulation of the detector response, data reconstruction, physics analysis. Below in Fig. 4.5 is a brief description of these tasks and applications which provide the respective functionality, organized in the sequence as they are executed in the process of modeling of the LHCb detector.

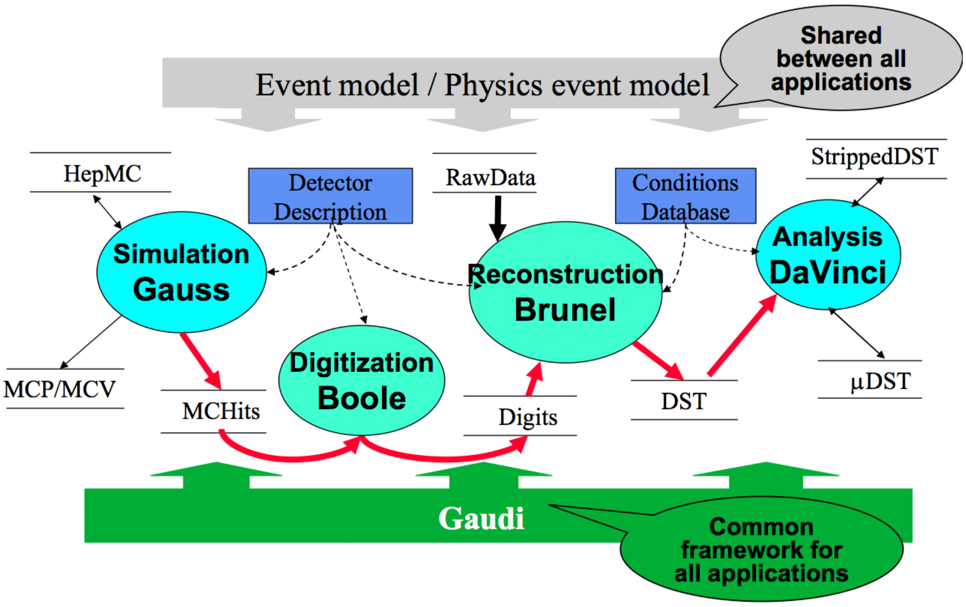


Figure 4.5: The main LHCb applications: Gauss (simulation), Boole (digitization), Brunel (reconstruction) and DaVinci (physics analysis) [43].

Simulation is the first step in which events are generated. It is handled by the GAUSS application [44] which delegates the simulation of the *pp* interactions to the external Monte Carlo generator program PYTHIA [9]. PYTHIA output consists of the products of the collision which are represented by their four-momentum vectors and space points of production. The decay of the *B* hadrons is controlled by a separate application – EvtGen [45]. The collected output of the generation phase is then handled by the GEANT4 toolkit [40], which simulates the evolution of the particles in the detector: propagation of charged particles in the magnetic field, interaction with material inside the detector, decay of unstable particles, etc.

Digitization is the next major phase in the process, and is used to simulate the response of the detector electronics. The BOOLE application [46] has dedicated algorithms which describe each subdetector response and is responsible for digitizing the hits of the particles in the sensitive material of the detector, presenting them as electrical signals.

Trigger is the intermediate step between the digitization and the offline reconstruction (though, in the Monte Carlo tuning studies this step could be executed after the reconstruction phase). This is done by the MOORE program [47]. In Fig. 4.5, the process would sit anywhere past the digitization process and prior to the execution of actual physics analysis.

Reconstruction is the final step in preparation of the data for analysis, which is executed by the BRUNEL program [48]. Its main tasks are the track reconstruction and particle identification.

Physics analysis: the main tool for the LHCb scientific community is the DaVinci application [49] which allows to execute event selection algorithms. It uses the reconstructed information to compose particles from tracks and particle identification objects, search and fit vertices, combine the appropriate particles to select specific decays, apply cuts on different variables such as mass life time of a particle.

Visualization of the events in the detector can be done with the PANORAMIX application [50]. It translates the information on the events and detector geometry into the graphical overview of the event, which helps in understanding of the data and the detector behavior.

Chapter 5

Mathematical methods

Aut viam inveniam, aut faciam.

Lat.: I'll either find a way or make one.

This chapter defines the problem of this work (Section 5.1) and discusses methods which were used to solve it and which were implemented in the `Lagrange` application. Section 5.2 gives an overview of the tuning methods, followed by Section 5.3 which explains the parametrized response function. Section 5.4 describes the fitting of the response function, and Section 5.5 discusses the fitting algorithm. Sections 5.6 and 5.7 describe the χ^2 function and its minimization, respectively. Finally, Section 5.8 discusses the sources of tuning uncertainties.

5.1 Brief introduction into the problem

One of the most common problems of particle physics is the necessity of comparisons between experimental data (i.e. measurements of final state observables, produced in collisions) and theory, represented by Monte Carlo simulations.

Physics models contain miscellaneous parameters, which are essential for the description of physical processes, and as was discussed in the previous chapter, some phenomenological models contain parameters which must be tuned to data [51] (see tables of parameters in [9,51,52]). This leads to the fact that Monte Carlo event generators contain a number of free parameters, which have to be tuned to best describe the experimental data. Thus, in order to obtain the Monte Carlo event generator predictions, which would agree with the experimental data, it is necessary to simultaneously tune many phenomenological parameters in a high-dimensional space, and the real challenge behind this task is the requirement to perform it with affordable computing time costs.

Unfortunately, the MC generator is not a single analytic formula and the minimization of the fit function with many parameters requires multiple runs of the generator with large data samples (high number of events per run). Even though the single collision events in particle physics are independent of the other events in the same data sample, meaning the procedure of event generation is trivially parallelized, such a task, depending on the number of simultaneously tuned parameters and the size of data samples, could require many days or even months.

All of this made it essential to find a new and faster way of solving the problem, which naturally lead to the idea of the bin-wise interpolation of the Monte Carlo event generator response (also known as the DELPHI approach), which was implemented in 1995 as a Fortran application [53,54], and as a more modern example - the Professor tool package [52,55], developed in Python. The core idea of the approach is in creating an analytical model of the Monte Carlo event generator for fitting to the experimental data.

This work presents a certain extension and improvement of the described technique, further developed and implemented in the application `Lagrange`, which together with its helper-programs and the LHCb framework (refer to the Chapter 4) forms a single system for tuning model parameters of the Monte Carlo event generators to experimental data. The `Lagrange` application is based on parameterizing the response of Monte Carlo generator by a polynomial; it was developed in C++ with the use of ROOT, the object oriented framework for data analysis.

Some of the key features of the `Lagrange` application include:

- Actively developed interpolation idea:
  1. interpolation by a polynomial of the 2nd or 3rd order,
  2. computation of the unknown polynomial coefficients by two different numerical implementations, using the method of least-squares,
  3. guaranteed estimation of computational errors,
  4. estimation of statistical errors via covariance matrices,
  5. ability to emphasize or to exclude individual bins or distributions by assigning weights,
  6. simultaneous use of multiple distributions, which serve as the comparison criteria;
- integration of the application into the LHCb software framework.

The `Lagrange` application and its helper programs are tools for systematic tuning of event generator parameters, and in this work they have been applied for optimization of the PYTHIA 6.4 event generator, with respect to experimental data, collected by the LHCb detector. The results are described in Chapter 8 of this thesis.

The sections below describe in detail the parameterization of the "cost functions" (the bin values in the Monte Carlo generator histograms).

5.2 The tuning method

The long history of efforts [53, 54] to create an automated and systematic approach of tuning generator parameters to data led to the idea of parameterizing the generator behavior, and eventually to the introduction of the *parameterization-based tune* [51, 52].

The essence of the method is the following. The true fit function is expected to be complicated, not analytic and a very "computationally expensive function" of an unknown form, for technical reasons consisting of large sets of bins b in histograms of many distributions. In order for the problem to be solved, the behavior of the cost function is simulated by inexpensive ones. The true fit function is represented by a set of functions $f_b(\vec{p})$ and every function $f_b(\vec{p})$ (a polynomial) is fitted to the Monte Carlo generator response $MC_b(\vec{p})$ of each bin b – to the changes in the n-dimensional parameter vector $\vec{p} = (p_1, p_2, \dots p_n)$. After that, the only thing remaining is to define a goodness of fit function between the generated and the experimental (reference) data and to minimize it. The result of the minimization is the predicted set of optimal parameters $\vec{p}_{\text{best}}$.

The further development of the parameterization-based optimization method was implemented in the `Lagrange` application and is described onwards in detail:

1 Choice of the parameterization function;

2 The algorithm of fitting the parameterization function to the MC generator response;

3 Goodness of fit function;

4 Minimization.

5.3 The parameterized response function

As was noted in the previous section, it is possible to fit a polynomial to the MC generator response MC_b in each observable bin b in every distribution, with a variable n parameter vector $\vec{p} = (p_1, p_2, \ldots, p_n)$, i.e. to define the function $f_b(\vec{p})$ that simulates the real MC generator response $MC_b(\vec{p})$ of each observable bin b in the n-dimensional parameter space.

Recall that an n-dimensional Taylor series [56] is

$$f(p_1, p_2, \ldots, p_n) = f(p_{0i}) + \sum_{i=1}^{n} \left.\frac{\partial f}{\partial p_i}\right|_{p_{0i}} \cdot (p_i - p_{0i}) + \frac{1}{2!}\sum_{i=1}^{n}\sum_{j=1}^{n} \left.\frac{\partial^2 f}{\partial p_i \partial p_j}\right|_{p_{0i}} \cdot (p_i - p_{0i})(p_j - p_{0j}) +$$

$$+\frac{1}{3!}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \left.\frac{\partial^3 f}{\partial p_i \partial p_j \partial p_k}\right|_{p_{0i}} \cdot (p_i - p_{0i})(p_j - p_{0j})(p_k - p_{0k}) + \cdots$$

With the following notations

$$\alpha_0 = f(p_{0i}), \quad \beta_i = \left.\frac{\partial f}{\partial p_i}\right|_{p_{0i}}, \quad \gamma_{ij} = \left.\frac{\partial^2 f}{\partial p_i \partial p_j}\right|_{p_{0i}}, \quad \delta_{ijk} = \left.\frac{\partial^3 f}{\partial p_i \partial p_j \partial p_k}\right|_{p_{0i}},$$

the expression above can be written as

$$f(\vec{p}) = \alpha_0 + \sum_{i=1}^{n} \beta_i (p_i - p_{0i}) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_{ij}(p_i - p_{0i})(p_j - p_{0j}) +$$

$$+\frac{1}{6}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \delta_{ijk}(p_i - p_{0i})(p_j - p_{0j})(p_k - p_{0k}) + \cdots$$

Hence, an n-dimensional second-order polynomial, that is usually used to account for lowest-order is

$$MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p}) = \alpha_0^{(b)} + \sum_{i=1}^{n} \beta_i^{(b)} p_i' + \sum_{i=1}^{n}\sum_{j=i}^{n} \gamma_{ij}^{(b)} p_i' p_j' \tag{5.1}$$

Here it is taken into account, that $p_i'p_j' = p_j'p_i'$ and the shifted parameter vector $\vec{p}\,' = \vec{p} - \vec{p}_0$, where

$$\vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}, \qquad \vec{p}_0 = \begin{pmatrix} p_{01} \\ p_{02} \\ \vdots \\ p_{0n} \end{pmatrix},$$

with $\vec{p}_0$ chosen as the center of the hypercube of chosen parameter values. The elements of the vector $\vec{p}$ are determined at random positions from the volume of the parameter hypercube, taking into account the range of values of the parameters $[\vec{p}_{\text{min}}, \vec{p}_{\text{max}}]$. Here, the choice of $\vec{p}_0$ is irrelevant [55]: the result of a shift of the central point is simply a redefinition of the polynomial coefficients, but the function remains the same.

The minimal number of independent coefficients $\gamma_{ij}^{(b)}$ in the double sum is calculated on the basis of the combinatorics theorem, which proves that the total number of different combinations when choosing m elements out of n, permutations without repetition, is equal to C_{n+m-1}^{m}. Thus, in the case of the 2^{nd} order polynomial ($m = 2$)

$$C_{n+m-1}^{m} = \frac{(n+m-1)!}{m!(n-1)!} = C_{n+1}^{2} = \frac{(n+1)!}{2!(n-1)!} = \frac{n(n+1)}{2}$$

If necessary, it is possible to use a 3^{rd} order polynomial

$$MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p}) = \alpha_0^{(b)} + \sum_{i=1}^{n} \beta_i^{(b)} p_i' + \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma_{ij}^{(b)} p_i' p_j' + \sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} \delta_{ijk}^{(b)} p_i' p_j' p_k' \quad (5.2)$$

In this case, the minimal number of the coefficients $\delta_{ijk}^{(b)}$ in the triple sum with $m = 3$ equals to

$$C_{n+m-1}^{m} = \frac{(n+m-1)!}{m!(n-1)!} = C_{n+2}^{3} = \frac{(n+2)!}{3!(n-1)!} = \frac{n(n+1)(n+2)}{6}$$

If we know the coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}, \delta_{ijk}^{(b)}$, we can use the formula (5.1) or (5.2) to compute the events $MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p})$. The number of unknown polynomial coefficients in formulas (5.1), (5.2) for n parameters is equal to

$$N_{\text{min}} = 1 + n + \frac{n(n+1)}{2} + \underbrace{\frac{n(n+1)(n+2)}{6}}_{\text{for the 3}^{\text{rd}}\text{ order only}} . \quad (5.3)$$

Table 5.1 presents the values of N_{min}, depending on the number of parameters n for polynomials of 2^{nd} and 3^{rd} order.

n (p_i)	N_{min} (2^{nd} order)	N_{min} (3^{rd} order)
1	3	4
2	6	10
3	10	20
4	15	35
5	21	56
6	28	84
7	36	120
8	45	165
9	55	220
10	66	286

Table 5.1: Values N_{min}, depending on the number of parameters n, for polynomials of 2^{nd} and 3^{rd} order.

Taking all of this into consideration, it is possible to derive a general formula for the number of independent coefficients $N_{\min}$, depending on the number of parameters n and the order of the polynomial m

$$N_{\min}(n,m) = 1 + C_n^1 + C_{n+1}^2 + C_{n+2}^3 + \cdots + C_{n+m-1}^m =$$
$$= 1 + \frac{1}{1!}n + \frac{1}{2!}n(n+1) + \frac{1}{3!}n(n+1)(n+2) + \cdots$$
$$\cdots + \frac{1}{m!}n(n+1)(n+2)\cdots(n+m) = 1 + \sum_{i=1}^{m} \frac{1}{i!} \prod_{j=0}^{i-1}(n+j).$$

5.4 Fitting the response function

In order to use the parameterized response function f_b (5.1) in the place of the Monte Carlo generator, it is necessary to calculate the unknown coefficients of the polynomial for each bin b. For that, we initially define the hypercube in the n-dimensional parameter space, i.e. for each parameter we define the boundaries in which it may vary, so that every parameter p_i has lower and upper limits $p_{i\,\min}, p_{i\,\max}$ (this interval contains reasonable values). Here, it is necessary to understand that the parameter range of values should not be too large, in order to have smaller parameter space volume and to create confidence that the approximation (5.1) or (5.2) gives a good fit to the real Monte Carlo generator response.

The next step is to define the system of linear equations, using the formula (5.1) or (5.2) with N sets of randomly chosen values of n parameters $p_1, p_2, \ldots, p_n$ (N points in the parameter hypercube $[\vec{p}_{\min}, \vec{p}_{\max}]$), determining $n_b(\vec{p})$ as N initial runs of the Monte Carlo generator.

In the case when $N = N_{\min}$ (the number of the initial runs is the same as the number of the unknown coefficients), the system of equations forms a square matrix which can be solved in the standard way (gaussian elimination), and we have the “exact” fit of the polynomial to the generator response. However, in this case there is a systematic uncertainty involved, since the real Monte Carlo response is not an exact polynomial. Also it is necessary to consider the existence of statistical errors of the MC generator.

In the case when the number of initial runs is larger than the number of the unknown coefficients ($N > N_{\min}$), the system of equations becomes overdetermined, with a non-square matrix. There are several well known ways to solve such a system and one of them is *pseudoinverse* [57,58], which gives a least-squares best fit to the oversampled generator points. Hence, with pseudoinverse the overdetermined system is usually solved by two methods (see Section 6.1 in Chapter 6): via solving a so-called normal system, or via singular value decomposition (SVD).

The first method, solving via the normal system, is considered the fastest (computationally lighter), but less precise, thus it is better to keep track of computational errors. The method of guaranteed estimation of accuracy of computation of the coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$ and $\delta_{ijk}^{(b)}$ (when the system is solved via a normal system) is described in detail in Section 6.1 of Chapter 6. The errors of coefficients permit to estimate the accuracy of computing the approximated Monte Carlo generator response $f_b(\vec{p})$ by Formulas (5.1) or (5.2).

Similarly, the estimation of statistical errors for the coefficients and for the parameterized response function $f_b(\vec{p})$ is discussed in Section 6.3 of Chapter 6.

The calculated coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$ and $\delta_{ijk}^{(b)}$ allow to compute the approximated Monte Carlo generator response (the parameterized response function) $MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p})$ by the formula, allowing to describe the parameter hyperspace analytically. Hence, it is possible to say that the parameterization (5.1), (5.2) gives us **a fast, analytical Monte Carlo “pseudo-generator”**.

5.5 The fitting algorithm

1. Define a hypercube.

The first step is to define the boundaries of a hypercube in the n-dimensional parameter space, i.e. based on physics reasons, from previous experiments, and theoretical predictions, to define the boundaries for each of the tuned parameters $[p_{i\,\mathrm{min}}, p_{i\,\mathrm{max}}]$, while at the same time trying to retain the size of the hypercube, to get a good fit of the parabolic approximation to the real Monte Carlo generator response (see Tab. 5.2).

Parameter	minimum value	maximum value
p_1	$p_{1\,\mathrm{min}}$	$p_{1\,\mathrm{max}}$
p_2	$p_{2\,\mathrm{min}}$	$p_{2\,\mathrm{max}}$
$\vdots$	$\vdots$	$\vdots$
p_n	$p_{n\,\mathrm{min}}$	$p_{n\,\mathrm{max}}$

Table 5.2: Definition of the range of values of the parameters.

After the boundaries $[\vec{p}_{\mathrm{min}}, \vec{p}_{\mathrm{max}}]$ have been set, the next step is to define the center $\vec{p}_0$ of the parameter hypercube

$$\vec{p}_0 = \vec{p}_{\mathrm{min}} + (\vec{p}_{max} - \vec{p}_{\mathrm{min}})/2$$

2. Choice of values of the tuned parameters.

In this step, it is necessary to generate N parameter vectors

$$\vec{p}(\mathrm{set}\ i) = (p_1, p_2, \ldots, p_n)^T_{\mathrm{set}\ i}, \quad i = 1, 2, \ldots, N$$

in the parameter hypercube (see Fig. 5.2), on the condition that $N \geq N_{\mathrm{min}}$ (see Tab. 5.3): N points in the n-dimensional parameter space.

The question is how to choose the values for every set of $\vec{p}(\mathrm{set}\ i)$. It is technically impossible to sample the parameter space in every its point, so it is a common approach to choose a finite number of sampling points, such that the computational time would be acceptable. Since the form of the approximated true function is not known, it is a poor choice to base one's perameterization on a sampling along a straight line. The conventional strategy is to select the coordinates of the points such that they cover the sampled space uniformly. For that one usually uses a pseudorandom number generator, the algorithm of which produces sequences which are uniformly distributed. In addition to this conventional approach, the `Lagrange` application also allows one to use the Sobol quasi-random sequence to set the sampling points. Fig. 5.1 presents the comparison of coverage of a 2D space with a pseudorandom number generator and the Sobol sequence. It is seen that in the case of the Sobol sequence the space is covered more homogeneously.

set 1	set 2	$\cdots$	set N
p_1(set 1)	p_1(set 2)	$\cdots$	p_1(set N)
p_2(set 1)	p_2(set 2)	$\cdots$	p_2(set 1)
$\vdots$	$\vdots$	$\cdots$	$\vdots$
p_n(set 1)	p_n(set 2)	$\cdots$	p_n(set N)

Table 5.3: N sets of random values of n tuned parameters.

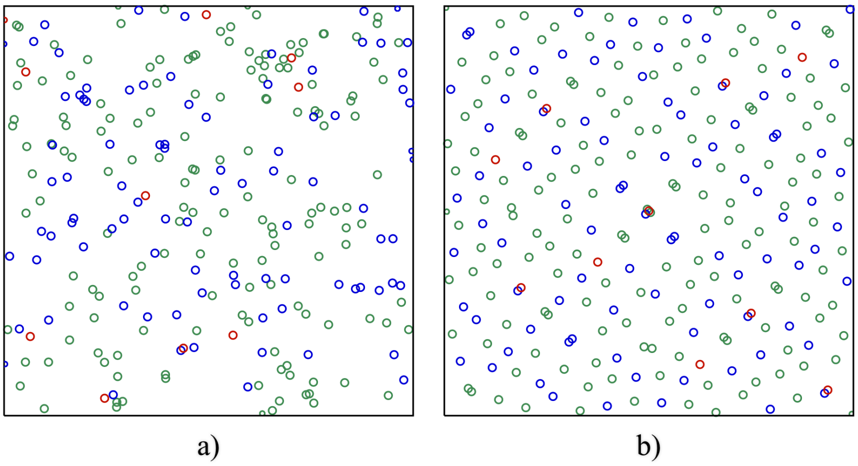


Figure 5.1: 2D space covered with 256 points from a pseudorandom number generator (a)) compared with the first 256 points from the Sobol sequence (b)). The red markers correspond to the first ten points, the blue ones are the points from 11 to 100, and green are from 101 to 256.

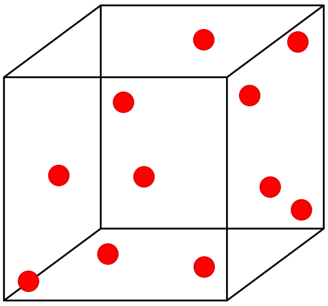


Figure 5.2: N parameter points in the 3-dimensional space.

3. The Monte Carlo generator runs.

Execute the Monte Carlo generator at each sampled point, fill the histograms of distributions for K bins (see Fig. 5.3)

$$MC_b(\vec{p}(set\ i)) = n_b(\vec{p}(set\ i)), \quad b = 1, 2\ldots, K, \quad i = 1, 2, \ldots, N$$

and construct the matrix $V(N, K)$

$$V = \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ n_1(\text{set1}) & n_2(\text{set1}) & \cdots & n_K(\text{set1}) \\ n_1(\text{set2}) & n_2(\text{set2}) & \cdots & n_K(\text{set2}) \\ \vdots & \vdots & \vdots & \vdots \\ n_1(\text{set}N) & n_2(\text{set}N) & \cdots & n_K(\text{set}N) \end{pmatrix},$$

each row of which corresponds to the histogram of the Monte Carlo generator run for the specific vector of parameters. Also, estimate the statistical errors for the numbers of events n_b, generated in each bin b.

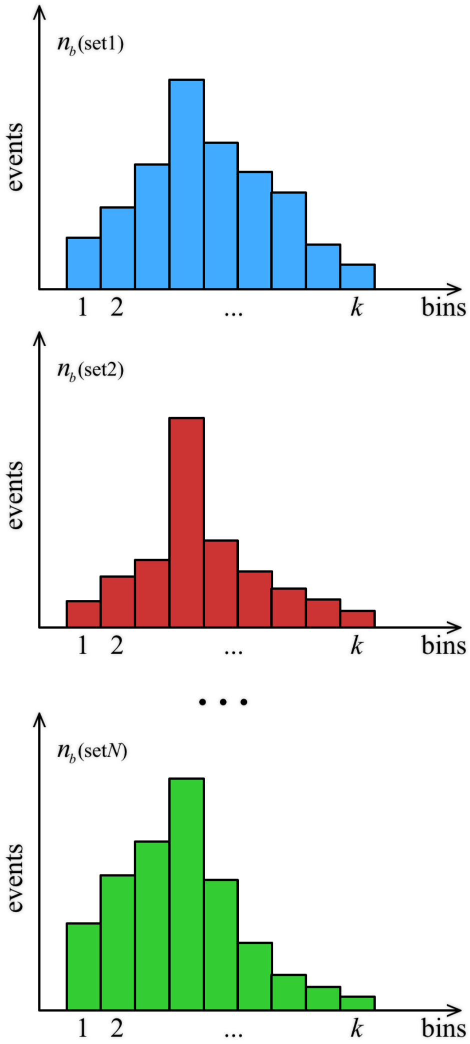


Figure 5.3: The Monte Carlo generator histograms for N points in the n-dimensional space.

4. **Computation of the unknown polynomial coefficients in Formula (5.1).**
   Fit the polynomial (5.1) to the N sampled points in the hypercube for each bin b.

   In order to compute the unknown polynomial coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$, the following system of N linear equations is constructed: Equation (5.1) is used to relate the data for bin b to each of the randomly chosen parameter vectors $\vec{p}(set\ i)$, using column b from the matrix V

$$n_b(\vec{p}(set\ 1)) = \alpha_0^{(b)} + \sum_{i=1}^{n} \beta_i^{(b)} \cdot p'_i(set\ 1) + \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma_{ij}^{(b)} \cdot p'_i(set\ 1) \cdot p'_j(set\ 1)$$

$$n_b(\vec{p}(set\ 2)) = \alpha_0^{(b)} + \sum_{i=1}^{n} \beta_i^{(b)} \cdot p'_i(set\ 2) + \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma_{ij}^{(b)} \cdot p'_i(set\ 2) \cdot p'_j(set\ 2) \qquad (5.4)$$

$$\vdots$$

$$n_b(\vec{p}(set\ N)) = \alpha_0^{(b)} + \sum_{i=1}^{n} \beta_i^{(b)} \cdot p'_i(set\ N) + \sum_{i=1}^{n}\sum_{j=i}^{n} \gamma_{ij}^{(b)} \cdot p'_i(set\ N) \cdot p'_j(set\ N)$$

where $\vec{p'} = \vec{p} - \vec{p}_0$.

In case of a square matrix, the solution of the system is simple and straightforward, whereas in the case of the non-square matrix the system is solved by using the pseudoinverse with the help of the least-squares method, which determines the best polynomial coefficients for each bin b.

The use of covariance matrices allows for an estimation of statistical errors of coefficients and events f_b computed by the formula (5.1), and in the case of use of pseudoinversion, computational errors are also estimated.

5. **Estimation of the optimal parameter vector $\vec{p}_{\text{best}}$.**
Having the estimated parameterization and the experimental data, it is now possible to fit the tuned parameters $p_1, p_2, \ldots, p_n$.

   The numerical minimization of the χ^2 function

$$\chi^2(\vec{p}) = \sum_{\text{bins}} \frac{(polynomial(\vec{p}) - data)^2}{error^2} = \min \quad \Longrightarrow \quad \vec{p}_{\text{best}} \tag{5.5}$$

   gives the vector of the tuned parameters $\vec{p}_{\text{best}} = (p_1, p_2, \ldots p_n)^T_{\text{best}}$, i.e. the predicted optimal set of the parameters.

6. **Comparison of the prediction with the experimental data.**
It is essential to test the new estimated set of the values of the tuned parameters to understand whether the prediction makes sense and properly tunes the Monte Carlo generator to better represent the experimental data. For that it is necessary to generate Monte Carlo events with the new estimated parameter values $\vec{p}_{\text{best}}$ and to compare the result to the experimental data, taking into account the estimated errors.

7. **Checking the parameterization: line-scans in the parameter space.**
A line-scan in parameter hyperspace is a form of a sanity test. It allows one to compare the performance of the estimated formula with the response of the Monte Carlo generator and to check for the abnormal behavior of the former one.

   In this case, the strategy is to sample the parameters from a straight line within the parameter hyperspace, estimating $f_b(\vec{p})$ by the formula (5.1) and comparing the result to the corresponding Monte Carlo generator run $n_b(\vec{p})$, followed by the computation of the parameterized χ^2 function and its comparison with the χ^2 function of the Monte Carlo generator response.

   The `Lagrange` application implements two different line-scan tests. The first one allows to probe the area around the estimated minimum $\vec{p}_{\text{best}}$ of the approximated function. The new limits for the parameters are defined as $p^i_{\text{best}} \pm p^i_{\text{best}}\delta$, where δ is a user-defined value. and the scan is then conducted along the diagonal of this redefined hypercube (see Fig. 5.4). The second method scans along a hyper-line, which is constructed to pass via the two points, defined by the default $\vec{p}_{\text{default}}$ and minimized $\vec{p}_{\text{best}}$ vectors of parameters. In this case, $\vec{p}_A = \vec{p}_{\text{default}}$ serves as a starting point of the line-scan, the end point $\vec{p}_B$ is calculated as $\vec{p}_B = 2\vec{p}_{\text{best}} - \vec{p}_A$, and the coordinates of sampling points in between are calculated as $(\vec{p}_B - \vec{p}_A)/M$, where M is the number of steps along the line.

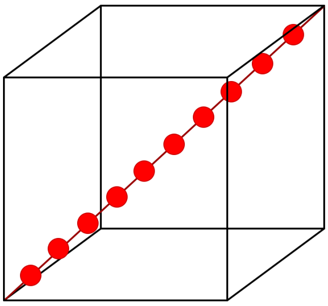
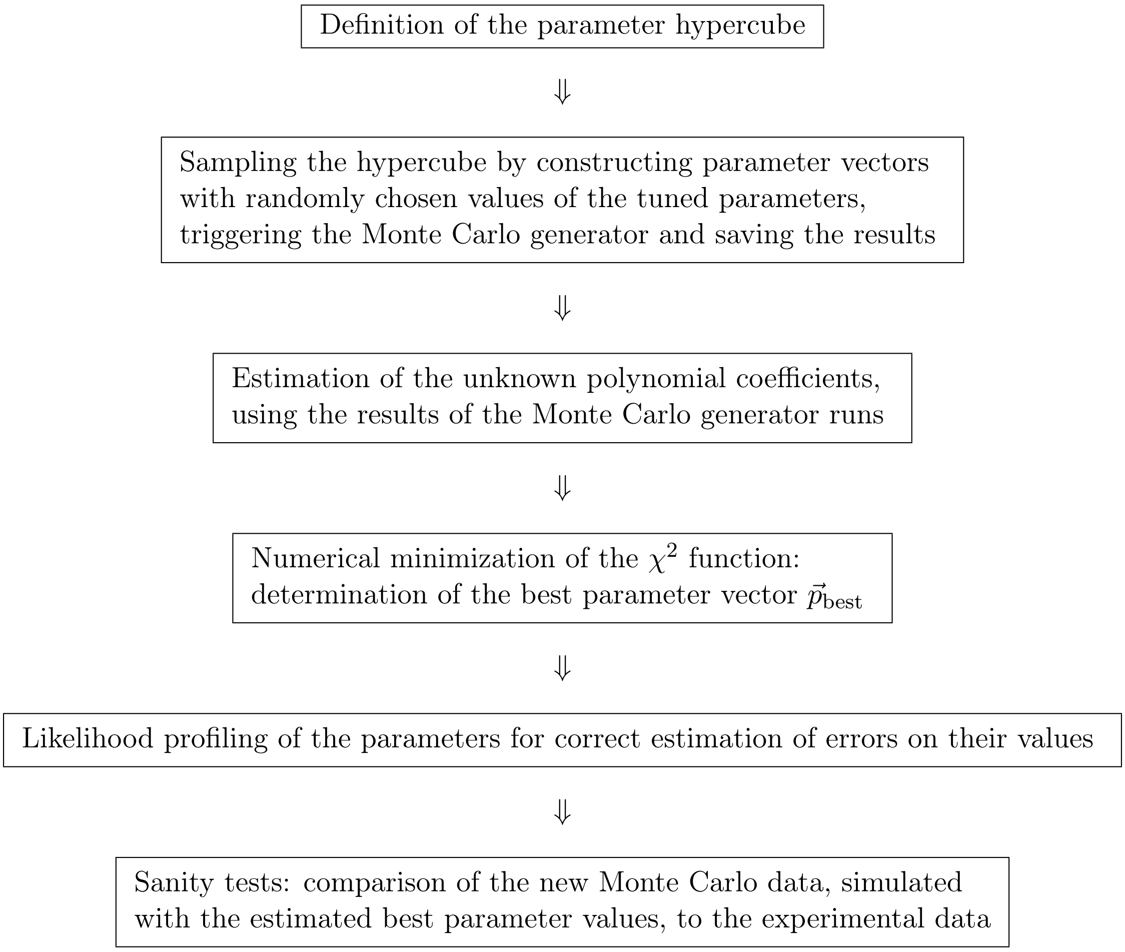


Figure 5.4: A line-scan in the parameter hyperspace.

The algorithm described above could be in short represented by the following scheme:



Definition of the parameter hypercube

⇓

Sampling the hypercube by constructing parameter vectors with randomly chosen values of the tuned parameters, triggering the Monte Carlo generator and saving the results

⇓

Estimation of the unknown polynomial coefficients, using the results of the Monte Carlo generator runs

⇓

Numerical minimization of the χ^2 function: determination of the best parameter vector $\vec{p}_{\text{best}}$

⇓

Likelihood profiling of the parameters for correct estimation of errors on their values

⇓

Sanity tests: comparison of the new Monte Carlo data, simulated with the estimated best parameter values, to the experimental data

5.6 The χ^2 function

After estimating the parameterization of the Monte Carlo generator response for each bin b, it is necessary to choose a goodness of fit function, compute and minimize it.

The χ^2 function (5.5) is usually used for this purpose and we can rewrite this expression taking into consideration (5.1) and introducing weights w_b for each bin b

$$\chi^2(\vec{p}) = \sum_{b=1}^{K} w_b \frac{(\tilde{f}_b(\vec{p}) - R_b)^2}{\Delta_b^2} \tag{5.6}$$

Here, K corresponds to the total number of bins, $\tilde{f}_b(\vec{p})$ is the value computed by Formula (5.1) for bin b, R_b is the experimental value for bin b, the total error Δ_b^2 is the sum in quadrature of the experimental errors, the computing and statistical errors of the simulation.

The use of weights w_b permits us:

- to emphasise individual observations or certain distributions;
- to emphasise individual bins of an observation;
- to exclude individual bins or certain distributions from the minimization.

The best set of values of the tuned parameters is determined via the minimization of the χ^2 function (5.6). For example, in the most simple case with only one tuned parameter p and a compared distribution, represented by a histogram with a single bin b, we have the situation illustrated in Fig. 5.5.

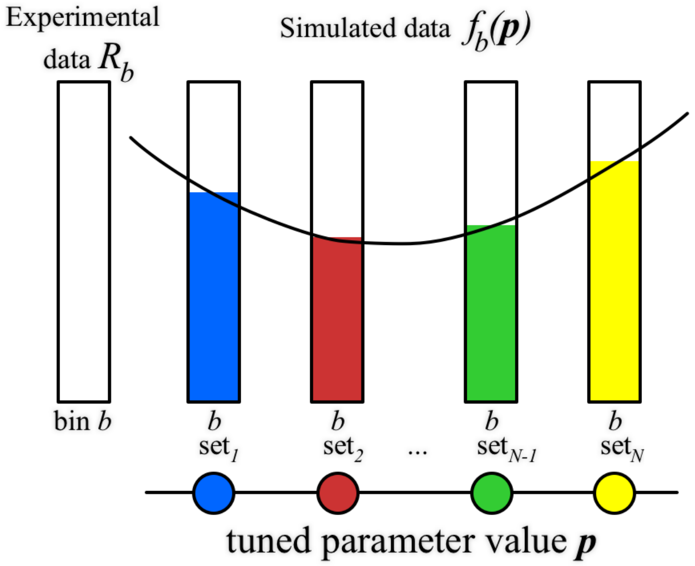


Figure 5.5: Simplified single-dimension, single-bin example, using N points for a fit.

$$f_b(p) = \alpha_0^{(b)} + \beta^{(b)} p + \gamma^{(b)} p^2 \quad \Longrightarrow \quad \chi^2(p) = \sum_b \left(\frac{f_b(p) - R_b}{\Delta_b} \right)^2 = \min \quad \Longrightarrow \quad p_{\text{best}}$$

Herewith, the estimated values $\vec{p}_{\text{best}} = (p_1, p_2, \cdots p_n)_{\text{best}}$ are called the least squares solutions, which are obtained by solving the system of n equations

$$\frac{\partial \chi^2(\vec{p})}{\partial \vec{p}} = 0$$

The variable χ^2 (5.6) coincides with the χ^2 probability distribution with N_{df} degrees of freedom.

In the given case, the number of degrees of freedom N_{df} is defined by the total number of bins K, the number of the tuned parameters n and the number of bins with zero weights (and all other bins with weights equal to one):

$$N_{\text{df}} = K - n - N_{w=0}$$

In general, the estimated $\vec{p}_{\text{best}} = (p_1, p_2, \cdots p_n)^T_{\text{best}}$ vectors are correlated [59,60]. The covariance matrix $C_{\vec{p}_{\text{best}}}$ is defined as follows

$$C_{\vec{p}_{\text{best}}} = 2H^{-1},$$

where

$$H_{ij} = \left(\frac{\partial^2 \chi^2}{\partial p_i \partial p_j} \right) |_{\vec{p}=\vec{p}_{\text{best}}}$$

are the elements of the symmetrical matrix of second derivatives (the Hesse matrix) of the minimized χ^2 function.

The square roots of the values of the main diagonal of the covariance matrix $C_{\vec{p}_{\text{best}}}$ are the errors for the estimated parameters values.

$$\Delta p_i = \sqrt{c_{ii}}.$$

5.7 Minimization

The final step of the fitting algorithm is the minimization of the parameterized χ^2 function (5.6). The choice of the minimization algorithm is important because the parameterized χ^2 function is a very complex function, therefore numerical minimization is usually used. In the case of this work, the well known application `Minuit` [61] was used for the purpose of numerical minimization of the χ^2 function (5.6).

In order to avoid being stuck in a non-global minimum, multiple starting points are used as the input for the `Minuit` program during the minimization procedure.

The result of the numerical minimization is the vector of best values of the tuned parameters $\vec{p}_{\text{best}} = (p_1, p_2, \ldots, p_n)^T_{\text{best}}$, i.e. the predicted optimal set of parameters values with errors taken into account.

The `Minuit` application also estimates the uncertainties Δp_i of the optimal parameters values p_i, by computing the parameter points (the contour in the parameter space) in which the χ^2 exceeds the value of the $(\chi^2)_{\text{min}}$ in the minimum by one unit

$$\chi^2(\vec{p}) = \chi^2(\vec{p}_{\text{best}}) + 1 = (\chi^2)_{\text{min}} + 1$$

which corresponds to a ±1 standard deviation $\sigma_{\vec{p}_{\text{best}}}$ (68 % Confidence Level estimate) for a truly χ^2-distributed test statistic.

The computation of confidence intervals of theory parameters is one of the most important problems in experimental physics. The method used widely in high-energy physics is the extraction of confidence intervals by finding the points where the $-2\ln$ likelihood function increases by a factor defined by the required confidence level ('$\ln L + \frac{1}{2}$' method) [62].

The profile likelihood method (described in Chapter 8), is used for determining the asymmetric parameter errors and their comparison to the uncertainties, estimated by `Minuit`. In this method the many-dimensional likelihood function is reduced to a function that only depends on the parameter of interest [63].

5.8 Analysis of sources of tuning uncertainties

An important feature of the fitting algorithm is that it makes a qualitative estimate of the tune's robustness. Let us consider the various sources of uncertainties in order to understand which of them are computationally controllable and which are unquantifiable. The principal sources of tuning errors are:

1. Reference data (experimental data collected by the detector) uncertainties.

2. Statistical errors of the parameterization result, which depend on the simulated data size (number of events per simulated data sample).

3. Computational errors, arising from the accuracy of the interpolation of events in every bin b by the polynomial (related to the number of initial sampling points N).

4. Computational errors of the parameterization result.

5. Choice of the Goodness of Fit function (in this particular case, the χ^2 function).

6. Computational errors, which arise from the numerical minimization of the GoF (e.g., errors of the `Minuit` minimizer).

7. Systematic errors, which arise from the approximation of the event generator to the true fit function $f(\vec{p})$, by the set of functions $f_b(\vec{p})$.

These sources of uncertainties can be estimated qualitatively or even quantitatively. For example, the contribution of the uncertainties from source 4 can be reduced by increasing the number of sampling points N in the parameter hyperspace, such that it would exceed the number of the unknown polynomial coefficients $N > N^{(n)}_{\mathrm{min}}$.

Statistical uncertainties, which depend on the strategy of choosing the initial sampling points N in the parameter hyperspace, from which the parameterization is calculated, are a part of sources 3 and 7. Influence of these errors are estimated by repeating the tuning procedure using different algorithms of selecting initial sampling points in the parameter hypercube. Contributions of these errors could be reduced by rerunning the Monte Carlo generator multiple times for each initial sampling point, or simply increase the total number of simulated events per generated data sample. They can also be probed by overconstraining the fit.

Evidently, the influence of statistical uncertainties of source 3 grows smaller with increasing number of generated events in the simulated data samples.

The uncertainties of sources 1, 2, 4 and 6, which can be estimated numerically, are accounted for in the tuning procedure of the `Lagrange` tuning application.

Chapter 6

Computation of the parameterization

'If this were only cleared away',
they said,'it would be grand!'

by Lewis Carroll
"Through The Looking-Glass
And What Alice Found There"

This chapter focuses on a detailed discussion of the parameterization mechanisms. Section 6.1 describes the computation of the polynomial coefficients via a system of linear equations, followed by Section 6.2, which presents the method of calculating the Monte Carlo generator response f_b by Equation (5.1). Concluding this chapter, Section 6.3 describes the estimation of the computational and statistical errors.

6.1 Solving the system

This chapter contains a detailed description of the core method for the linear system solution and errors estimation.

6.1.1 N parameter sets, one bin b

This simplified example considers the case, when the system with n parameters is solved for a single bin b.

The following notations are used: n is the number of parameters $(p_1, p_2, \ldots, p_n)$, $N_{\min}$ is the number of unknown polynomial coefficients for n parameters (5.3) and $N \geq N_{\min}$ is the number of parameter sets (see Tab. 5.1).

To determine the unknown polynomial coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$ for bin b, it is necessary to find the solution of the system (5.4) with N sets of the randomly chosen values of n parameters $p_1, p_2, \ldots, p_n$.

It is possible to convert (5.4) to matrix form:

$$\begin{pmatrix} \text{column1} & \text{column2} & \cdots & \text{columnN}_{\text{min}} \\ 1 & p'_1(\text{set1}) & \cdots & p'_n p'_n(\text{set1}) \\ 1 & p'_1(\text{set2}) & \cdots & p'_n p'_n(\text{set2}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p'_1(\text{set}N) & \cdots & p'_n p'_n(\text{set}N) \end{pmatrix} \begin{pmatrix} \text{bin}b \\ \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix} = \begin{pmatrix} \text{bin}b \\ n_b(\text{set1}) \\ n_b(\text{set2}) \\ \vdots \\ n_b(\text{set}N) \end{pmatrix}, \qquad (6.1)$$

or

$$PC_b = V_b.$$

Here, P stands for the matrix of size (N, N_{min}), V_b corresponds to the right hand side vector of size N, and C_b is the vector of unknown polynomial coefficients of size N_{min}.

The components of vector V_b are the Monte Carlo generator response n_b in bin b for randomly chosen sets of parameter vector $\vec{p}$:

$$\vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}$$

and $\vec{p}\,' = \vec{p} - \vec{p}_0$.

1. If $N = N_{\text{min}}$, P is a square matrix of size $P(N_{\text{min}}, N_{\text{min}})$.

In this case, the solution of the system for bin b is the vector of coefficients C_b, which can be written as

$$C_b = P^{-1} V_b$$

2. If $N > N_{\text{min}}$, P is a non-square matrix of size $P(N, N_{\text{min}})$ and the system $PC_b = V_b$ is overdetermined.

In general, for such a case there is no exact solution, and it is only possible to obtain an approximate solution C_b with a minimal residual $PC_b - V_b = r$.

This is a *linear least squares problem*, and its solution is a vector C_b, which minimizes

$$\|PC_b - V_b\| = \|r\|, \qquad (6.2)$$

where $\|\cdot\|$ is the Euclidean norm. Data approximation is one of traditional applications of the least squares method. The solution is based on the following definition from Ref. [57].

Definition 1 *Let us consider a matrix P, which is a matrix of size $n \times m$, with $n \geq m$. If P is a matrix of full rank, with $P = V\Sigma U^T$ its singular value decomposition (SVD), then the matrix*

$$P^+ = (P^T P)^{-1} P^T = U \Sigma^{-1} V^T \qquad (6.3)$$

is called the pseudoinverse (Moore-Penrose) matrix for P. If $n < m$ then $P^+ = P^T(PP^T)^{-1}$.

Definition 2 *We say that the matrix A of size $n \times m$ is of a full rank, if its rank $rank(A)$ is equal to $N_0 = \min\{n, m\}$.*

It is safe to assume, that matrix P is a full rank matrix, because all its rows are linearly independent, since they are formed of the randomly chosen values of parameters.

The rank of the matrix P is equal to $rank(P) = N_0 = \min\{N, N_{\text{min}}\} = N_{\text{min}}$, in our case this is the number of columns of the matrix, since the number of rows is $N \geq N_{\text{min}}$.

The pseudoinverse matrix makes it possible to write the solution of the full rank overdetermined least squares problem (6.2), with

$$C_b = P^+ V_b.$$

There are two different ways of calculating P^+ with Eq. (6.3), and both of them, well known and widely used, have been implemented in the `Lagrange` application (for more information on the application see Chapter 7). The first one is the method of reduction of this system to "a normal system of equations" with a square matrix. It is based on Theorem 1 from Ref. [64].

Consider the system $Ax = b$, where the matrix A has the size $n \times m$ and $n > m$, and b is the vector of size n.

Theorem 1 *Among all of the vectors x, which minimize*

$$\|Ax - b\|^2 = \|r\|^2,$$

the vector x, which has the least possible norm, is the unique vector, that satisfies the equation

$$A^T Ax = A^T b.$$

This means, that by solving the system $PC_b = V_b$ via the normal system, we automatically obtain the solution with a minimal residual. To do this, the system of equations with the non-square matrix P

$$PC_b = V_b$$

is multiplied by the transposed matrix P^T from the left, and the obtained system is the "normal system" of the linear equations with the square matrix

$$P^T P\, C_b = P^T V_b \tag{6.4}$$

or

$$DC_b = h_b,$$

where C_b is the vector of unknown coefficients, $h_b = P^T V_b$ is the right hand side vector, and $D = P^T P$ is the square matrix of the normal system.

From here, if the inverse matrix $(P^T P)^{-1}$ exists, the solution of the system is found in the following way:

$$C_b = D^{-1} h_b = (P^T P)^{-1} P^T V_b = P^+ V_b,$$

where the pseudoinverse matrix is equal to

$$P^+ = (P^T P)^{-1} P^T.$$

This method of solving the system is fast and computationally light because with $N > N_{\text{min}}$ the computational cost of calculating the solution is approximately $N_{\text{min}}^2 \cdot N$ floating-point operations. But at the same time it is the least precise method, because of the matrix multiplication [57]. It is preferably used in the case where the condition number of the matrix is not large, providing a level of precision approximately the same as with SVD. Since the method of normal system of equations is a fast and less precise way of solving an overdetermined system, it becomes preferable when the matrix P is a well-posed matrix.

The condition number matters, because (according to Theorem 2 (vide infra in Subsec. 6.3.1)) the solution accuracy is determined by the condition number of the square matrix $D = P^T P$, which is twice larger than the condition number of the matrix P

$$\text{cond}(P^T P) = \text{cond}(P^T) \cdot \text{cond}(P) = \text{cond}^2(P),$$

and solving a system with a large condition number with conventional tools (i.e. gaussian elimination) may prove problematic.

Consequently, in the case when this method is used, it is necessary to take into account the condition number of the matrix and the computational errors. Specifically for this purpose, the `Lagrange` application provides an option for the guaranteed estimation of precision of the system solution.

The second method for solving overdetermined systems, implemented in the `Lagrange` program, is best suited for the case of an ill-posed matrix P, i.e. for matrices with a large condition number.

Theoretically, this method is more computationally heavy and depending on the size of the matrices in question, could take significantly more time to process. It is used if the singular value decomposition of the matrix P is known: $P = V_s \Sigma U_s^T$.

In Eq. (6.3), the square matrix V_s of size (N, N) and the square matrix U_s of size $(N_{\min}, N_{\min})$ are orthogonal matrixes, i.e. $V_s^{-1} = V_s^T$, $U_s^{-1} = U_s^T$, the matrix Σ of size $(N_{\min}, N_{\min})$ is a diagonal matrix $\Sigma = \text{diag}(\sigma_1, \sigma_2, ...\sigma_{N_{\min}})$, which has the positive singular values $\sigma_1, \sigma_2, ...\sigma_{N_{\min}}$ on the diagonal. Usually $\sigma_1 = \sigma_{\max}$ is the maximum singular value, $\sigma_{N_{\min}} = \sigma_{\min}$ is the minimal singular number.

The condition number of the non-square matrix P is equal to

$$\text{cond}(P) = \frac{\sigma_{\max}}{\sigma_{\min}}.$$

The solution of the system in this case is found by substituting the SVD of matrix P into the system of Eq. (6.3)

$$P = V_s \Sigma U_s^T \quad \Rightarrow \quad P C_b = V_b$$

$$(V_s \Sigma U_s^T) \cdot C_b = V_b$$

and multiplying the equation on the left by the inverse matrix $(V_s \Sigma U_s^T)^{-1}$

$$(V_s \Sigma U_s^T)^{-1} (V_s \Sigma U_s^T) \cdot C_b = (V_s \Sigma U_s^T)^{-1} V_b$$

considering, that $(V_s \Sigma U_s^T)^{-1} (V_s \Sigma U_s^T) = I$, it follows that

$$C_b = (V_s \Sigma U_s^T)^{-1} V_b = P^+ V_b.$$

This method of solving the system is exact and does not introduce increased errors, since the multiplication of orthogonal matrixes does not worsen the matrix condition number, but in this case the time needed to solve the system may increase by several factors (with $N > N_{\min}$ the computational cost of finding the least square problem solution using SVD is approximately $4N_{\min}^2 \cdot N$ floating-point operations [57]). It is implemented in the `Lagrange` application for solving systems with large condition numbers in particular, with a large condition number, and does not require an estimation of computational errors.

The SVD method is widely used in physics and mathematics, but Godunov S. K. from the Sobolev Institute of Mathematics of Siberian Branch of Russian Academy of Science (Akademgorodok, Novosibirsk) [65] claims that solving matrix systems of equations via SVD is not really applicable, because in spite of the fact that singular values are uniquely determined via the matrix of the system, for the matrices V_s and U_s transformations might be unstable.

This means, for example, that solving one and the same system via SVD on different computers may produce different V_s and U_s matrices and therefore results might differ.

Given this, the two methods implemented in the `Lagrange` application are applied not only due to their differences in speed and precision, but also for the ability to control the result of the system solution.

6.1.2 The case of N parameter sets and K bins

In the most general case, where the number of tuned parameters is n and the number of bins $b = 1, 2, \dots K$, the system has the following form

$$PC = V, \tag{6.5}$$

with the matrix P of size $(N, N_{\min})$

$$P = \begin{pmatrix} 1 & p'_1(\text{set}1) & \cdots & p'_n p'_n(\text{set}1) \\ 1 & p'_1(\text{set}2) & \cdots & p'_n p'_n(\text{set}2) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p'_1(\text{set}N) & \cdots & p'_n p'_n(\text{set}N) \end{pmatrix},$$

and at the right hand side, the matrix V of size (N, K):

$$V = \begin{pmatrix} n_1(\text{set}1) & n_2(\text{set}1) & \cdots & n_K(\text{set}1) \\ n_1(\text{set}2) & n_2(\text{set}2) & \cdots & n_K(\text{set}2) \\ \vdots & \vdots & \vdots & \vdots \\ n_1(\text{set}N) & n_2(\text{set}N) & \cdots & n_K(\text{set}N) \end{pmatrix}.$$

Each row of the matrix V is a histogram or a set of multiple histograms (with the length of K bins), and represent the Monte Carlo generator response of each observable bin b ($b = 1, 2, \dots, K$) for a randomly chosen parameter vector $\vec{p}$

$$\vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}.$$

The solution of the system $PC = V$ is the matrix of coefficients C of size $(N_{\min}, K)$

$$C = \begin{pmatrix} \alpha_0^{(1)} & \alpha_0^{(2)} & \cdots & \alpha_0^{(K)} \\ \beta_1^{(1)} & \beta_1^{(2)} & \cdots & \beta_1^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{nn}^{(1)} & \gamma_{nn}^{(2)} & \cdots & \gamma_{nn}^{(K)} \end{pmatrix},$$

where the b-th column C_b of matrix C is the estimated set of coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$ for bin b ($b = 1, 2, \dots, K$), in the formula (5.1).

The system $PC = V$ is solved in the same way as is described in the previous section of this chapter.

6.2 Computation of the MC generator response

6.2.1 Simplified example of a single parameter set and bin b

After estimating the coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$, it is possible to simulate the Monte Carlo generator response $MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p})$, using Equation (5.1), for each bin b, for a given set n of parameters $p_1, p_2, ...p_n$.

$$MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p}) = \alpha_0^{(b)} \cdot 1 + \sum_{i=1}^{n} \beta_i^{(b)} p'_i + \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma_{ij}^{(b)} p'_i p'_j, \tag{6.6}$$

where $\vec{p}\,' = \vec{p} - \vec{p}_0$.

It is possible to regard the computation by Equation (6.6), as a scalar product of two vectors.

$$MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p}) = (C_b, \mathbf{p}),$$

where the vector C_b contains independent computed polynomial coefficients for bin b (the b-th column C_b from the calculated matrix C), and $\mathbf{p}$ is *an extended parameter vector* composed from the chosen parameters $p_1, p_2, \ldots, p_n$

$$C_b = \begin{pmatrix} \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \beta_n^{(b)} \\ \gamma_{11}^{(b)} \\ \gamma_{12}^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} 1 \\ p'_1 \\ \vdots \\ p'_n \\ p'_1 p'_1 \\ p'_1 p'_2 \\ \vdots \\ p'_n p'_n \end{pmatrix}.$$

6.2.2 General example of M parameter sets and K bins

To make a scan along a straight line in the parameter hyperspace, it is necessary to define M sets of parameter vectors $\vec{p}$ (which simply correspond to M points on the line). In this case, having defined M sets of n tuned parameters, the calculations of $f_b(\vec{p})$ for a bin b would be performed the following way

$$\begin{pmatrix} \text{column1} & \text{column2} & \cdots & \text{column}N_{\min} \\ 1 & p'_1(\text{set1}) & \cdots & p'_n p'_n(\text{set1}) \\ 1 & p'_1(\text{set2}) & \cdots & p'_n p'_n(\text{set2}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p'_1(\text{set}M) & \cdots & p'_n p'_n(\text{set}M) \end{pmatrix} \begin{pmatrix} \text{bin}b \\ \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix} = \begin{pmatrix} \text{bin}b \\ f_b(\text{set1}) \\ f_b(\text{set2}) \\ \vdots \\ f_b(\text{set}M)) \end{pmatrix},$$

and in general, for K bins:

$$\begin{array}{c} \\ \text{set1} \\ \text{set2} \\ \vdots \\ \text{set}M \end{array} \begin{pmatrix} \text{column1} & \text{column2} & \cdots & \text{column}N_{\text{min}} \\ 1 & p_1'(\text{set1}) & \cdots & p_n' p_n'(\text{set1}) \\ 1 & p_1'(\text{set2}) & \cdots & p_n' p_n'(\text{set2}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p_1'(\text{set}M) & \cdots & p_n' p_n'(\text{set}M) \end{pmatrix} \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ \alpha_0^{(1)} & \alpha_0^{(2)} & \cdots & \alpha_0^{(K)} \\ \beta_1^{(1)} & \beta_1^{(2)} & \cdots & \beta_1^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{nn}^{(1)} & \gamma_{nn}^{(2)} & \cdots & \gamma_{nn}^{(K)} \end{pmatrix} =$$

$$= \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ f_1(\text{set1}) & f_2(\text{set1}) & \cdots & f_K(\text{set1}) \\ f_1(\text{set2}) & f_2(\text{set2}) & \cdots & f_K(\text{set2}) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(\text{set}M) & f_2(\text{set}M) & \cdots & f_K(\text{set}M) \end{pmatrix},$$

or

$$BC = F, \tag{6.7}$$

where each row of F corresponds to a single histogram, and approximates the MC generator response $MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p})$ for each observable bin b ($b = 1, 2, \ldots, K$) for the chosen parameter vector $\vec{p}$:

$$\vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}.$$

During numerical minimization of the χ^2 function by Minuit, it is sufficient to calculate the response $f_b(\vec{p})$ for a single set of the tuned parameters, and in this case the formula (6.7) is reduced to the multiplication of a row from matrix B by matrix C

$$\begin{pmatrix} \text{column1} & \text{column2} & \cdots & \text{column}N_{\text{min}} \\ 1 & p_1'(\text{set 1}) & \cdots & p_n' p_n'(\text{set 1}) \end{pmatrix} \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ \alpha_0^{(1)} & \alpha_0^{(2)} & \cdots & \alpha_0^{(K)} \\ \beta_1^{(1)} & \beta_1^{(2)} & \cdots & \beta_1^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{nn}^{(1)} & \gamma_{nn}^{(2)} & \cdots & \gamma_{nn}^{(K)} \end{pmatrix}$$

$$= \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ f_1(\text{set 1}) & f_2(\text{set 1}) & \cdots & f_K(\text{set 1}) \end{pmatrix} \tag{6.8}$$

6.3 Analysis of computational errors

Any algorithm, is bound to have computational errors which are the sums of the rounding errors (machine epsilon), arising from every arithmetical operation on a computer, and the errors of the algorithm itself. The estimation of the rounding errors is based on the parameter ε_1, which depends on the machine precision, effectively corresponding to the minimal nonzero number, represented by the computer. This is the case of single-precision, $\varepsilon_1 \approx 10^{-8}$, and $\varepsilon_1 \approx 10^{-16}$ for double-precision.

At present, most calculations are carried out on personal computers with the help of popular software frameworks for computations, which implement standard, well known methods. Although all computations are made approximate due to rounding errors in performing elementary arithmetical operations, users rely on the results obtained and ignore the real error on the result, considering that a double-precision for number representation (i.e. the reliable accuracy of any arithmetical operation being of order 10^{-16}) is sufficient enough for obtaining a reliable result. Sometimes such faith leads to numerical paradoxes.

Let us consider one such paradox arising in a numerical computation of a spectral problem

$$Av = \lambda v.$$

The numerical experiment was made with one of the program packages containing methods for matrix spectrum computation using a relative accuracy of 10^{-16} [66].

For the integer square matrix C of order 7×7

$$C = \begin{pmatrix} 289 & 2064 & 336 & 128 & 80 & 32 & 16 \\ 1152 & 30 & 1312 & 512 & 288 & 128 & 32 \\ -29 & -2000 & 756 & 384 & 1008 & 224 & 48 \\ 512 & 128 & 640 & 0 & 640 & 512 & 128 \\ 1053 & 2256 & -504 & -384 & -756 & 800 & 208 \\ -287 & -16 & 1712 & -128 & 1968 & -30 & 2032 \\ -2176 & -287 & -1565 & -512 & -541 & -1152 & -289 \end{pmatrix}$$

the eigenvalues were computed using MATLAB. The following result was obtained:

Approximate spectrum of matrix C:

$$\begin{array}{rcl} \lambda_1(C) & = & \mathbf{6.624}, \\ \lambda_{2,3}(C) & = & \mathbf{4.065} \pm 4.381i, \\ \lambda_{4,5}(C) & = & -\mathbf{1.472} \pm 5.449i, \\ \lambda_{6,7}(C) & = & -\mathbf{5.906} \pm 2.371i. \end{array}$$

Then, using the orthogonal matrix P $(P^{-1} = P^T)$

$$P = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

the following matrices D and F were obtained

$$D = P^{-1}CP, \qquad F = P^{-1}DP = P^{-1}(P^{-1}CP)P.$$

It is readily seen that matrices D and F are obtained from matrix C by rearranging the whole-number rows and columns. In this case in computing D and F via C, there are no roundoff errors.

For matrices D and F, eigenvalues were computed using the same standard program. The eigenvalues of matrices D and F should coincide with the eigenvalues of matrix C. However, the following results were obtained

Approximate spectrum of matrix D:

$$\begin{array}{lcl} \lambda_1(D) & = & \mathbf{6.005}, \\ \lambda_{2,3}(D) & = & \mathbf{3.651} \pm 3.784i, \\ \lambda_{4,5}(D) & = & -\mathbf{1.331} \pm 4.699i, \\ \lambda_{6,7}(D) & = & -\mathbf{5.322} \pm 2.014i. \end{array}$$

Approximate spectrum of matrix F:

$$\begin{array}{lcl} \lambda_1(F) & = & \mathbf{8.095}, \\ \lambda_{2,3}(F) & = & \mathbf{4.984} \pm 5.709i, \\ \lambda_{4,5}(F) & = & -\mathbf{1.825} \pm 7.052i, \\ \lambda_{6,7}(F) & = & -\mathbf{7.207} \pm 3.090i. \end{array}$$

Actually, all eigenvalues of matrices C, D, F are real integer numbers. This follows from the fact that matrix C itself

$$C = L^{-1}RL$$

is obtained using matrix L

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

from the upper triangular matrix R

$$R = \begin{pmatrix} 1 & 2048 & 256 & 128 & 64 & 32 & 16 \\ 0 & -2 & 1024 & 512 & 256 & 128 & 32 \\ 0 & 0 & 4 & 512 & 1024 & 256 & 64 \\ 0 & 0 & 0 & 0 & 512 & 512 & 128 \\ 0 & 0 & 0 & 0 & -4 & 1024 & 256 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2048 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

Consequently, the spectrum of matrix C must coincide with the spectrum of matrix R, which consists of the main diagonal numbers.

Exact spectrum of matrix C:

$$\begin{array}{lcl} \lambda_1(C) & = & \mathbf{0}, \\ \lambda_{2,3}(C) & = & \pm\mathbf{1}, \\ \lambda_{4,5}(C) & = & \pm\mathbf{2}, \\ \lambda_{6,7}(C) & = & \pm\mathbf{4}. \end{array}$$

Comparing the computed spectra of matrices C, D, F and the exact spectrum of matrix C, creates the following question: “What does the standard program compute?”. It is not surprising that one will obtain a different set of eigenvalues by repeating these calculations on one’s own computer. The discrepancy in the results of computations is not caused by the defects of the computational algorithm or the application, but is connected to the errors in performing arithmetical operations and to the construction of this example. It is a delusion to hope for the improvement of computational results by using more modern program packages. It should be realized, that no matter how irreproachable the standard computational application may be, the arithmetical operations performed in the process of its work are necessarily accompanied by roundoff errors, and the value of these errors depend on the word size used for number representation. Although matrices C, D, F are precisely known, it is impossible to avoid arithmetical errors when computing their eigenvalues. The exact eigenvalues of matrix C were obtained in this example by increasing computational precision to the order of 10^{-22}, but this is not always possible and rather expensive.

This example demonstrates that it is not such a bad idea to consider computational errors and be able to estimate them when having to deal with computations of matrices.

Our aim is to estimate the computational errors, which arise in calculations with Eq. (5.1). It is obvious that these errors depend on how precise the polynomial coefficients have been estimated, or in other words, on the precision of the solution of the system of equations (6.1).

As was explained in the previous chapter, the precision of solving this system depends on the chosen method, and there are two different methods implemented in the `Lagrange` application:

a) a faster but less precise method, which uses a standard approach (gaussian elimination), and requires an estimate of computational errors;

b) a slower, computationally heavier method of solving the system via singular value decomposition, which is considered precise and thus doesn't require estimating computational errors.

The following sections describe in detail the algorithm of estimation of computational errors for systems, which are solved via the first method.

6.3.1 Guaranteed precision estimation of the system solution

Perturbation theory. System with a square matrix.

The following theorem is usually used to estimate computational errors when the solved system has a square matrix.

Let us consider the system of equations $Ax = f$, where A is a square, nonsingular matrix. Assume that the matrix A and the right hand side vector f are perturbed: $\widetilde{A} = A + \triangle A$ $\widetilde{f} = f + \triangle f$, where perturbations $\triangle A$, $\triangle f$ are very small. The following theorem gives the answer to the question of how large the perturbation $\triangle x$ of the solution can be [57, 65].

Theorem 2 *Given the system*

$$Ax = f \tag{6.9}$$

and the perturbed system

$$(A + \triangle A)(x + \triangle x) = f + \triangle f,$$

where A and $\triangle A$ are square matrixes of order $N \times N$, f and $\triangle f$ are vectors of size N, and parameters α and φ set limits on the perturbations,

$$\frac{\|\triangle A\|}{\|A\|} \leq \alpha, \qquad \frac{\|\triangle f\|}{\|f\|} \leq \varphi$$

are sufficiently small, and also

$$\alpha \cdot \mathrm{cond}(A) < 1.$$

Then the relative error $\|\triangle x\|/\|x\|$*, which arises due to perturbations* $\triangle A$ *and* $\triangle f$ *of the matrix, satisfies the following inequality*

$$\frac{\|\triangle x\|}{\|x\|} \le \left(\frac{\|\triangle A\|}{\|A\|} + \frac{\|\triangle f\|}{\|f\|}\right) \frac{\operatorname{cond}(A)}{1 - \dfrac{\|\triangle A\|}{\|A\|} \cdot \operatorname{cond}(A)}$$

$$\le (\alpha + \varphi) \frac{\operatorname{cond}(A)}{1 - \alpha \cdot \operatorname{cond}(A)}, \qquad (6.10)$$

where $\operatorname{cond}(A) = \|A\| \cdot \|A^{-1}\|$ *is the condition number of the system matrix (6.9) and* $\|\cdot\|$ *is the operator matrix norm.*

The *operator norm* (or simply a norm) of a matrix A of a matrix A of order $N \times M$ is defined as [57, 65]

$$\|A\| = \max_{x \ne 0} \frac{\|Ax\|}{\|x\|} \qquad (6.11)$$

and has the following properties:

1) $\|A\| \ge 0$, and $\|A\| = 0$ only when $A = 0$;
2) $\|\alpha A\| = |\alpha| \cdot \|A\|$, α – a scalar;
3) $\|A + B\| \le \|A\| + \|B\|$;
4) $\|AB\| \le \|A\| \cdot \|B\|$.

Norms are used for estimating errors in matrix computations, therefore it is essential to know how to compute them. Usually, to estimate the norm of the matrix one uses such numerical characteristics that are readily calculated immediately via its elements and are the *equivalent matrix norms.*

For example, the *Euclidean* (or *Frobenius*) norm of a $N \times M$–matrix A is easily calculated with the explicit formula

$$\|A\|_F = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{M} |a_{ij}|^2}$$

and it is known [65] that

$$\|A\| \le \|A\|_F. \qquad (6.12)$$

The system condition number $\operatorname{cond}(A)$ characterizes how sensitive the solution is to the perturbation of the input data. From the definition it follows that $\operatorname{cond}(A) \ge 1$ and $\operatorname{cond}(A) \Rightarrow \infty$ for singular matrixes. Note that $\operatorname{cond}(A)$ is an important criterion of difficulty in solving a linear system. If $\operatorname{cond}(A)$ is large, then A and the system are said to be *ill-conditioned*, and the smaller is $\operatorname{cond}(A)$, the better the system is conditioned.

The error estimation of the system's approximate solution is divided into two problems:

1) analysis of the rounding errors, which arise from every arithmetical operation at each step of solving the system. It is done to show that the *approximate* result $\tilde{x}$ of the solution of the system $Ax = f$ is the *exact* solution of the perturbed system $(A + \Delta A)\tilde{x} = f + \Delta f$ with small perturbations ΔA and Δf. This, so-called *backward analysis of errors*, makes it possible to calculate ΔA and Δf.

2) after ΔA and Δf have been calculated, the next step is to estimate the error of solving $\frac{\|\triangle x\|}{\|x\|}$, by applying the perturbation theory (for example, Theorem 2).

To solve a system with a square matrix, the `Lagrange` application relies on the conventional method, provided by the ROOT framework, i.e. by means of gaussian elimination or its modifications. Taking this into account, it is not possible to implement a detailed backward analysis of errors to calculate ΔA and Δf, and to estimate (6.10). For that reason, this work uses a different method in constructing the perturbation theory for the problem $Ax = f$.

Let us consider the system $Ax = f$, for which it is necessary to estimate the error of its solution $x = A^{-1}f$, and $\tilde{x}$ is an arbitrary vector. Then it is possible to estimate the difference $\tilde{x} - x = \tilde{x} - A^{-1}f$ in the following way. We shall call vector $r = A\tilde{x} - f$ the *residual* of vector $\tilde{x}$, and r is equal to zero when $\tilde{x} = x$, therefore

$$r = A\tilde{x} - f \quad \text{or} \quad r + f = A\tilde{x}$$

By multiplying the equality by A^{-1} on the left, we arrive at the following equation:

$$A^{-1} \cdot (r + f) = A^{-1} \cdot A \cdot \tilde{x}$$

Taking into account $A^{-1}A = I$ (from the definition of an inverse matrix), then

$$A^{-1}r + A^{-1}f = \tilde{x}.$$

Substituting $\tilde{x}$ into $\tilde{x} - x = \tilde{x} - A^{-1}f$

$$\Delta x = \tilde{x} - x = \tilde{x} - A^{-1}f = A^{-1}r + A^{-1}f - A^{-1}f = A^{-1}r$$

The estimation of the error norm then given by

$$\|\Delta x\| = \|A^{-1} \cdot r\| \leq \|A^{-1}\| \cdot \|r\|. \tag{6.13}$$

Here we used the fact that the norm of the product of two mattices A, B is less or equals to the product of their norms $\|A\|$, $\|B\|$:

$$\|AB\| \leq \|A\| \cdot \|B\|.$$

This simple estimation is attractive from the practical point of view, since it is easily calculated when the approximate solution $\tilde{x}$ is known. Moreover, there is no need to estimate perturbations ΔA and Δf, which are required for the calculation of the estimate (6.10) of the solution error, according to Theorem 2.

In order to determine if the system is a well-posed or an ill-posed one, and to choose a more appropriate method of solving it, the `Lagrange` application calculates the condition number of the system in question, and conveniently uses it in the estimation (6.13)

$$\|\Delta x\| \leq \|A^{-1}\| \cdot \|r\| = \frac{\|A\| \cdot \|A^{-1}\| \cdot \|r\|}{\|A\|} = \text{cond}(A)\frac{\|r\|}{\|A\|}, \tag{6.14}$$

where the residual vector is equal to $r = A\tilde{x} - f \neq 0$, and $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$ is the condition number of matrix A.

The case of N parameter sets and a single bin b

In this simple example, to calculate the unknown coefficients in formula (5.1) for bin b, it is necessary to solve the system of equations (6.1) with N sets of randomly chosen values of n parameters $p_1, p_2, \ldots, p_n$, with $\vec{p}\,' = \vec{p} - \vec{p}_0$:

$$\begin{pmatrix} \text{column1} & \text{column2} & \cdots & \text{column}N_{\text{min}} \\ 1 & p'_1(\text{set1}) & \cdots & p'_n p'_n(\text{set1}) \\ 1 & p'_1(\text{set2}) & \cdots & p'_n p'_n(\text{set2}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p'_1(\text{set}N) & \cdots & p'_n p'_n(\text{set}N) \end{pmatrix} \begin{pmatrix} \text{bin}b \\ \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix} = \begin{pmatrix} \text{bin}b \\ n_b(\text{set1}) \\ n_b(\text{set2}) \\ \vdots \\ n_b(\text{set}N) \end{pmatrix}$$

or

$$PC_b = V_b. \tag{6.15}$$

1. Let us first consider the case, where $N = N_{\text{min}}$ and P is a square matrix of size $N_{\text{min}} \times N_{\text{min}}$.

Solving the system with the square matrix

$$PC_b = V_b$$

by a standard method actually yields an approximate solution $\tilde{C}_b$. To determine the error of the system solution via Formula (6.14) gives

$$\|\Delta C_b\| \leq \text{cond}(P) \frac{\|r_{\text{sq}}^{(b)}\|}{\|P\|}. \tag{6.16}$$

Hence it is necessary to calculate the following quantities

$$\text{cond}(P), \quad \|r_{\text{sq}}^{(b)}\|, \quad \|P\|.$$

According to the definition, given in Theorem 2 and Formula 6.12, the condition number of the square matrix P is equal to

$$\text{cond}(P) = \|P\| \cdot \|P^{-1}\| \leq \|P\|_F \cdot \|P^{-1}\|_F, \tag{6.17}$$

where the Euclidean norm of a matrix is defined in the following way

$$\|P\|_F = \sqrt{\sum_{i=1}^{N_{\text{min}}} \sum_{j=1}^{N_{\text{min}}} (a_{ij})^2}, \quad \|P^{-1}\|_F = \sqrt{\sum_{i=1}^{N_{\text{min}}} \sum_{j=1}^{N_{\text{min}}} (b_{ij})^2},$$

here, a_{ij} and b_{ij} are elements of matrices P and P^{-1} respectively. Equality in Equation (6.17) gives a maximum estimate for the condition number.

The residual vector $r_{\text{sq}}^{(b)}$ of size $N = N_{\text{min}}$ for bin b is determined by substituting the acquired solution $\tilde{C}_b$ into the initial system of equations $PC_b = V_b$ in place of the exact solution C_b:

$$r_{\text{sq}}^{(b)} = P\tilde{C}_b - V_b \neq 0.$$

To avoid zero values when calculating the norm which could appear from the process of squaring the elements of the residual vector, we proceed the following way. Among the elements of the residual vector we choose the maximal element modulo $r^{(b)}_{\text{sq}_{\max}}$ such that

$$|r^{(b)}_{\text{sq}}(i)| \leq r^{(b)}_{\text{sq}_{\max}}.$$

and replace all the elements of the residual vector with that maximal one (the increase of error in this case would be negligible in view of smallness of the values of the elements of the residual vector). Then the norm of the residual vector is equal to

$$\|r^{(b)}_{\text{sq}}\| = \sqrt{(r^{(b)}_{\text{sq}}(\text{set1}))^2 + (r^{(b)}_{\text{sq}}(\text{set2}))^2 + ... + (r^{(b)}_{\text{sq}}(\text{set}N))^2} \leq$$

$$\leq \sqrt{N(r^{(b)}_{\text{sq}_{\max}})^2} = \sqrt{N_{\min}}\, r^{(b)}_{\text{sq}_{\max}}$$

i.e.

$$\|r^{(b)}_{\text{sq}}\| \leq \sqrt{N_{\min}}\, r^{(b)}_{\text{sq}_{\max}}. \tag{6.18}$$

Substituting the calculated values into Formula (6.16) gives us the absolute error of the solution of the system

$$\|\Delta C_b\| \leq \text{cond}(P)\frac{\|r^{(b)}_{\text{sq}_{\max}}\|}{\|P\|_F} \leq \text{cond}(P)\frac{\sqrt{N_{\min}}\, r^{(b)}_{\text{sq}_{\max}}}{\|P\|_F} = \sqrt{N_{\min}}\, \xi^{(b)}_{\text{sq}},$$

where

$$\xi^{(b)}_{\text{sq}} = \text{cond}(P)\frac{r^{(b)}_{\text{sq}_{\max}}}{\|P\|_F}. \tag{6.19}$$

The absolute error of the solution $\|\Delta C_b\|$ makes it possible to determine computational errors for every coefficient in Formula (5.1) for bin b.

$\|\Delta C_b\| = \sqrt{N_{\min}}\, \xi^{(b)}_{\text{sq}}$ is the *guaranteed maximal error* of the vector-column calculation. From here it is possible to estimate the value of the maximal element of the vector of errors of ΔC_b, by calculating its norm according to (6.18) and by using the value of its maximal element $|c^{(b)}_i| \leq c^{(b)}_{\max}$

$$\|\Delta C_b\| \leq \sqrt{N_{\min}}\, c^{(b)}_{\max} = \sqrt{N_{\min}}\, \xi^{(b)}_{\text{sq}}.$$

Consequently

$$c^{(b)}_{\max} = \xi^{(b)}_{\text{sq}}$$

and every element of the vector of errors ΔC_b can be replaced by that maximal value

$$\Delta C_b = \begin{pmatrix} c^{(b)}_1 \\ c^{(b)}_2 \\ \vdots \\ c^{(b)}_{N_{\min}} \end{pmatrix} \leq \begin{pmatrix} c^{(b)}_{\max} \\ c^{(b)}_{\max} \\ \vdots \\ c^{(b)}_{\max} \end{pmatrix} = \begin{pmatrix} \xi^{(b)}_{\text{sq}} \\ \xi^{(b)}_{\text{sq}} \\ \vdots \\ \xi^{(b)}_{\text{sq}} \end{pmatrix}.$$

This means that every element of the vector of the approximate solution $\tilde{C}_b$ is estimated with a guaranteed maximal error:

$$\tilde{C}_b = C_b + \Delta C_b = \begin{pmatrix} \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix} + \begin{pmatrix} \xi_{\text{sq}}^{(b)} \\ \xi_{\text{sq}}^{(b)} \\ \vdots \\ \xi_{\text{sq}}^{(b)} \end{pmatrix} = \begin{pmatrix} \alpha_0^{(b)} + \xi_{\text{sq}}^{(b)} \\ \beta_1^{(b)} + \xi_{\text{sq}}^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} + \xi_{\text{sq}}^{(b)} \end{pmatrix} = \begin{pmatrix} \tilde{\alpha}_0^{(b)} \\ \tilde{\beta}_1^{(b)} \\ \vdots \\ \tilde{\gamma}_{nn}^{(b)} \end{pmatrix}.$$

One can see from these formulas that we can now determine $\xi_{\text{sq}}^{(b)}$, which corresponds to the computational errors of coefficients $\alpha_0^{(b)}, \beta_1^{(b)}, \dots \gamma_{nn}^{(b)}$ in the case of a square system.

2. Let us now consider the case when $N > N_{\min}$ and P a non-square matrix of size $(N, N_{\min})$.

The elements of the unknown coefficients vector C_b are determined by solving the normal system of equations. In order to do that, the following system (composed for a single bin b)

$$PC_b = V_b$$

is multiplied on the left by the transpose matrix P^T

$$P^T P C_b = P^T V_b \tag{6.20}$$

and we obtain the normal system of equations with the square matrix $P^T P$ of size $(N_{\min}, N_{\min})$. Using the matrices $D = P^T P$ and $h_b = P^T V_b$, the system can be written as

$$DC_b = h_b.$$

To solve the normal system we use standard methods provided by the ROOT framework, and estimate the solution errors with Formula (6.14), which in this case has the following form

$$\|\Delta C_b\| \leq \text{cond}(D) \frac{\sqrt{N_{\min}}\, r_{\text{rec}_{\max}}^{(b)}}{\|D\|_F} = \sqrt{N_{\min}}\, \xi_{\text{rec}}^{(b)}, \tag{6.21}$$

where

$$\xi_{\text{rec}}^{(b)} = \text{cond}(D) \frac{r_{\text{rec}_{\max}}^{(b)}}{\|D\|_F}. \tag{6.22}$$

Here, the maximum estimate of the condition number of the square matrix $D = P^T P$ is equal to

$$\text{cond}(D) = \|D\|_F \cdot \|D^{-1}\|_F,$$

and $|r_i^{(b)}| \leq r_{\text{rec}_{\max}}^{(b)}$ is the maximal modulo element of the residual vector.

The residual vector $r_{\text{rec}}^{(b)}$ of length $N_{\min}$ is determined by substituting the acquired approximate solution $\tilde{C}_b$ into the system of equations

$$r_{\text{rec}}^{(b)} = D\tilde{C}_b - h_b \neq 0 \qquad \text{or} \qquad r_{\text{rec}}^{(b)} = P^T P \tilde{C}_b - P^T V_b \neq 0.$$

From here it follows, that the error of the solution of the normal system with the square matrix D is proportional to the *squared condition number* of the non-square matrix P from the initial non-square system $PC_b = V_b$, considering (6.3.1)

$$\operatorname{cond}(D) = \operatorname{cond}(P^T P) = \operatorname{cond}(P^T) \cdot \operatorname{cond}(P) = \operatorname{cond}^2(P).$$

Because of this, such a method of solving a non-square system is said to be fast but imprecise and requires careful consideration of errors.

The case of N parameter sets and K bins

In this case, by solving the system (6.5)

$$PC = V \tag{6.23}$$

we obtain the approximate solution, represented by the matrix of coefficients $\tilde{C}$

$$\tilde{C} = \begin{pmatrix} \tilde{\alpha}_0^{(1)} & \tilde{\alpha}_0^{(2)} & \cdots & \tilde{\alpha}_0^{(K)} \\ \tilde{\beta}_1^{(1)} & \tilde{\beta}_1^{(2)} & \cdots & \tilde{\beta}_1^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\gamma}_{nn}^{(1)} & \tilde{\gamma}_{nn}^{(2)} & \cdots & \tilde{\gamma}_{nn}^{(K)} \end{pmatrix},$$

where the b-th column $\tilde{C}_b$ of matrix $\tilde{C}$ are the polynomial coefficients, estimated with errors $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ for bin b ($b = 1, 2, \ldots, K$) in Formula (5.1).

In the previous section we estimated the error of the approximate solution $\tilde{C}_b$ for a single bin b as

$$P\tilde{C}_b = V_b.$$

We determined that each column of coefficients $\tilde{C}_b$ from the matrix $\tilde{C}$ is calculated with the guaranteed maximal error $\xi^{(b)}$:

$$\tilde{C}_b = C_b + \Delta C_b = \begin{pmatrix} \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix} + \begin{pmatrix} \xi^{(b)} \\ \xi^{(b)} \\ \vdots \\ \xi^{(b)} \end{pmatrix} = \begin{pmatrix} \alpha_0^{(b)} + \xi^{(b)} \\ \beta_1^{(b)} + \xi^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} + \xi^{(b)} \end{pmatrix} = \begin{pmatrix} \tilde{\alpha}_0^{(b)} \\ \tilde{\beta}_1^{(b)} \\ \vdots \\ \tilde{\gamma}_{nn}^{(b)} \end{pmatrix}.$$

Thus, solving the system (6.23) we obtain the approximate solution

$$\tilde{C} = \begin{pmatrix} \text{bin}1 & \text{bin}2 & \cdots & \text{bin}K \\ \tilde{\alpha}_0^{(1)} & \tilde{\alpha}_0^{(2)} & \cdots & \tilde{\alpha}_0^{(K)} \\ \tilde{\beta}_1^{(1)} & \tilde{\beta}_1^{(2)} & \cdots & \tilde{\beta}_1^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\gamma}_{nn}^{(1)} & \tilde{\gamma}_{nn}^{(2)} & \cdots & \tilde{\gamma}_{nn}^{(K)} \end{pmatrix} =$$

$$= \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ \alpha_0^{(1)} & \alpha_0^{(2)} & \cdots & \alpha_0^{(K)} \\ \beta_1^{(1)} & \beta_1^{(2)} & \cdots & \beta_1^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{nn}^{(1)} & \gamma_{nn}^{(2)} & \cdots & \gamma_{nn}^{(K)} \end{pmatrix} + \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ \xi^{(1)} & \xi^{(2)} & \dots & \xi^{(K)} \\ \xi^{(1)} & \xi^{(2)} & \dots & \xi^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \xi^{(1)} & \xi^{(2)} & \dots & \xi^{(K)} \end{pmatrix} = C + \Delta C_{\text{cal}},$$

where the *matrix of computational errors* of the system solution ΔC_{cal} has the following form

$$\Delta C_{\text{cal}} = \begin{pmatrix} \text{bin1} & \text{bin2} & \cdots & \text{bin}K \\ \xi^{(1)} & \xi^{(2)} & \dots & \xi^{(K)} \\ \xi^{(1)} & \xi^{(2)} & \dots & \xi^{(K)} \\ \vdots & \vdots & \vdots & \vdots \\ \xi^{(1)} & \xi^{(2)} & \dots & \xi^{(K)} \end{pmatrix}. \tag{6.24}$$

And $\xi^{(b)}$ is equal to
$\xi^{(b)} = \xi^{(b)}_{\text{sq}}$ (6.19), if $N = N_{\min}$;
$\xi^{(b)} = \xi^{(b)}_{\text{rec}}$ (6.22), if $N > N_{\min}$.

Therefore, having estimated the coefficients for Formula (5.1) by solving the system (6.23) with matrix P of size $(N, N_{\min})$ we are now able to calculate the number of events $f_b(\vec{p})$ in every bin b, using Formula (5.1) for any given set of n parameters $p_1, p_2, ...p_n$.

6.3.2 Guaranteed estimation of errors of parameterization $f_b(\vec{p})$

Using the estimated matrix of coefficients $\tilde{C}$, it is now possible to simulate the Monte Carlo generator response $MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p})$ by using Formula (5.1) for each bin b and a chosen set of n parameters $p_1, p_2, \dots, p_n$

$$MC_b(\vec{p}) = n_b(\vec{p}) \approx f_b(\vec{p}) = \alpha_0^{(b)} \cdot 1 + \sum_{i=1}^{n} \beta_i^{(b)} p'_i + \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma_{ij}^{(b)} p'_i p'_j. \tag{6.25}$$

Case for one parameter set and one bin b

We can now substitute the polynomial coefficients $\tilde{\alpha}_0^{(b)}, \tilde{\beta}_1^{(b)}, \dots \tilde{\gamma}_{nn}^{(b)}$, estimated together with their errors $\xi^{(b)}$, into Formula (6.25), and determine the error of the calculated expectation $f_b(\vec{p})$.

It is clear that by using Formula (6.6), instead of the exact value $f_b(\vec{p})$ we obtain a perturbed value $\tilde{f}_b(\vec{p})$ with an error $\Delta^{(b)}_{f b\text{cal}}(\vec{p})$

$$\tilde{f}_b(\vec{p}) = f_b(\vec{p}) + \Delta^{(b)}_{f b\text{cal}}(\vec{p}).$$

By substituting the estimated coefficients with their respective errors into Formula (6.25) we can determine the error for calculating the number of events $f_b(p)$ in bin b

$$\tilde{f}_b(\vec{p}) = \tilde{\alpha}_0^{(b)} \cdot 1 + \sum_{i=1}^{n} \tilde{\beta}_i^{(b)} p'_i + \sum_{i=1}^{n} \sum_{j=i}^{n} \tilde{\gamma}_{ij}^{(b)} p'_i p'_j =$$

$$= (\alpha_0^{(b)} + \xi^{(b)}) \cdot 1 + \sum_{i=1}^{n} (\beta_i^{(b)} + \xi^{(b)}) p'_i + \sum_{i=1}^{n} \sum_{j=i}^{n} (\gamma_{ij}^{(b)} + \xi^{(b)}) p'_i p'_j =$$

$$= f_b(\vec{p}) + \xi^{(b)} \cdot (1 + \sum_{i=1}^{n} p'_i + \sum_{i=1}^{n}\sum_{j=i}^{n} p'_i p'_j) = f_b(\vec{p}) + \Delta^{(b)}_{fb\mathrm{cal}}(\vec{p}).$$

The computational error $\tilde{f}_b(\vec{p})$ is equal to

$$\Delta^{(b)}_{fb\mathrm{cal}}(\vec{p}) = \xi^{(b)} \cdot (1 + \sum_{i=1}^{n} p'_i + \sum_{i=1}^{n}\sum_{j=i}^{n} p'_i p'_j), \tag{6.26}$$

where
$\xi^{(b)} = \xi^{(b)}_{\mathrm{sq}}$ (6.19), if $N = N_{\mathrm{min}}$;
$\xi^{(b)} = \xi^{(b)}_{\mathrm{rec}}$ (6.22), if $N > N_{\mathrm{min}}$.

This is the *guaranteed maximal computational error* of the calculation of the expected bin contents $\tilde{f}_b(\vec{p})$ in bin b by Formula (6.25). This error also takes into account the error $\xi^{(b)}$ of solving the system $PC_b = V_b$ for bin b.

For convenience of calculations, the `Lagrange` application saves all the computational errors $\Delta^{(b)}_{fb\mathrm{cal}}(\text{set } i)$ into a separate matrix ΔF_{cal}

$$\Delta F_{\mathrm{cal}} = \Delta_{\mathrm{cal}} = \begin{pmatrix} \Delta^{(1)}_{fb\mathrm{cal}}(\mathrm{set}1) & \Delta^{(2)}_{fb\mathrm{cal}}(\mathrm{set}1) & \cdots & \Delta^{(K)}_{fb\mathrm{cal}}(\mathrm{set}1) \\ \Delta^{(1)}_{fb\mathrm{cal}}(\mathrm{set}2) & \Delta^{(2)}_{fb\mathrm{cal}}(\mathrm{set}2) & \cdots & \Delta^{(K)}_{fb\mathrm{cal}}(\mathrm{set}2) \\ \vdots & \vdots & \vdots & \vdots \\ \Delta^{(1)}_{fb\mathrm{cal}}(\mathrm{set}M) & \Delta^{(2)}_{fb\mathrm{cal}}(\mathrm{set}M) & \cdots & \Delta^{(K)}_{fb\mathrm{cal}}(\mathrm{set}M) \end{pmatrix} \tag{6.27}$$

In the process of numerical minimization of the χ^2 by Minuit, the response $f_b(\vec{p})$ is calculated by Formula (6.25) for K bins and a single set of parameters. In the case of a single set, it is the multiplication of a single row of matrix B by matrix $\tilde{C}$ (6.8)

$$\begin{pmatrix} \mathrm{column}1 & \mathrm{column}2 & \cdots & \mathrm{column}N_{\mathrm{min}} \\ 1 & p'_1(\mathrm{set}\ 1) & \cdots & p'_n p'_n(\mathrm{set}\ 1) \end{pmatrix} \begin{pmatrix} \mathrm{bin}1 & \mathrm{bin}2 & \cdots & \mathrm{bin}K \\ \tilde{\alpha}^{(1)}_0 & \tilde{\alpha}^{(2)}_0 & \cdots & \tilde{\alpha}^{(K)}_0 \\ \tilde{\beta}^{(1)}_1 & \tilde{\beta}^{(2)}_1 & \cdots & \tilde{\beta}^{(K)}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\gamma}^{(1)}_{nn} & \tilde{\gamma}^{(2)}_{nn} & \cdots & \tilde{\gamma}^{(K)}_{nn} \end{pmatrix} =$$

$$= \begin{pmatrix} \mathrm{bin}1 & \mathrm{bin}2 & \cdots & \mathrm{bin}K \\ \tilde{f}_1(\mathrm{set}\ 1) & \tilde{f}_2(\mathrm{set}\ 1) & \cdots & \tilde{f}_K(\mathrm{set}\ 1) \end{pmatrix},$$

and the result is a histogram, estimated with errors taken into account

$$\begin{pmatrix} \mathrm{bin}1 & \mathrm{bin}2 & \cdots & \mathrm{bin}K \\ \Delta^{(1)}_{fb\mathrm{cal}}(\mathrm{set}\ 1) & \Delta^{(2)}_{fb\mathrm{cal}}(\mathrm{set}\ 1) & \cdots & \Delta^{(K)}_{fb\mathrm{cal}}(\mathrm{set}\ 1) \end{pmatrix}.$$

6.3.3 Summary

The `Lagrange` application has two different methods for solving the system of equations $PC = V$: via a normal system, and via SVD. When the first method is used, the guaranteed computational errors of the solution are estimated by formulas (6.19) or (6.22), depending on whether matrix P is a square or a non-square matrix. These errors are taken into account in the calculation of the approximated Monte Carlo generator response $f_b(\vec{p})$ by Formulas (5.1) or (5.2).

The error of calculating $\tilde{f}_b(\vec{p})$ by Formula (5.1) (for the 2nd-order polynomial) is equal to (6.26)

$$\Delta^{(b)}_{fb\mathrm{cal}}(\vec{p}) = \xi^{(b)} \cdot (1 + \sum_{i=1}^{n} p'_i + \sum_{i=1}^{n}\sum_{j=i}^{n} p'_i p'_j) \tag{6.28}$$

and with Formula (5.2) for the 3rd-order polynomial

$$\Delta^{(b)}_{fb\mathrm{cal}}(\vec{p}) = \xi^{(b)} \cdot (1 + \sum_{i=1}^{n} p'_i + \sum_{i=1}^{n}\sum_{j=i}^{n} p'_i p'_j + \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} p'_i p'_j p'_k), \tag{6.29}$$

where
n is the number of parameters $p_1, p_2, \ldots, p_n$;
N_{min} is the number of unknown polynomial coefficients (5.3) in the system;
N is the number of parameter sets in the system, where $N \geq N_{\mathrm{min}}$ (see 5.1);

if $N = N_{\mathrm{min}}$ then

$$\xi^{(b)} = \xi^{(b)}_{\mathrm{sq}} = \mathrm{cond}(P)\frac{r^{(b)}_{\mathrm{sq}_{\mathrm{max}}}}{\|P\|_F}, \tag{6.30}$$

$$\mathrm{cond}(P) = \|P\|_F \cdot \|P^{-1}\|_F,$$

$$\|P\|_F = \sqrt{\sum_{i=1}^{N_{\mathrm{min}}}\sum_{j=1}^{N_{\mathrm{min}}} (a_{ij})^2}, \qquad \|P^{-1}\|_F = \sqrt{\sum_{i=1}^{N_{\mathrm{min}}}\sum_{j=1}^{N_{\mathrm{min}}} (b_{ij})^2},$$

here a_{ij} and b_{ij} are the elements of the square matrices P and P^{-1};

$r^{(b)}_{\mathrm{sq}_{\mathrm{max}}}$ is the maximal modulo element of the residual vector $r^{(b)}_{\mathrm{sq}} = P\tilde{C}_b - V_b$ for bin b.

if $N > N_{\mathrm{min}}$ then

$$\xi^{(b)} = \xi^{(b)}_{\mathrm{rec}} = \mathrm{cond}(D)\frac{r^{(b)}_{req_{\mathrm{max}}}}{\|D\|_F}, \quad D = P^T P, \tag{6.31}$$

$$\mathrm{cond}(D) = \|D\|_F \cdot \|D^{-1}\|_F,$$

$$\|D\|_F = \sqrt{\sum_{i=1}^{N_{\mathrm{min}}}\sum_{j=1}^{N_{\mathrm{min}}} (s_{ij})^2}, \qquad \|D^{-1}\|_F = \sqrt{\sum_{i=1}^{N_{\mathrm{min}}}\sum_{j=1}^{N_{\mathrm{min}}} (d_{ij})^2},$$

here s_{ij} and d_{ij} are the elements of the square matrices D and D^{-1};

$r^{(b)}_{req_{\mathrm{max}}}$ is the maximal modulo element of the residual vector $r^{(b)}_{\mathrm{rec}} = D\tilde{C}_b - h_b$ for bin b.

In the case when the second method (SVD) is used to find the solution of the system of equations, we assume that the computational errors of the solution are equal to zero $\xi^{(b)} = 0$ and $\Delta F_{\mathrm{cal}} = 0$, since the error in this case (for double-precision calculations) is of order of 10^{-16}.

6.4 Analysis of statistical erros

6.4.1 Propagation of errors

In the overdetermined system (5.4)

$$PC_b = V_b$$

the statistical uncertainties of the generator propagate from the right side of the equation

$$V_b = \begin{pmatrix} n_b(\text{set1}) \\ n_b(\text{set2}) \\ \vdots \\ n_b(\text{setN}) \end{pmatrix}$$

into the solution

$$C_b = \begin{pmatrix} \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix}.$$

The elements of matrix P consist of a fixed set of parameters, i.e. constant numbers, thus V_b is a *linear* function of the unknown coefficients of the polynomial $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$. This allows to use the *linear* case of the theory of error propagation [60].

Unweighted case

Let us consider the overdetermined system

$$PC_b = V_b. \tag{6.32}$$

In this case, there is no exact solution, therefore we seek the solution with the minimal residual

$$PC_b - V_b = r = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} \neq 0$$

i.e. among all of the possible solutions, we seek the solution C_b with the minimal length d of the residual vector r

$$d = ||r|| = \sqrt{\sum_{i=1}^{n} r_i^2} \quad \Rightarrow \quad \sum_{i=1}^{n} r_i^2 = \text{min}.$$

The least-squares method minimizes the sum of squared residuals, i.e. the following expression is minimized

$$\sum_{i=1}^{n} r_i^2 = r^T \cdot r = (PC_b - V_b)^T \cdot (PC_b - V_b) = \text{min}. \tag{6.33}$$

This is done by calculating the solution of the system $PC_b = V_b$ via the pseudoinverse matrix

$$C_b = P^+ V_b \quad \Rightarrow \quad C_b = MV_b, \quad \text{where} \quad M = P^+ \tag{6.34}$$

where, as follows from Definition 1, the pseudoinverse matrix is found as

$$P^{+} = (P^T P)^{-1} P^T = (V \Sigma U^T)^{-1}.$$

Hence, from (6.34) it follows that the covariance matrix of the solution equals to [60]

$$\mathfrak{D}(C_b) = M \cdot \mathfrak{D}(V_b) \cdot M^T$$

or

$$\mathfrak{D}(C_b) = P^{+} \cdot \mathfrak{D}(V_b) \cdot (P^{+})^T.$$

Consequently, when the solution of the system is found via the normal system of equations, the covariance matrix of the solution is found as

$$\mathfrak{D}(C_b) = [(P^T P)^{-1} P^T] \cdot \mathfrak{D}(V_b) \cdot [(P^T P)^{-1} P^T]^T$$

and similarly in the case of solving the system via SVD

$$\mathfrak{D}(C_b) = (V \Sigma U^T)^{-1} \cdot \mathfrak{D}(V_b) \cdot [(V \Sigma U^T)^{-1}]^T.$$

Weighted case

Similarly to the previous case, we solve the following system

$$PC_b = V_b$$

and if the covariance matrix of the right hand side, $\mathfrak{D}(V_b)$, is known, then it is possible to calculate its inverse, $\mathfrak{D}^{-1}(V_b)$, which is the weights matrix, and then, in accordance with the least-squares method, the minimum of the following expressions is found

$$r^T \cdot \mathfrak{D}^{-1}(V_b) \cdot r = (PC_b - V_b)^T \cdot \mathfrak{D}^{-1}(V_b) \cdot (PC_b - V_b) = \min. \tag{6.35}$$

In order to understand how the statistical errors of the generator propagate into the estimated polynomial coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$, we shall consider the events $n_b(\text{set } i)$ (the Monte Carlo generator response) in bin b for the i-th set of n parameters, as a result of *independent* numerical experiments.

We shall assume that the total number of generated events is sufficiently large, so that every bin b has enough events $n_b(\text{set } i)$ and the distribution of the individual events $n_b(\text{set } i)$ can be approximated by a Gaussian distribution with standard deviation $\sigma_b = \sqrt{n_b(\text{set } i)} = \triangle n_b(\text{set } i)$ [59].

Then it is possible to write down the square covariance matrix $\mathfrak{D}(V_b)$ of size (N, N) of the right hand side vector V_b as

$$\mathfrak{D}(V_b) = \begin{pmatrix} \sigma_{b1}^2 & & & 0 \\ & \sigma_{b2}^2 & & \\ & & \ddots & \\ 0 & & & \sigma_{bN}^2 \end{pmatrix} = \begin{pmatrix} n_b(\text{set}1) & & & 0 \\ & n_b(\text{set}2) & & \\ & & \ddots & \\ 0 & & & n_b(\text{set}N) \end{pmatrix}.$$

Covariances are equal to zero $\text{cov}(n_b(\text{set } i), n_b(\text{set } j)) = 0$, and the dispersions on the main diagonal are $\sigma_b^2 = n_b(\text{set } i)$.

Hence, the matrix of weights $\mathfrak{D}^{-1}(V_b)$ of the right hand side vector V_b is equal to

$$\mathfrak{D}^{-1}(V_b) = \begin{pmatrix} \frac{1}{\sigma_{b1}^2} & & & 0 \\ & \frac{1}{\sigma_{b2}^2} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\sigma_{bN}^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{n_b(\text{set}1)} & & & 0 \\ & \frac{1}{n_b(\text{set}2)} & & \\ & & \ddots & \\ 0 & & & \frac{1}{n_b(\text{set}N)} \end{pmatrix}, \tag{6.36}$$

which follows from the definition of the inverse matrix $\mathfrak{D}(V_b) \cdot \mathfrak{D}^{-1}(V_b) = I$.

The expression (6.35) can be simplified by the Cholesky decomposition for a positive-definite symmetric matrix of weights, which can be represented as a product

$$\mathfrak{D}^{-1}(V_b) = H_b^T \cdot H_b, \tag{6.37}$$

where H_b is the upper triangular matrix of size (N, N), with positive elements on its diagonals.

From (6.37) it follows that matrix H_b equals to

$$\mathfrak{D}^{-1}(V_b) = H_b^2 \quad \Rightarrow H_b = \sqrt{\mathfrak{D}^{-1}(V_b)} \tag{6.38}$$

that is

$$H_b = \begin{pmatrix} \frac{1}{\sigma_{b1}} & & & 0 \\ & \frac{1}{\sigma_{b2}} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\sigma_{bN}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{n_b(\text{set}1)}} & & & 0 \\ & \frac{1}{\sqrt{n_b(\text{set}2)}} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\sqrt{n_b(\text{set}N)}} \end{pmatrix}. \tag{6.39}$$

Let us introduce the following notation

$$P' = H_b P, \qquad V_b' = H_b V_b \tag{6.40}$$

and by substituting $\mathfrak{D}^{-1}(V_b) = H_b^T \cdot H_b$ into expression (6.35), we obtain its simplified form

$$\begin{aligned} (PC_b - V_b)^T \cdot \mathfrak{D}^{-1}(V_b) \cdot (PC_b - V_b) &= (PC_b - V_b)^T \cdot H_b^T H_b \cdot (PC_b - V_b) = \\ = (H_b PC_b - H_b V_b)^T \cdot (H_b PC_b - H_b V_b) &= (P'C_b - V_b')^T \cdot (P'C_b - V_b') = \\ &= (P'C_b - V_b')^2 = \min. \end{aligned} \tag{6.41}$$

Thus, on the basis of the least-squares method, solving the system $PC_b = V_b$ with weights is reduced to the minimization of expression (6.41), i.e. is reduced to finding solution of the primed system $P'C_b = V_b'$.

The method of solving the obtained equation (6.41) is described in detail in [57, 59, 60], and the solution can be expressed in the following form

$$C_b = P'^{\,+} \cdot V_b' \tag{6.42}$$

where $P'^{\,+}$ is the pseudoinverse matrix of matrix P', and according to Definition 1 it is equal to

$$P'^{\,+} = (P'^{\,T} \cdot P')^{-1} P'^{\,T} = (V \Sigma U^T)^{-1}$$

Hence, the system $P'C_b = V_b'$ can be solved in the two ways described further.

1. The first method. Solution via the normal system.

In this case, the system

$$PC_b = V_b$$

is multiplied on the left by H_b

$$H_bP \cdot C_b = H_b \cdot V_b$$

and by noting $P' = H_bP$, $V_b' = H_bV_b$, we obtain the following system, the solution of which we want to find

$$P'C_b = V_b'. \tag{6.43}$$

By multiplying this system on the left by $P'^{\,T}$, we obtain the normal system

$$P'^{\,T}P' \cdot C_b = P'^{\,T} \cdot V_b'$$

or

$$D \cdot C_b = P'^{\,T} \cdot V_b'$$

where $D = P'^{\,T}P'$ is the square matrix of the normal system.

Hence, the approximate solution is equal to

$$C_b = D^{-1}P'^{\,T} \cdot V_b' = (P'^{\,T}P')^{-1}P'^{\,T} \cdot V_b'.$$

Substituting the notations (6.40) $P' = H_bP$, $V_b' = H_bV_b$ and taking into account that $\mathfrak{D}^{-1}(V_b) = H_b^T \cdot H_b$ and $P'^{\,T} = (H_b \cdot P)^T = P^T \cdot H_b^T$, we get the following

$$C_b = (P'^{\,T}P')^{-1}P'^{\,T} \cdot V_b' = (P^T \cdot \underbrace{H_b^T H_b} \cdot P)^{-1}P^T \cdot \underbrace{H_b^T H_b} \cdot V_b =$$

$$= \underbrace{(P^T \cdot \mathfrak{D}^{-1}(V_b) \cdot P)^{-1}}_{D^{-1}} P^T \cdot \mathfrak{D}^{-1}(V_b) \cdot V_b = \underbrace{D^{-1} \cdot P^T \cdot \mathfrak{D}^{-1}(V_b)}_{M} \cdot V_b$$

or

$$C_b = M \cdot V_b \tag{6.44}$$

where

$$M = D^{-1} \cdot P^T \cdot \mathfrak{D}^{-1}(V_b) = (P^T \cdot \mathfrak{D}^{-1}(V_b) \cdot P)^{-1}P^T \cdot \mathfrak{D}^{-1}(V_b).$$

The covariance matrix of solution C_b is equal to the inverse matrix of the normal system of equations [59, 60]

$$\mathfrak{D}(C_b) = D^{-1} = (P^T \cdot \mathfrak{D}^{-1}(V_b) \cdot P)^{-1} \tag{6.45}$$

i.e. the diagonal elements of matrix D^{-1} are the dispersions σ_i^2 of the elements of the vector-column C_b, and the errors (standard deviations) are equal to $\triangle C_b(i) = \sqrt{\sigma_i^2} = \sigma_i$.

The covariance matrix of the solution C_b (6.44) can be written as [60]

$$\mathfrak{D}(C_b) = M \cdot \mathfrak{D}(V_b) \cdot M^T = \tag{6.46}$$

$$= [D^{-1} \cdot P^T \cdot \mathfrak{D}^{-1}(V_b)] \cdot \mathfrak{D}(V_b) \cdot [D^{-1} \cdot P^T \cdot \mathfrak{D}^{-1}(V_b)]^T =$$

$$= [(P^T \cdot \mathfrak{D}^{-1}(V_b) \cdot P)^{-1} \cdot P^T \cdot \mathfrak{D}^{-1}(V_b)] \cdot \mathfrak{D}(V_b) \cdot [(P^T \cdot \mathfrak{D}^{-1}(V_b) \cdot P)^{-1} \cdot P^T \cdot \mathfrak{D}^{-1}(V_b)]^T,$$

where the last expression can be reduced to the form of (6.45) [60].

2. The second method. Solving the overdetermined system of equations via SVD. In this case we are looking for the solution of the primed system (6.43)

$$P'C_b = V'_b$$

and if the singular value decomposition of matrix P' is known

$$P' = (H_b P) = V \Sigma U^T$$

we can substitute it into the system in the place of P', obtaining the following expression

$$P'C_b = V'_b \quad \Rightarrow \quad (V\Sigma U^T) \cdot C_b = V'_b$$

and accounting for $V'_b = H_b V_b$, the solution is as follows

$$C_b = (V\Sigma U^T)^{-1} V'_b = \underbrace{(V\Sigma U^T)^{-1} \cdot H_b}_{M} \cdot V_b$$

or

$$C_b = M \cdot V_b, \quad \text{where} \quad M = (V\Sigma U^T)^{-1} \cdot H_b.$$

Hence, it is possible to calculate the covariance matrix for the solution vector C_b as

$$\mathfrak{D}(C_b) = M \cdot \mathfrak{D}(V_b) \cdot M^T = \tag{6.47}$$

$$= [(V\Sigma U^T)^{-1} \cdot H_b] \cdot \mathfrak{D}(V_b) \cdot [(V\Sigma U^T)^{-1} \cdot H_b]^T,$$

where $H_b = \sqrt{\mathfrak{D}^{-1}(V_b)}$, $H_b^T = H_b$.

6.4.2 Statistical errors matrix of the system solution

We have found the solution of the overdetermined system of linear equations

$$PC = V \tag{6.48}$$

where matrix $C = (C_1, C_2, ...C_b, ...C_K)$, the matrix of polynomial coefficients, of size $(N_{\min}, N_{\min})$, consists of vector-columns C_b and represents the solution of the system, and matrix $V = (V_1, V_2, ..., V_b, ...V_K)$ the right hand side matrix of size (N, K), consists of vector-columns V_b.

We have discussed the two methods of solving system (6.48), via the normal system of equations and via SVD, and in both cases, for each column of the estimated solution C_b

$$C_b = \begin{pmatrix} \text{bin}b \\ \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix}$$

we have calculated the covariance matrix $\mathfrak{D}(C_b)$ (6.45) or (6.47).

The values of the main diagonal of the covariance matrix $\mathfrak{D}(C_b)$ are the dispersions σ_i^2 of the corresponding elements of C_b

$$\mathfrak{D}(C_b) = \begin{pmatrix} a_1 & * & \dots & * \\ * & a_2 & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & a_{N_{\min}} \end{pmatrix} = \begin{pmatrix} \sigma_b^2(\alpha_0) & * & \dots & * \\ * & \sigma_b^2(\beta_1) & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & \sigma_b^2(\gamma_{nn}) \end{pmatrix} \tag{6.49}$$

and the off-diagonal matrix elements are denoted by asterisks.

Consequently, the statistical errors of the Monte Carlo generator propagate into every element of C_b, and the estimates for standard deviation are equal to the square root of the diagonal elements of the covariance matrix $\mathfrak{D}(C_b)$. Thus, for bin b

$$\begin{aligned} \triangle\alpha_0^{(b)} &= \sqrt{a_1} = \sqrt{\sigma_b^2(\alpha_0)} = \sigma_b(\alpha_0) \\ \triangle\beta_1^{(b)} &= \sqrt{a_2} = \sqrt{\sigma_b^2(\beta_1)} = \sigma_b(\beta_1) \\ &\vdots \\ \triangle\gamma_{nn}^{(b)} &= \sqrt{a_{N_{\min}}} = \sqrt{\sigma_b^2(\gamma_{nn})} = \sigma_b(\gamma_{nn}) \end{aligned}$$

and these errors form the matrix of statistical errors ΔC_{stat} of the system solution

$$\Delta C_{\text{stat}} = \begin{pmatrix} \text{bin } 1 & \dots & \text{bin } b & \dots & \text{bin } K \\ \vdots & \dots & \sigma_b(\alpha_0) & \dots & \vdots \\ \vdots & \dots & \sigma_b(\beta_1) & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \sigma_b(\gamma_{nn}) & \dots & \vdots \end{pmatrix}.$$

Hence, the matrix of statistical errors ΔC_{stat} of size $(N_{\min}, K)$, which consolidates the statistical errors, propagating into the solution of system $PC = V$ because of the uncertainties of the Monte Carlo generator, can be expressed in the following form

$$\Delta C_{\text{stat}} = \begin{pmatrix} \triangle\alpha_0^{(1)} & \triangle\alpha_0^{(2)} & \dots & \triangle\alpha_0^{(K)} \\ \triangle\beta_1^{(1)} & \triangle\beta_1^{(2)} & \dots & \triangle\beta_1^{(K)} \\ \vdots & \vdots & & \vdots \\ \triangle\gamma_{nn}^{(1)} & \triangle\gamma_{nn}^{(2)} & \dots & \triangle\gamma_{nn}^{(K)} \end{pmatrix} = \begin{pmatrix} \sigma_1(\alpha_0) & \sigma_2(\alpha_0) & \dots & \sigma_K(\alpha_0) \\ \sigma_1(\beta_1) & \sigma_2(\beta_1) & \dots & \sigma_K(\beta_1) \\ \vdots & \vdots & & \vdots \\ \sigma_1(\gamma_{nn}) & \sigma_2(\gamma_{nn}) & \dots & \sigma_K(\gamma_{nn}) \end{pmatrix}.$$

6.4.3 Statistical errors of parameterization

The obtained solution $C = (C_1, C_2, ..., C_b, ..., C_K)$ is used for simulating the response of the Monte Carlo generator f_b, using formula (5.1), and its errors matrix $\mathfrak{D}(C_b)$ is used to estimate the statistical errors $f_b(\vec{p})$.

In the case of the line-scan in the parametrical hypercube (M parameter sets, K bins), matrix B, constructed from the parameter vectors, is multiplied by matrix C, the matrix of polynomial coefficients, to obtain the matrix of resulting histograms F (6.7)

$$BC = F$$

and in order to estimate statistical errors for each f_b, we calculate the matrix of statistical errors ΔF_{stat}.

For each column b we can write down the following expression

$$BC_b = F_b \quad \Rightarrow \quad B \begin{pmatrix} \text{bin}b \\ \alpha_0^{(b)} \\ \beta_1^{(b)} \\ \vdots \\ \gamma_{nn}^{(b)} \end{pmatrix} = \begin{pmatrix} \text{bin}b \\ f_b(\text{set}1) \\ f_b(\text{set}2) \\ \vdots \\ f_b(\text{set}M) \end{pmatrix}.$$

In this case, the statistical error of column C_b propagates into F_b, according to the rule of error propagation [60], which is expressed in matrix form for each bin b in the form of the covariance matrix for F_b, and equals to

$$\mathfrak{D}(F_b) = B \cdot \mathfrak{D}(C_b) \cdot B^T = \tag{6.50}$$

$$= \begin{pmatrix} d_1 & * & \dots & * \\ * & d_2 & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & d_M \end{pmatrix} = \begin{pmatrix} \sigma^2_{fb}(\text{set}1) & * & \dots & * \\ * & \sigma^2_{fb}(\text{set}2) & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & \sigma^2_{fb}(\text{set}M) \end{pmatrix}$$

where the off-diagonal matrix elements are denoted by asterisks. The square root values of the diagonal elements d_i of this matrix correspond to the uncertainties of estimating $f_b(\text{set } i)$ of the column F_b.

Thus, for bin b

$$\begin{aligned} \triangle^{(b)}_{fb\text{stat}}(\text{set}1) &= \sqrt{d_1} = \sigma_{fb}(\text{set}1) \\ \triangle^{(b)}_{fb\text{stat}}(\text{set}2) &= \sqrt{d_2} = \sigma_{fb}(\text{set}2) \\ &\vdots \\ \triangle^{(b)}_{fb\text{stat}}(\text{set}M) &= \sqrt{d_M} = \sigma_{fb}(\text{set}M) \end{aligned} \tag{6.51}$$

and these values correspond to the column b of the matrix of statistical errors of the events of the calculated histograms, ΔF_{stat}

$$\Delta F_{\text{stat}} = \Delta_{\text{stat}} = \begin{pmatrix} \text{bin } 1 & \dots & \text{bin } b & \dots & \text{bin } K \\ \vdots & \dots & \sigma_{fb}(\text{set}1) & \dots & \vdots \\ \vdots & \dots & \sigma_{fb}(\text{set}2) & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \sigma_{fb}(\text{set}M) & \dots & \vdots \end{pmatrix}.$$

Thereby, in the case of solving the overdetermined system $PC = V$, the errors of the solution, which arise from the statistical uncertainties of the Monte Carlo generator, propagate into the system solution C, and further into events f_b, calculated by the formula (5.1), and are estimated and consolidated in the matrix of statistical errors ΔF_{stat}

$$\Delta F_{\text{stat}} = \begin{pmatrix} \sigma_{f1}(\text{set}1) & \sigma_{f2}(\text{set}1) & \dots & \sigma_{fK}(\text{set}1) \\ \sigma_{f1}(\text{set}2) & \sigma_{f2}(\text{set}2) & \dots & \sigma_{fK}(\text{set}2) \\ \vdots & \vdots & & \vdots \\ \sigma_{f1}(\text{set}M) & \sigma_{f2}(\text{set}M) & \dots & \sigma_{fK}(\text{set}M) \end{pmatrix} =$$

$$= \begin{pmatrix} \triangle^{(1)}_{fb\mathrm{stat}}(\mathrm{set1}) & \triangle^{(2)}_{fb\mathrm{stat}}(\mathrm{set1}) & \dots & \triangle^{(K)}_{fb\mathrm{stat}}(\mathrm{set1}) \\ \triangle^{(1)}_{fb\mathrm{stat}}(\mathrm{set2}) & \triangle^{(2)}_{fb\mathrm{stat}}(\mathrm{set2}) & \dots & \triangle^{(K)}_{fb\mathrm{stat}}(\mathrm{set2}) \\ \vdots & \vdots & & \vdots \\ \triangle^{(1)}_{fb\mathrm{stat}}(\mathrm{set}M) & \triangle^{(2)}_{fb\mathrm{stat}}(\mathrm{set}M) & \dots & \triangle^{(K)}_{fb\mathrm{stat}}(\mathrm{set}M) \end{pmatrix}. \tag{6.52}$$

Both cases of calculation of the statistical errors (with and without weights) are implemented in the `Lagrange` application.

6.4.4 Properties of estimates, obtained by the least-squares method

The estimates of results, obtained by the least-squares method, have important properties, which do not depend on the type of errors distribution (either errors are distributed according to the normal distribution, or their distribution is unknown) [59, 60]:

- The estimate is *unbiased*, i.e. it is safe to assume that the estimated values of the polynomial coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$ and f_b are close to the true values, when a large number of events is generated.
- The least-squares method estimate has *minimal dispersion* (according to the Gauss-Markov theorem), i.e.

$$\sigma_b(\alpha_0) = \min, \;\; \sigma_b(\beta_1) = \min, \;\; \cdots, \;\; \sigma_b(\gamma_{nn}) = \min, \;\; \sigma_{fb} = \min.$$

6.4.5 Summary

While calculating χ^2 by formula (5.6), it is necessary to account for statistical and computational errors of f_b

$$\chi^2(\vec{p}) = \sum_{b=1}^{K} \frac{(\tilde{f}_b(\vec{p}) - R_b)^2}{\Delta_b^2}.$$

In order to do that, we calculate Δ_b^2 for each bin b, as the sum in quadrature
of the experimental error $\Delta^{(b)}_{\mathrm{exper}}$,
statistical error $\Delta^{(b)}_{fb\mathrm{stat}}(\vec{p})$ (6.52) and
computational error $\Delta^{(b)}_{fb\mathrm{cal}}(\vec{p})$ (6.27)

$$\Delta_b^2 = (\Delta^{(b)}_{\mathrm{exper}})^2 + (\Delta^{(b)}_{fb\mathrm{stat}}(\vec{p}))^2 + (\Delta^{(b)}_{fb\mathrm{cal}}(\vec{p}))^2.$$

Chapter 7

Developed tools

Unusquisque sua noverit ire via.
Lat.: Each and every man should know to go the way of his own.

This chapter presents a brief overview of the developed application package, describing its components, their architecture and main features. The package consists of three separate programs, the main of which is the tuning application `Lagrange`, which implements methods described previously in Chapters 5 and 6. Complementing the main tuning program are the two "helper" components: the analysis and job manager applications, which are all described in sections below.

7.1 Tuning cycle

The main tuning procedure consists of a number of iterations, in each of which the tuning application, with the help of LHCb software framework, generates a new sample of Monte Carlo data, which is compared to a sample of experimental data, recorded by the LHCb detector. The number of these iterations depends on the total number of parameters being tuned in the process, on the order of the polynomial which was chosen to fit the generator response, and on the degree of over-determination of the fit. Figure 7.1 presents the diagram of the tuning process.

In order to generate a new sample of Monte Carlo data, the tuning application relies on the `JobManager` program. The tuning application has no part in the generation process, it only specifies the desired size of the Monte Carlo data sample (i.e. the required number of events), and provides a vector of PYTHIA parameters, together with their values, which should be used during the simulation phase. From here, the `JobManager` application takes over and controls the whole sample-generation procedure by setting up the environment for the LHCb software, managing distribution of computational jobs across the available resources, and monitoring the execution.

In the final stage of each iteration, `JobManager` triggers the `Analysis` application, which runs over the newly generated data sample and produces a set of histograms of distributions, which serve as input for the `Lagrange` tuning application.

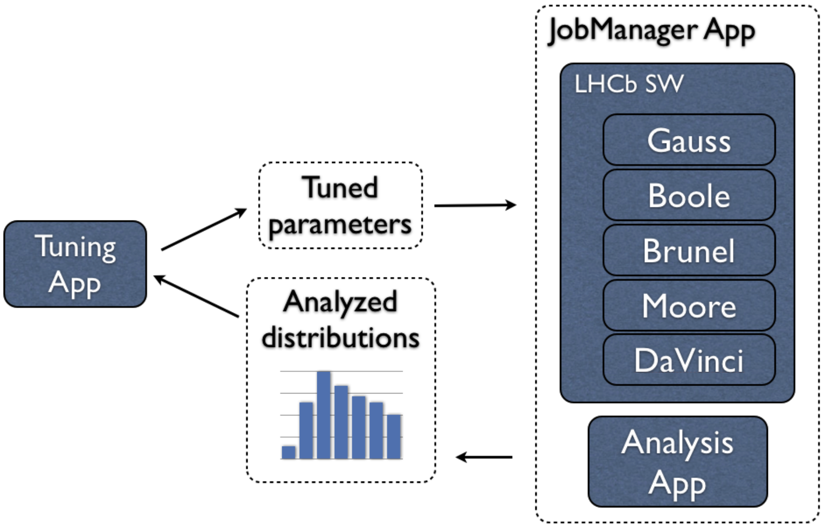


Figure 7.1: The diagram of the Lagrange application package tuning process.

7.2 Tuning application

The Lagrange tuning application is a modern tool for event generator tuning, which was developed using the C++ programming language and employs the ROOT object-oriented multipurpose data analysis package.

The program implements two methods of solving a linear system of equations (5.4): via the normal system of equations or via singular value decomposition. Since the first method is considered to be faster than the second one, but less precise, computational errors should be taken into account. The Lagrange application calculates the guaranteed estimation of precision for the computation of polynomial coefficients $\alpha_0^{(b)}, \beta_i^{(b)}, \gamma_{ij}^{(b)}$ and $\delta_{ijk}^{(b)}$ when the system is solved by the first method. The errors of the coefficients allow one to estimate the precision of the approximated Monte Carlo generator response $f_b(\vec{p})$ by Eq. (5.1) or (5.2). For both methods, the tuning application performs an estimate of the respective statistical errors, as described in Chapter 6.

Main features of the tuning application

A fast, analytical "pseudo-generator".

The key feature of the Lagrange tuning program is the simulation of the Monte Carlo generator response by an approximation formula (the applications provides an option to use a 2^{nd} or 3^{rd} order polynomial) for a given set of parameters, that considerably shortens the time spent for the investigation of the studied process.

Estimation of statistical and computational errors.
The Lagrange application provides not only an estimation of statistical errors of the parameterization results, but also computational ones. As is described in Chapter 6, the tuning program features a method of guaranteed estimation of precision for parameterization result, which is an important, distinctive feature of the Lagrange tuning package compared to similar software.

Full detector simulation.
One of the most important features of the tuning package is the junction with the LHCb software framework. This option allows the Lagrange application to recreate the full setup of LHCb detector (i.e. geometry, settings of triggers, reconstruction options, etc.), and simulate Monte Carlo data samples with the same conditions that were used to record the experimental data sample.

Flexibility in parameter hypercube sampling.
In order to parameterize the Monte Carlo generator response, it is necessary to sample the parameter hypercube, e.g. to choose a set of parameter vectors and their values. The conventional approach is to sample the hypercube uniformly, using a standard pseudorandom number generator to populate parameter values. In addition to this traditional method, the Lagrange application provides an option to use a quasi-random sequence generator, which implements Sobol quasi-random low-discrepancy sequences algorithm [67] and allows for a much more homogeneous sampling of multidimensional spaces.

Simultaneous parameterization of multiple distributions.
The tuning program is able to use a set, composed of multiple histograms of different distributions, in its parameterization process.

Parameterization checks.
The tuning application provides the functionality to validate the parameterization results with a straight line scan in the parameter space. One such validation can be executed after the computation of the unknown coefficients of the approximation formula and is done within the initial boundaries of parameter values, with the scan line passing through the point representing the default parameter values. The second scan test can be performed within user-defined limits around the optimal parameter values, determined by minimization. These line scans present a visual comparison of performance of the analytical pseudo-generator to the response of the Monte Carlo generator.

Numerical minimization.
The Lagrange application offers an option of numerical minimization of the χ^2 function of the estimated Monte Carlo generator response approximation formula, to determine the best values of the tuned parameters. It is possible to define multiple starting points for the minimization process, in order to search for the global minimum. Parameter values, which define these minimization entrance points, can be either assigned by the standard pseudorandom number generator, or by the quasi-random sequence generator.

Other features.
The Lagrange application package has many options which give the user a fine-grained control over the tuning process. They include the assignment of bin-wise weights for histograms, normalization of distributions, control of overdetermination of the fit, weighted or unweighted fit,

the choice of two methods for solving the main matrix equation: with a normal system or SVD, parameter likelihood profiling and others. The whole tuning process is logged in detail and thus can be checked when necessary.

Briefly, the main algorithm of the tuning application may be broken down into the following steps:

I. Parsing of reference data.

II. Generation of random parameter vectors $\vec{p} = (p_1, p_2, \ldots, p_n)$ in the n-dimensional parameter hypercube.

III. Calculation of unknown polynomial coefficients in (5.1) or (5.2):

- construction of a system of linear equations,
- solution of the system via the preferred method (SVD or a normal system, with or without account of weights),
- estimation of computational and statistical errors.

IV. An optional parameterization check by scanning along a straight line in parameter hyperspace and comparing the χ^2 of the parameterization and the Monte Carlo generator response.

V. Numerical minimization of the approximation χ^2 and determination of the corresponding best parameter vector p_{best} with the interval of errors.

VI. An optional validation test to compare the response of the Monte Carlo generator to the approximation in the region of the estimated χ^2 minimum around the $\vec{p}_{\text{best}}$ values.

VII. Optional likelihood profiling procedure of the tuned parameters to determine confidence intervals.

7.3 Helper programs

7.3.1 `JobManager`

The `JobManager` application was developed in the Perl language and serves as an interface between the core tuning application and the process of generating and analyzing Monte Carlo data samples. It acts as a wrapper around the LHCb software framework and the `Analysis` application, removing the functionality required to interact with them from the `Lagrange` program, making the procedure of requesting for a new simulated data sample as transparent is it can be: specify the parameters with their values and the number of events to generate as the input information, and collect the resulting histograms as the output, the result of the simulation.

The two main internal tasks of the `JobManager` application are:

1. Setup of the environment for computational jobs (see Fig. 7.2);

2. Distribution, execution and monitoring of the jobs on the computational cluster (see Fig. 7.3).

When the `JobManager` is executed, it reads its configuration and prepares the environment for the computational jobs. The program has information on the resources of the computational cluster: how many computational nodes it can access, the number of CPUs per node and the number of physical core, per CPU. Together with the requested total number of events it is enough to define the total number of jobs and the number of events per job, to be executed in parallel to make the maximal use of the available resources and to reduce time needed to complete the simulation.

For every computational job, `JobManager` creates a separate directory structure and configures every step of the simulation process. As explained in Chapter 4, the complete simulation cycle in the LHCb software framework consists of the following five steps:

1. Simulation of the events (GAUSS application).
2. Digitization, simulate the response of the detector electronics (BOOLE application).
3. Data reconstruction (BRUNEL application).
4. Application of trigger conditions (MOORE application).
5. Physics analysis (DaVinci application).

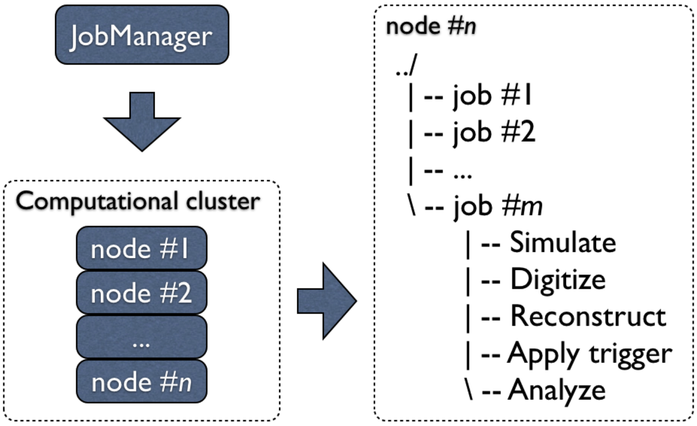


Figure 7.2: Environment setup by the `JobManager` application: creation of directory structures and configuration files for every step of the simulation process, for each computational node and job.

For each of the LHCb applications in these steps, the `JobManager` creates configuration files, which control all the different options, such as the required version of the program to be used, detector conditions databases, file paths, etc.

When the environment is ready, the `JobManager` creates job execution programs, one per computational node, which launch simulation jobs in parallel on the nodes and monitor their progress.

Storage space of the MPIK LHCb computational cluster is shared among all of its nodes via NFS and starting a large number of simulation jobs simultaneously creates a very high load on the storage system: launching of LHCb software applications involves a large number of internal files and databases of the framework checked, read and loaded into the host memory. All of this, done by many simulation jobs at the same time, impacts performance of the storage system and may slow it down tremendously. In order to avoid this scenario, the `JobManager` is able to configure job execution programs to set a certain delay time for the jobs, before they are launched. This solution reduces the immediate impact on the storage system and greatly reduces

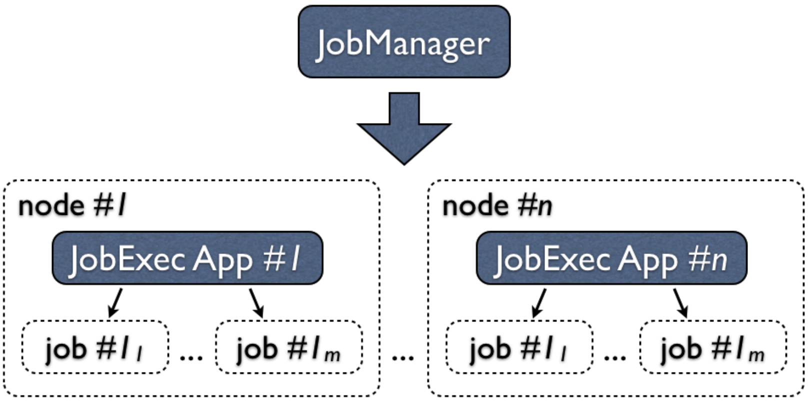


Figure 7.3: Hierarchy of job control: the `JobManager` controls job execution programs on every computational node; job execution programs control simulation jobs on a single computational node.

the time consumed by every simulation job to load and configure all the required packages of LHCb framework.

Like any other large software project with a long history and more than one programming language used in its development, the LHCb software framework is not free of bugs. During the development of the `Lagrange` tuning package, it was seen that the GAUSS application, responsible for the first phase of the simulation process, may enter an endless loop while generating an event, when certain conditions are met, and thus stalling the whole tuning process. To avoid such situations, the `JobManager` was upgraded with a mechanism which monitors the state of simulation jobs and terminates the stalled ones, preventing the whole tuning procedure from stopping. The data files of the terminated jobs can not be used and the events they have generated are wasted, but the observed percentage of stalling jobs was at most on the level of 5%. Additionally, an option was introduced to the `JobManager` application, allowing to specify an overhead in generated events per simulation job, to stay on the safe side and deliver the total requested number of events even if some jobs fail.

After the Monte Carlo data sample is fully ready, the `JobManager` invokes the `Analysis` application, the second helper program of the `Lagrange` tuning package, which runs over the simulated events and prepares the requested distributions, ready for the tuning application to input.

All the steps of the simulation procedure are logged and the `JobManager` saves the workspaces of simulation jobs, so that this information can be accessed whenever that is needed.

7.3.2 `Analysis` application

The `Lagrange Analysis` application is the second helper program of the tuning software package. It was developed in C++ and uses the ROOT framework to perform physics analysis of the Monte Carlo data, generated by LHCb software.

In the last phase of the simulation procedure, the `JobManager` application executes the DAVINCI package SDF (Simple Data Format), developed by Markward Britsch from MPIK, which converts a conventional LHCb framework DST data format to a simpler ROOT file, excluding information which is not necessary for the present analysis.

The `Analysis` program runs its physics analysis algorithm over these SDF Monte Carlo data files and fills histograms of requested distributions. All the produced histograms are saved both as C++ code files, so they could be used in ROOT framework, and as images, for fast visual inspection. A separate output is created for the `Lagrange` tuning application, in the form of a plain ASCII file. Depending on the physics analysis programmed in the application, the output file may have not just one, but a set of histograms.

Together with the resulting histograms, the `Analysis` application saves into its output the number of events which passed its physics algorithm, so that the `Lagrange` tuning program could perform normalization to the number of generated events in the case when it is requested.

7.4 Validating the solution of the system of equations

In the `Lagrange` tuning application, the unknown coefficients of the fitting polynomial are defined by solving the system of linear equations $PC = V$ (6.5). The solution of this system is calculated, using one of the methods described in the previous chapters. If a user doesn't enforce the application to use only the second method, the tuning program estimates the condition number of the matrix system, prior to choosing the appropriate method of solving it. If the condition number of the system is larger than a certain predefined threshold, e.g. cond $> 10^{15}$, the second method of solving the system is chosen automatically. In the opposite case, the program proceeds with the method of normal system.

Such a limit, imposed on the condition number of the matrix system, allows one to avoid potential significant computational uncertainties (see estimations (6.30), (6.31)) of the solution of the system and of the results, estimated by the approximation formula (5.1) (see estimations (6.28), (6.29)), i.e. for the Monte Carlo generator response simulations, in the case when the matrix system was solved via the normal system of equations.

7.5 Parameterization check with a line scan

In order to verify the result of the Monte Carlo generator response approximation, the `Lagrange` application uses a line scan test that performs a scan along a straight line in parameter space and produces a graphical comparison of the parameterization to the Monte Carlo generator response (see Fig. 7.4).

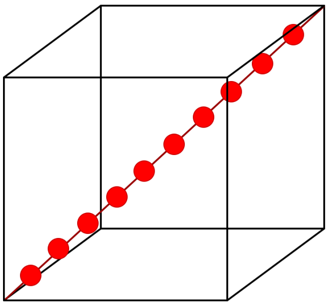


Figure 7.4: A scan along a straight line in a parameter hyperspace.

Each point on the line in the hypercube represents a certain set of parameters $\vec{p_i}$. Scanning along a line means a simultaneous shift of all the parameters p_i (within their limits) by a certain step value $h_i = (b_i - a_i)/M$, where M corresponds to the total number of shifts along the line ($M+1$ is the total number of points).

This procedure allows one to test the approximation of the Monte Carlo generator response in the two following ways: via directly comparing histograms of distributions, which use, as a source, data simulated by the Monte Carlo generator and histograms, populated by Formula (5.1) or (5.2), for one and the same set of parameters, and via calculation and comparison of the χ^2 for the parameterization and MC generator response in respect to the reference data.

In the first case, for every point on the line (i.e. for every set of parameters used in the line scan) the `Lagrange` application uses the Monte Carlo data to produce a set of histograms for the chosen distribution, and for each histogram it plots a band of uncertainties of $\triangle n_b(\vec{p}) = \pm\sigma_b(\vec{p}) = \pm\sqrt{n_b(\vec{p})}$ (corresponding to 68% percent of the expected events). The produced plots are then overlaid with histograms, filled by the approximation formula, and the resulting plots are a perfect check for the consistency between the approximation formula and the Monte Carlo generator.

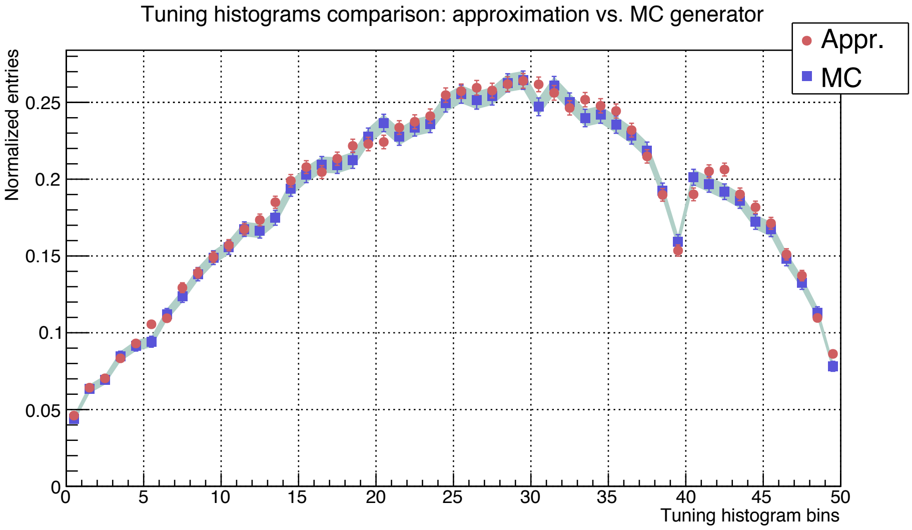


Figure 7.5: An example of the tuning histogram, containing the first 50 bins of the tuning data vector, which correspond to the pseudo-rapidity distribution (η) in the range $2 < \eta < 5$. The tuning histogram was normalized to the number of events, accepted by the analysis algorithm.

In the second case, in every point along the scanned line, the `Lagrange` application calculates the χ^2 function in the following two ways:

$$a) \qquad \chi^2(\vec{p}) = \sum_{b=1}^{K} \frac{(n_b(\vec{p}) - R_b)^2}{\Delta_b^2}$$

where $n_b(\vec{p})$ is the number of events of bin b of the histogram, based on the data simulated by the Monte Carlo generator for the set of parameters $\vec{p}$, and R_b, based on the experimental data collected by the detector, while Δ_b^2 is given by

$$\Delta_b^2 = (\Delta_{\text{exper}}^{(b)})^2 + (\Delta_{\text{gener}}^{(b)})^2 = (\sqrt{R_b})^2 + (\sqrt{n_b})^2 = R_b + n_b,$$

and

$$b) \qquad \chi^2(\vec{p}) = \sum_{b=1}^{K} \frac{(\tilde{f}_b(\vec{p}) - R_b)^2}{\Delta_b^2}$$

where $\tilde{f}_b(\vec{p})$ corresponds to the number of events in bin b, estimated by Formula (5.1) or (5.2) (approximation of the Monte Carlo generator response), and

$$\Delta_b^2 = (\Delta_{\text{exper}}^{(b)})^2 + (\Delta_{\text{fbstat}}^{(b)})^2 + (\Delta_{\text{fbcal}}^{(b)})^2$$

where

$$(\Delta_{\text{exper}}^{(b)})^2 = (\sqrt{R_b})^2 = R_b,$$

$\Delta_{\text{fbstat}}^{(b)}$ and $\Delta_{\text{fbcal}}^{(b)}$ represent statistical and computational uncertainties, estimated by Formulas (6.52) and (6.27) respectively.

These χ^2 functions should have close minima for the same parameter sets.

For example, in the simplest case with only a single tuned parameter, $n = 1$, we use the scan along a line limited by the initial and final values of the parameter, or in the case of $n = 2$, we scan along the diagonal in the rectangle, a two-dimension parameter space, changing simultaneously the two tuned parameters p_1, p_2 by their respective steps h_1, h_2.

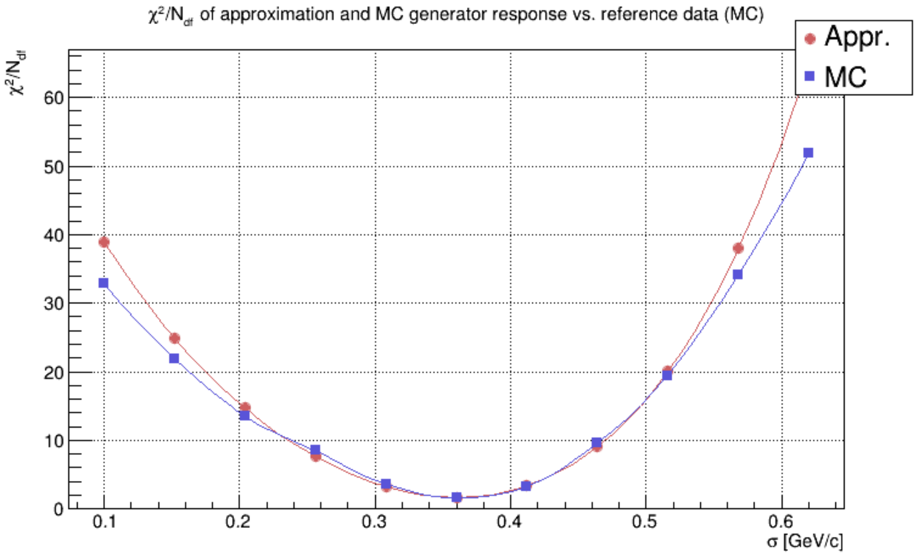


Figure 7.6: Example of a χ^2/N_{df} comparison plot of approximation (plotted in red) and Monte Carlo generator (plotted in blue) vs. reference data (default tune Monte Carlo), for a single parameter tuning (`PARJ(21)`) with 10^4 events per generated data sample. The horizontal axis corresponds to the value of the tuned parameter and the vertical axis - to the χ^2/N_{df} value. The points on the plot were connected by a simple poly-line for visual comfort.

For the chosen set of parameter values we can estimate the number of events $n_b(\vec{p})$ in bin b by using the polynomial (Formula (5.1) or (5.2))

$$n_b(\vec{p}) \approx \tilde{f}_b(\vec{p}) = f_b(\vec{p}) + \Delta_{fb}^{(b)}$$

and alternatively obtain the number from the Monte Carlo generator

$$\tilde{n}_b(\vec{p}) = n_b(\vec{p}) + \text{statistical generator error}$$

After calculating the χ^2, both results can be plotted as χ^2 divided by the number of degrees of freedom and can be compared. The optimal set of parameters values $\vec{p}_{\mathrm{min}}$ can be obtained by minimization (see Fig. 7.6).

Chapter 8

Numerical experiments and discussion of results

Quod erat faciendum (q.e.f.).
Lat.: Which was to be done.

This chapter presents the application of the fitting algorithm, described in Chapter 5. The algorithm is based on per-bin parameterization of the Monte Carlo generator response to parameter variations in n-dimensional parameter space and numerical optimization of the parameterized behavior. The presented results were obtained with the `Lagrange` application package which implements these techniques and tunes model parameters of the PYTHIA 6.4 event generator to the reference data, collected by LHCb, by interfacing to the LHCb software framework.

8.1 Parameter factorization strategy

Crosschecks between data, recorded by detectors, and theory, represented by Monte Carlo simulation, is a very common yet non-trivial task. Monte Carlo event generators, which are used to produce the simulated data, use complicated models to provide a full description of the simulated final states. Typically, these models have a large number of free or weakly constrained parameters, which must be tuned to better mimic the experimental data. It is practically impossible to tune all these parameters altogether. Even on a large super computer, using high parallelization techniques, it could take weeks or even months, depending on the number of simultaneously tuned parameters and the size of the simulated data sample. This already applies to the conventional approach when only the Monte Carlo generator level data is used, and in this work one of the key points is the use of the detector simulation for every generated event.

Additional computations, such as propagation of tracks in the material of the simulated detector, signal digitization, full event reconstruction, etc., require significant time to process. The local computational cluster of the LHCb group at Max Planck Institute for Nuclear Physics in Heidelberg, had 22 nodes dedicated to the numerical experiments described in this chapter.

Each node consisted of dual Intel® Xeon® CPUs with 4 physical cores each, which made an optimal number of 176 simulation jobs running in parallel on the cluster. Even having a 1:1 ratio of jobs per core, the GAUSS application on average required ≈ 1.5 minutes to produce a single event on each core (most of this time is consumed by GEANT4), and on average it took about ≈ 1.5 hours to complete the whole cycle of simulating, digitizing and reconstructing a 10^4 events data sample. Additional time costs may arise, depending on the total number of computational jobs per simulated data sample, since every job had a certain short delay time before it was executed, to reduce the immediate impact on the storage system of the cluster.

Due to the time-consuming event simulation procedure and limited computational resources, it was decided to limit the size of generated data samples to 10^4 events. Such size of data sample allow one to constrain only those parameters, which govern the overall event properties. Therefore this study focuses on the most basic parameters of the Lund string fragmentation and parton shower models, which are presented in Tab. 8.1, along with their default values for PYTHIA (v6.4) – the default Monte Carlo event generator used in the LHCb software framework.

Parameter	default value	description
`PARJ(21)`	0.36	σ
`PARJ(41)`	0.3	a
`PARJ(42)`	0.58	b
`PARJ(82)`	1	shower cut-off

Table 8.1: The selected fragmentation parameters: Lund string parameters a, b, σ and shower cut-off, and their default values. The parameters were described in detail in Section 4.2 of Chapter chap:ch4.

To prepare the tuning of the selected four parameters, it is necessary to:

1. select a set of distributions which will act as observables and will form the comparison criteria;
2. check how sensitive the selected distributions are with respect to the shifts of the tuned parameters in the parameter space;
3. analyze the influence of the number of the initial sampling points on the approximation of the Monte Carlo generator response;

These questions are discussed in detail in the next sections.

8.2 Comparison criteria

Another important step after selecting the tuned parameters, is the choice of comparison criteria between the simulated and reference data. The `Lagrange` tuning application uses as input sets of histograms which are produced by its analytical helper program. The analytical application was configured to use the following requirements on the data for the events and tracks selection:

- There must be no more than one primary vertex in the event;
- Only long tracks are used in the analysis (LHCb track types are described in Section 1.2 of Chapter 1);
- χ^2/N_{df} of the reconstructed track must be less than 4;
- The track's origin must be within a 10 mm distance from the beam and its z coordinate must be within $-250 < z < 250$ mm;

- The track's momentum must be greater than 2 GeV/c^2;
- The track's pseudorapidity η must be within $2 < \eta < 5$.

A sample of data recorded by the LHCb detector in proton-proton collisions at a centre-of-mass energy of $\sqrt{s} = 7$ TeV in 2010 was used as a reference data set. The data was triggered by the "MicroBias" trigger and had no prescaling (TCK 0x51710). The following LHCb software packages were used in the tuning process to produced simulated data samples:

- Gauss v40r0 for simulation;
- Boole v22r1 for digitization;
- Brunel v37r8p5 for data reconstruction;
- Moore v9r1p3 for trigger code application;
- DaVinci v28r2p2 for physics analysis;

These versions of the software packages fully recreate the same detector environment which was used during the data taking for the simulated data.

For this test with only four non-flavor hadronization parameters, the following distributions were selected to be used as the comparison criteria:

- η (pseudo-rapidity), in the range $2 < \eta < 5$, presented in Fig. 8.1;

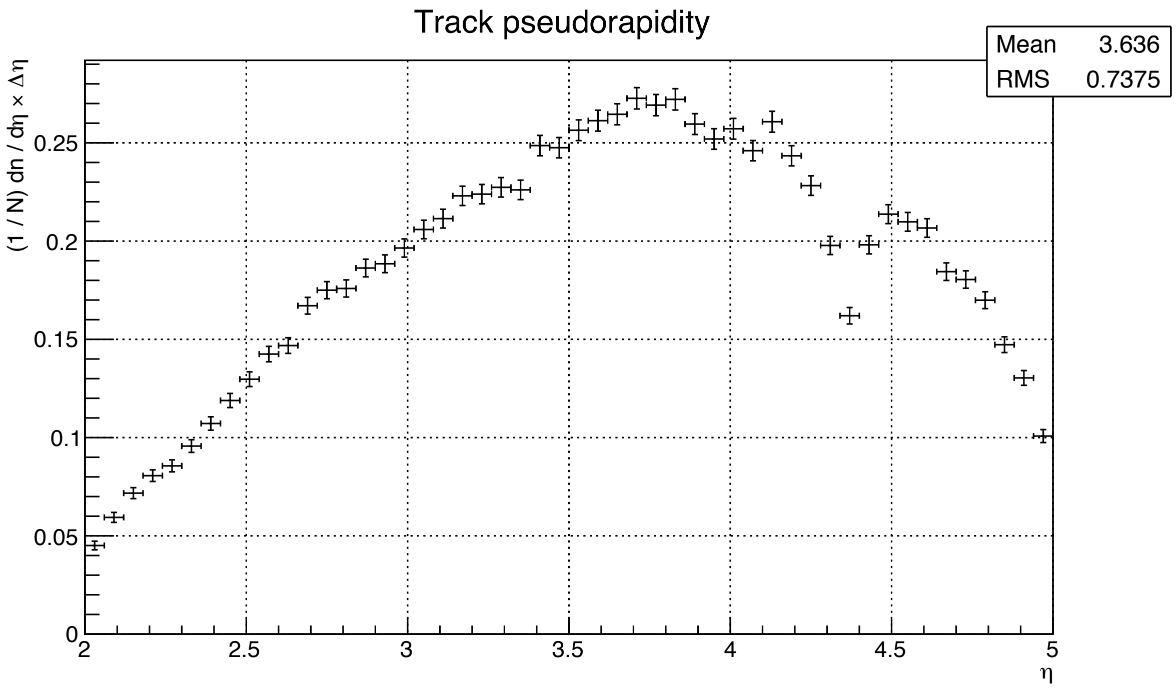


Figure 8.1: Track pseudorapidity distribution.

- p_{T} vs. η (pseudorapidity in bins of p_{T}), with $0 < p_{\mathrm{T}} < 3$ GeV/c^2 and $2 < \eta < 5$, plotted separately for positively and negatively charged tracks, presented in Fig. 8.2 through 8.7;

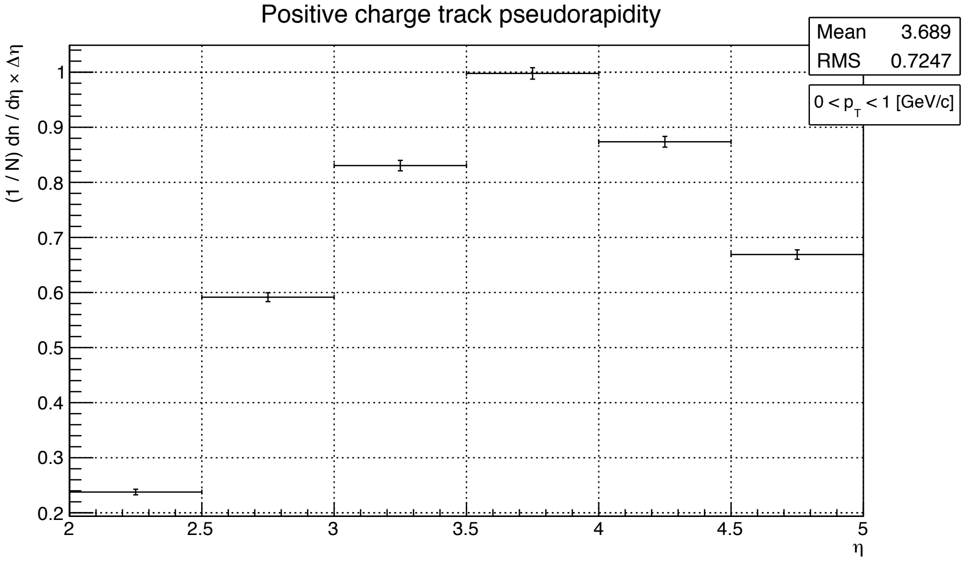


Figure 8.2: Pseudorapidity distribution for positively charged tracks with $0 < p_{\mathrm{T}} < 1$ GeV/c for $2 < \eta < 5$.

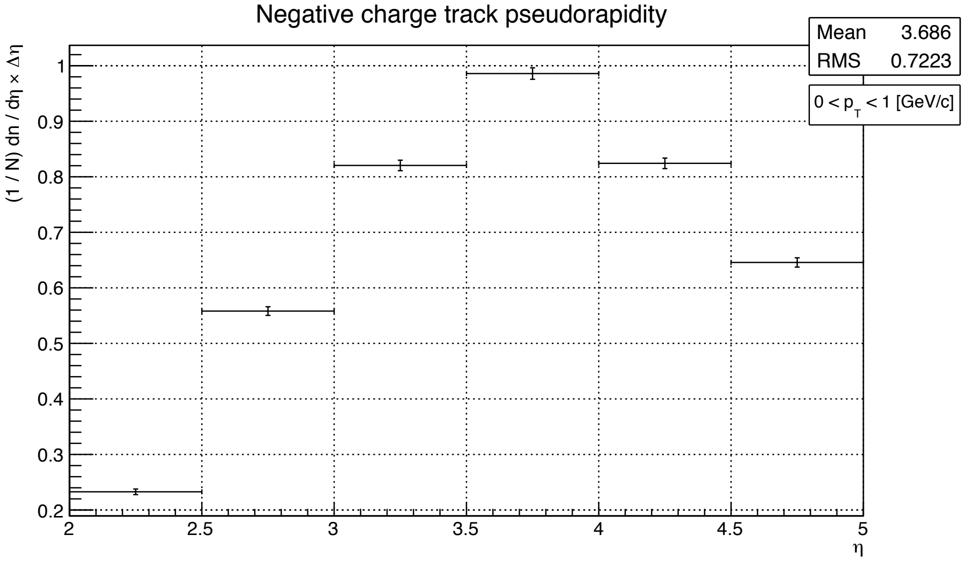


Figure 8.3: Pseudorapidity distribution for negatively charged tracks with $0 < p_{\mathrm{T}} < 1$ GeV/c for $2 < \eta < 5$.

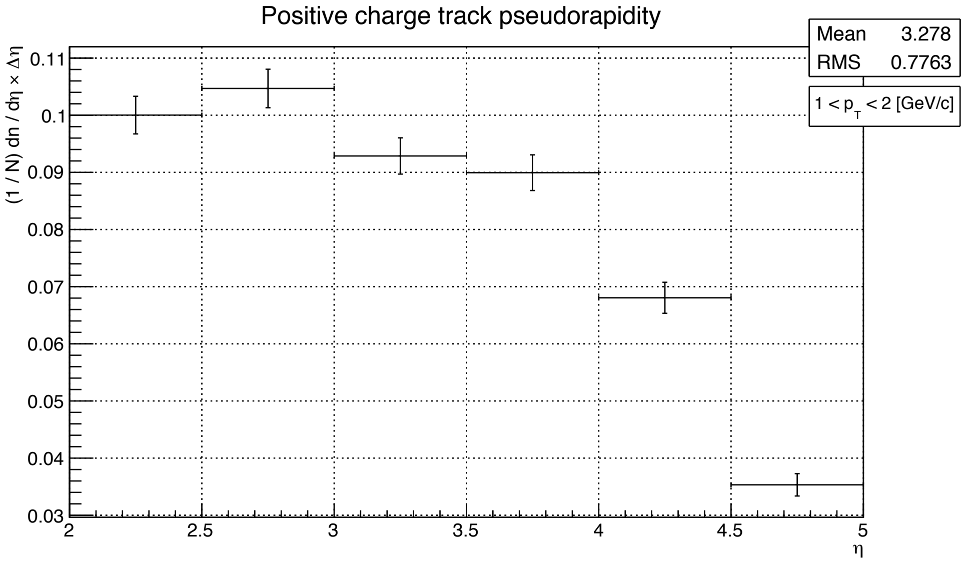


Figure 8.4: Pseudorapidity distribution for positively charged tracks with $1 < p_{\mathrm{T}} < 2$ GeV/c for $2 < \eta < 5$.

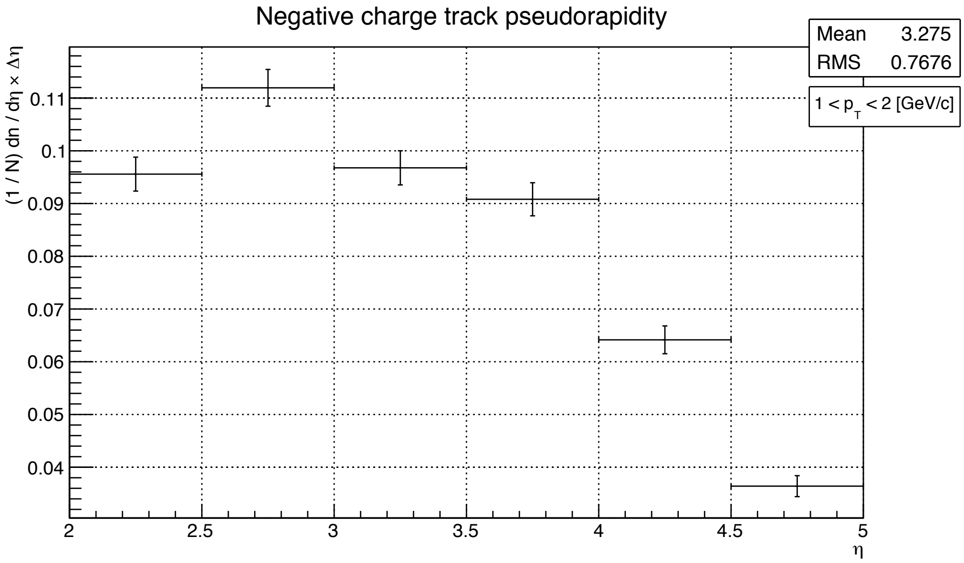


Figure 8.5: Pseudorapidity distribution for negatively charged tracks with $1 < p_{\mathrm{T}} < 2$ GeV/c for $2 < \eta < 5$.

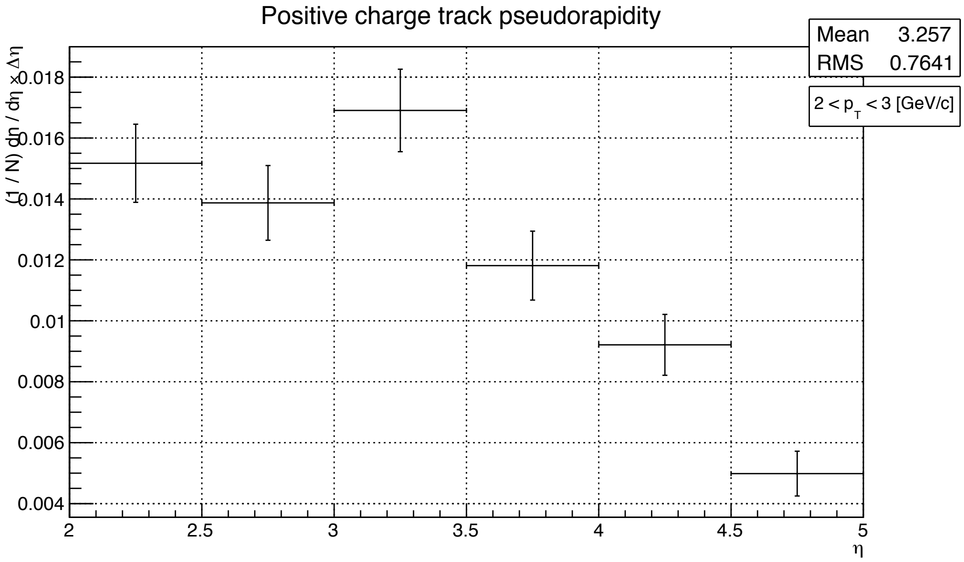


Figure 8.6: Pseudorapidity distribution for positively charged tracks with $2 < p_{\mathrm{T}} < 3$ GeV/c for $2 < \eta < 5$.

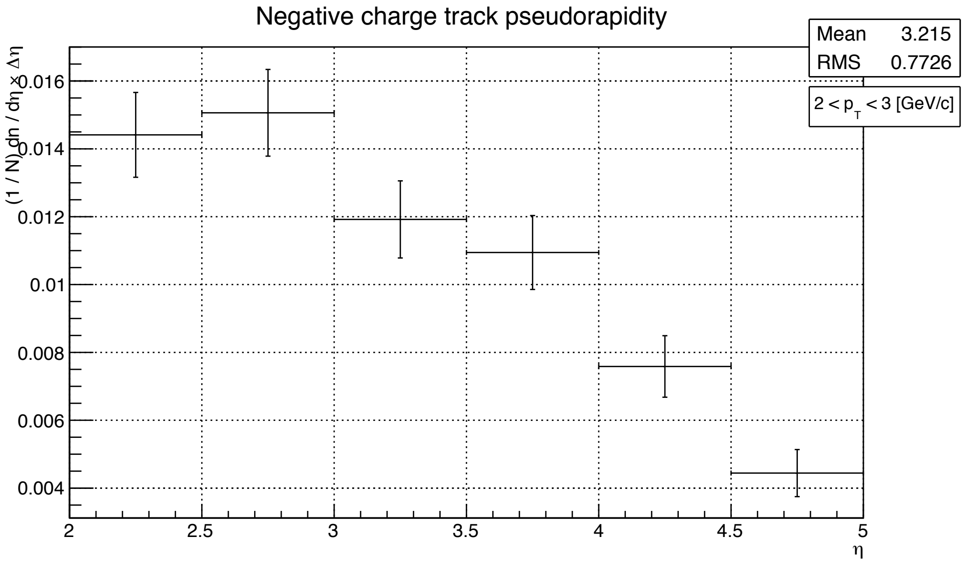


Figure 8.7: Pseudorapidity distribution for negatively charged tracks with $2 < p_{\mathrm{T}} < 3$ GeV/c for $2 < \eta < 5$.

- Multiplicities of positively and negatively charged tracks in the the η regions of $2 < \eta < 3$, $3 < \eta < 4$ and $4 < \eta < 5$, presented below in Fig. 8.8 to 8.13.

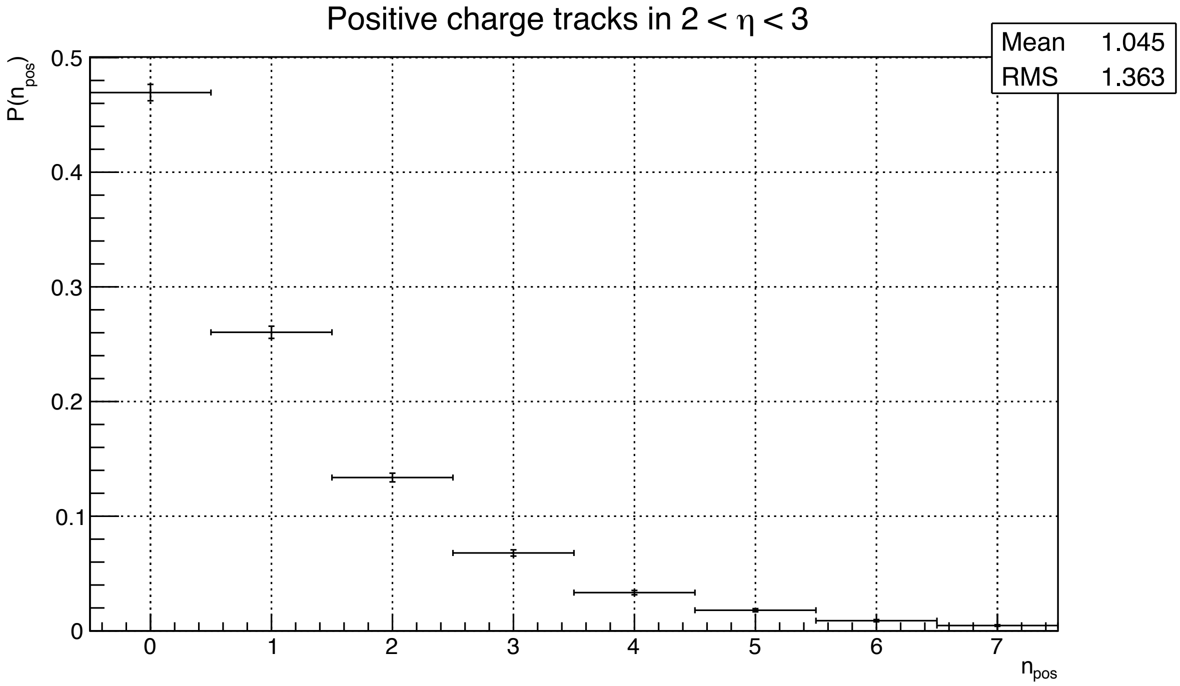


Figure 8.8: Multiplicity of positively charged tracks for $2 < \eta < 3$.

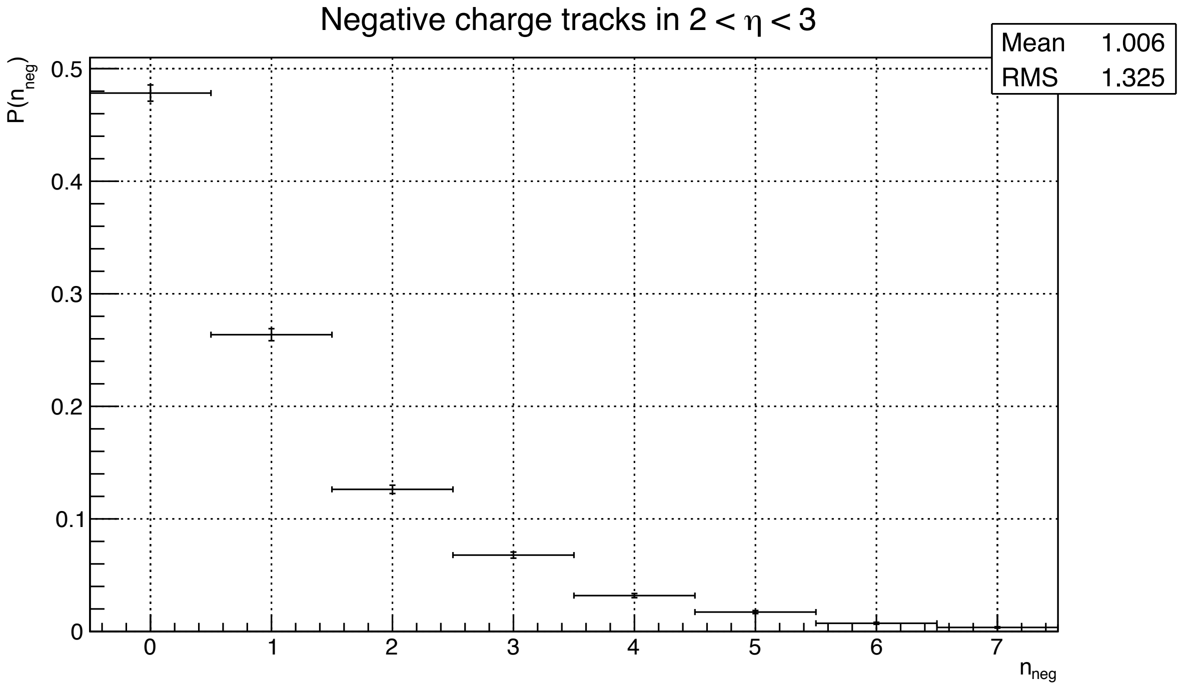


Figure 8.9: Multiplicity of negatively charged tracks for $2 < \eta < 3$.

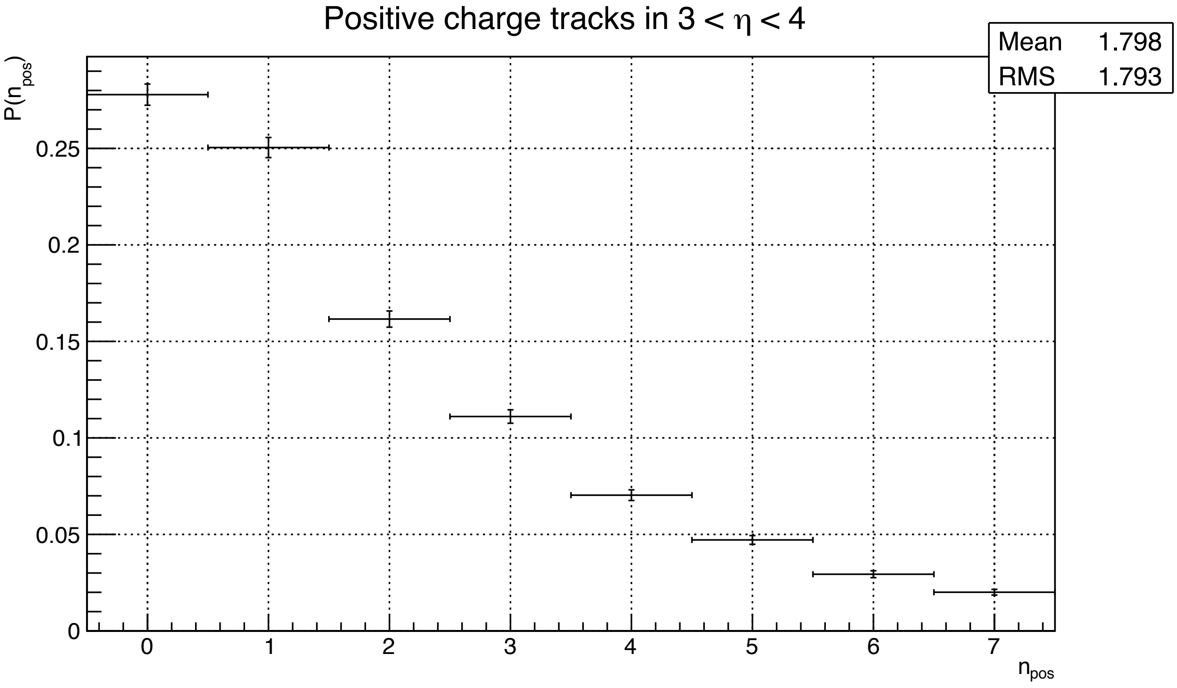


Figure 8.10: Multiplicity of positively charged tracks for $3 < \eta < 4$.

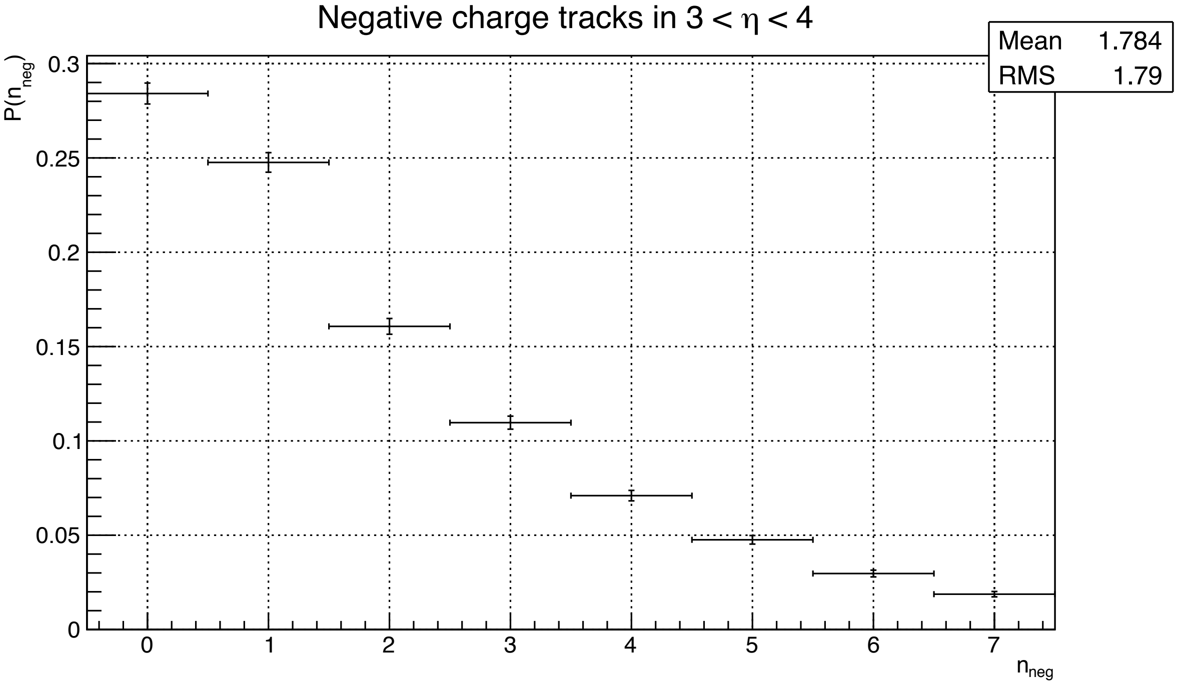


Figure 8.11: Multiplicity of negatively charged tracks for $3 < \eta < 4$.

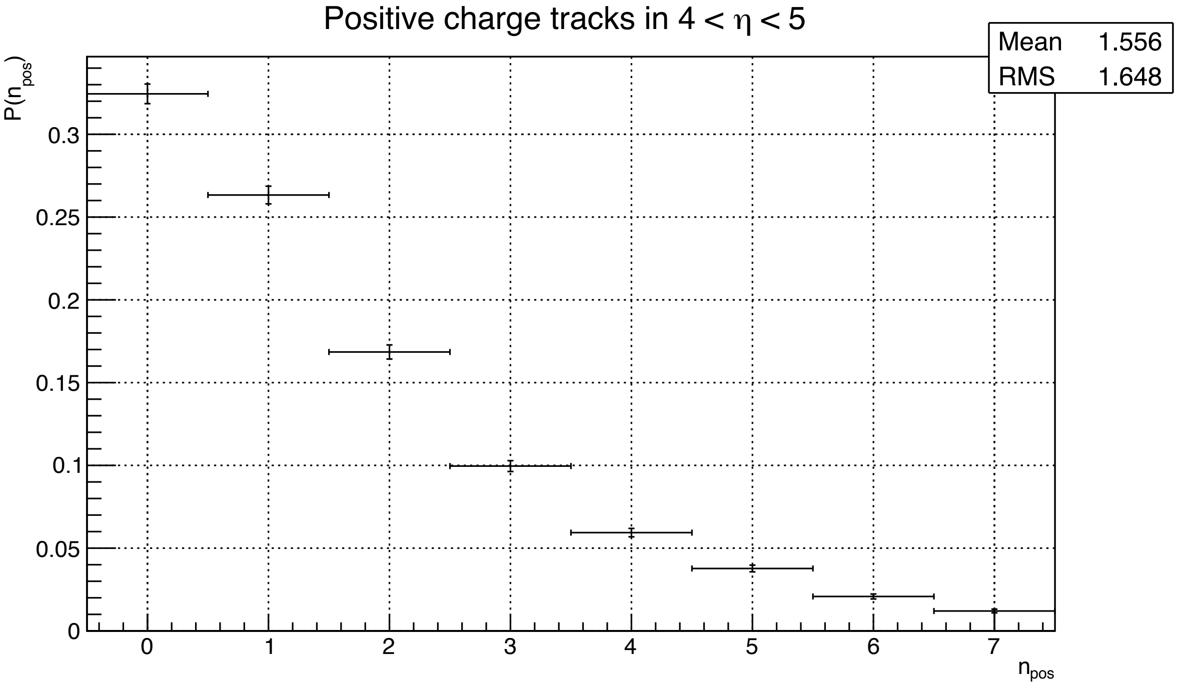


Figure 8.12: Multiplicity of positively charged tracks for $4 < \eta < 5$.

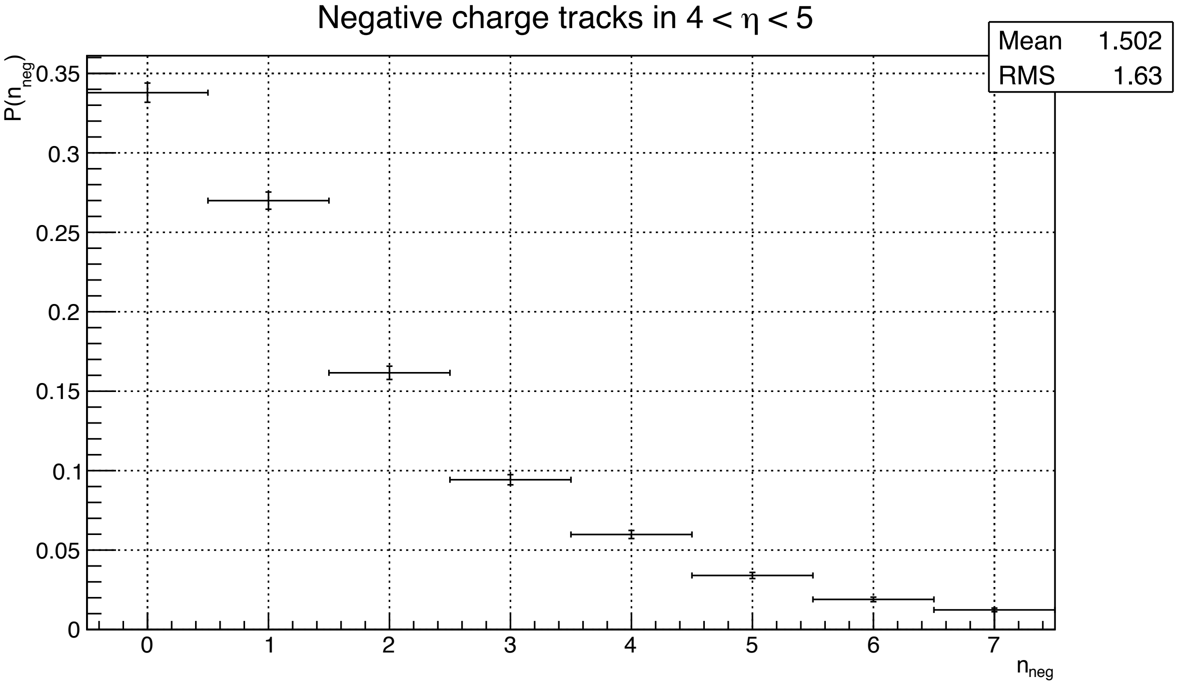


Figure 8.13: Multiplicity of negatively charged tracks for $4 < \eta < 5$.

The distributions presented above are merged into a data vector and used as a criterion in the tuning application. Fig. 8.14 shows a comparison of such two criteria histograms, built for the reference data sample (LHCb recorded data) and a sample of Monte Carlo data, simulated with default settings of the generator.

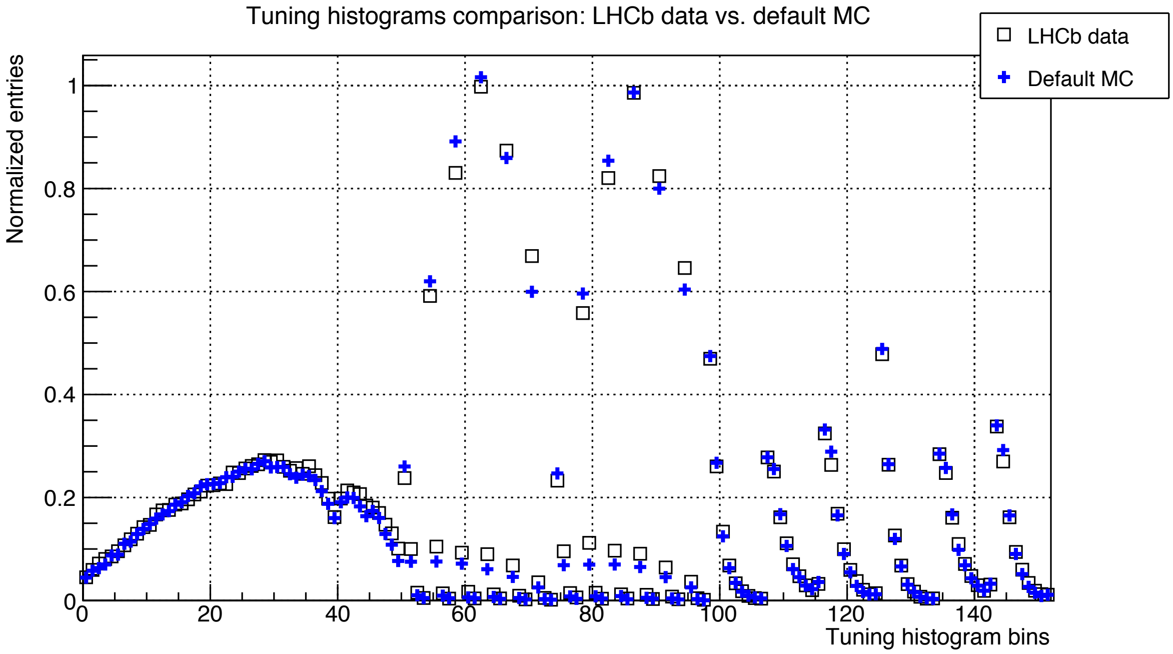


Figure 8.14: Comparison plot of criteria histograms for the reference (LCHb recorded) and default tune Monte Carlo data samples, as they are represented in the tuning application.

8.3 Setup of the tuning application

To prepare the tuning application, it is necessary to define the following options:

1. The total number of tuned parameters n;
2. Define the tuned PYTHIA parameters: their names, starting values and variation limits;
3. Choose the order of the approximation polynomial: 2^{nd} or 3^{rd} order;
4. Define the number of events N_{gen} to generate per data sample;
5. Fix the number of the initial sampling points: the default minimal number N_{min} is calculated by Formula (5.3) and could be increased (see Tab. 5.1);
6. Select the method of solving the matrix equation: via SVD or via the normal system of equations;
7. Choose whether to estimate statistical and computational uncertainties;
8. Define the number of bins K of the comparison histogram;

It is clear that some of these options strongly influence the computation time of the application. The possibility to have a large number of events per generated data sample to increase statistics is limited by the computation time, which increases even more with the growth of the number of tuned parameters.

The computation time was very important in the presented numerical experiments since it was necessary to produce many different variants of tunings to estimate how different factors influence the result and for debugging purposes. Thus the number of generated events per data sample was chosen to be $N_{\text{gen}} = 10^4$ as a compromise between statistical precision and computational time. The analysis of the influence of the order of the approximation polynomial showed that the choice of the 3$^{\text{rd}}$ order polynomial over the 2$^{\text{nd}}$ order one has little effect on the final result, but significantly increases the computation time.

Hence, in the presented numerical experiments the approximation was done with a 2$^{\text{nd}}$ order polynomial, the number of generated events per data sample was chosen to be $N_{\text{gen}} = 10^4$ and the number of the initial sampling points was set to be $N = 2N_{\text{min}}$. The comparison criteria, which from the point of the tuning application is a set of histograms of the chosen distributions glued together in one long data vector, was 152 bins long and the system of matrix equations was solved with the SVD method (in this case, estimation of the coefficients via a normal system yielded the same result).

8.4 Sensitivity test

It is possible to use the `Lagrange` tuning application to study the effect of the shift of the tuned parameters values on the parameterization.

In order to estimate of how sensitive the selected distributions are with respect to the change of the parameter values, the parameters were varied one at a time. The limits for the parameter values were set approximately to two times that of the corresponding default value to probe regions further away from it (Tab. 8.2). A Monte Carlo data sample produced with default LHCb software settings for PYTHIA was used as a reference. The Monte Carlo generator response was approximated by a 2$^{\text{nd}}$ order polynomial and used oversampling of $2N_{\text{min}}$, where N_{min} is calculated according to Formula (5.3) in Chapter 5.

Parameter	default value	lower limit	upper limit	description
`PARJ(21)`	0.36	0.1	0.62	σ
`PARJ(41)`	0.3	0.1	1	a
`PARJ(42)`	0.58	0.1	1.5	b
`PARJ(82)`	1	0.1	2	shower cut-off

Table 8.2: Default values and limits of the tuned parameters during the sensitivity test.

The plots below present the dependencies of χ^2/N_{df} of the approximation and Monte Carlo generator response vs. reference data, calculated for the shifts in parameter space for single parameters and groups of parameters. The horizontal axis corresponds to the coordinates of the scan points along an arbitrary line in the n-dimensional parameter space. For a single parameter case, the scan is done along the line in a one-dimensional parameter space, confined to the boundaries of the parameter, and passes through the parameter's default value.

The test for sensitivity of the observables to shifts of parameters in the parameter space allows one to understand how much the distributions change when the parameters are varied in their limits. Naturally, it is best to select observables which are the most sensitive to parameter changes. If the tuned parameters have no effect on an observable, they can be removed from the comparison criteria or replaced with different ones.

8.4.1 Parameter PARJ(21)$= \sigma$

The PYTHIA parameter PARJ(21) corresponds to the width σ in the Gaussian p_x and p_y transverse momentum distribution, relative to the color string. It is a free parameter and in the LHCb software distribution its default value is set to PARJ(21)$= 0.36$, as stated in Tab. 8.1. Fig. 8.15 presents the result of the scanning procedure for a single parameter PARJ(21) within the limits defined in Tab. 8.2. The vertical axis corresponds to the reduced χ^2 value, and the horizontal axis corresponds to the parameter value. The points on the plot are connected by a spline line for visual comfort. The reduced χ^2 for these tests was calculated taking into account statistical uncertainties of the reference data, ignoring statistical errors of the approximation and Monte Carlo generator respectively, because of the effects discussed further in Sec. 8.6.

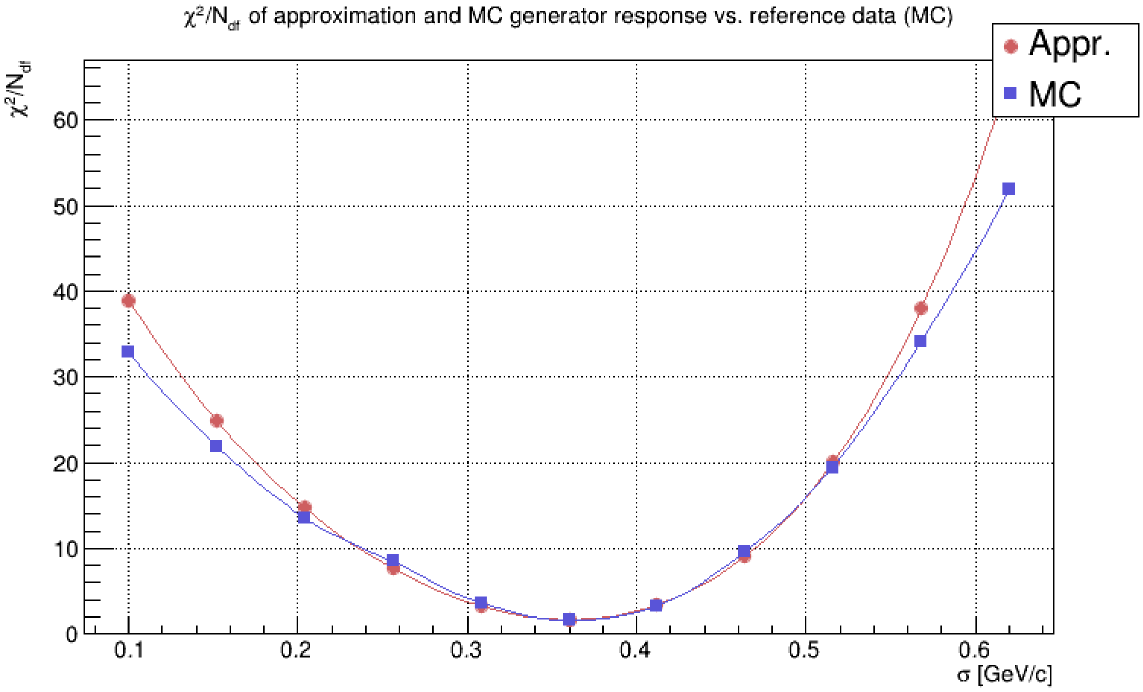


Figure 8.15: χ^2/N_{df} plot of the sensitivity test of parameter PARJ(21): round markers (red curve) correspond to the response of the approximation formula, square markers (blue curve) represent the Monte Carlo event generator response.

It can be seen that in the parameter value interval $[0.2, 0.5]$ the approximation (red) agrees well with the Monte Carlo generator response (blue) and has a minimum in the point with the parameter default value PARJ(21)$= 0.36$.

8.4.2 Parameters PARJ(41)$= a$ and PARJ(42)$= b$

In PYTHIA, the Lund symmetric string fragmentation function (4.4) has two free parameters a, b (discussed in Sec. 4.2 of Chapter 4) which need to be adjusted to bring the fragmentation into accordance with the reference data. These parameters are defined as PARJ(41)$= a$ and PARJ(42)$= b$ and their default values are set to PARJ(41)$= 0.3$ and PARJ(42)$= 0.58$ GeV^{-2}.

Figures 8.16 and 8.17 present the results of separate tunings of these two parameters within the limits defined in Tab. 8.2.

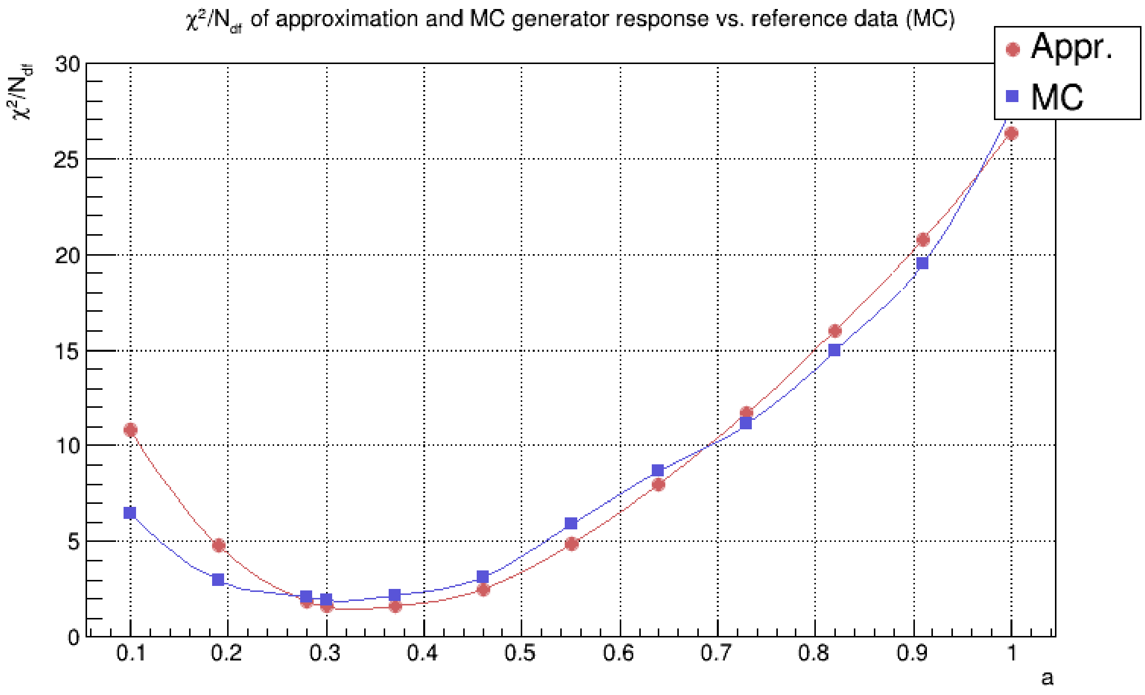


Figure 8.16: χ^2/N_{df} plot of the sensitivity test of parameter `PARJ(41)`.

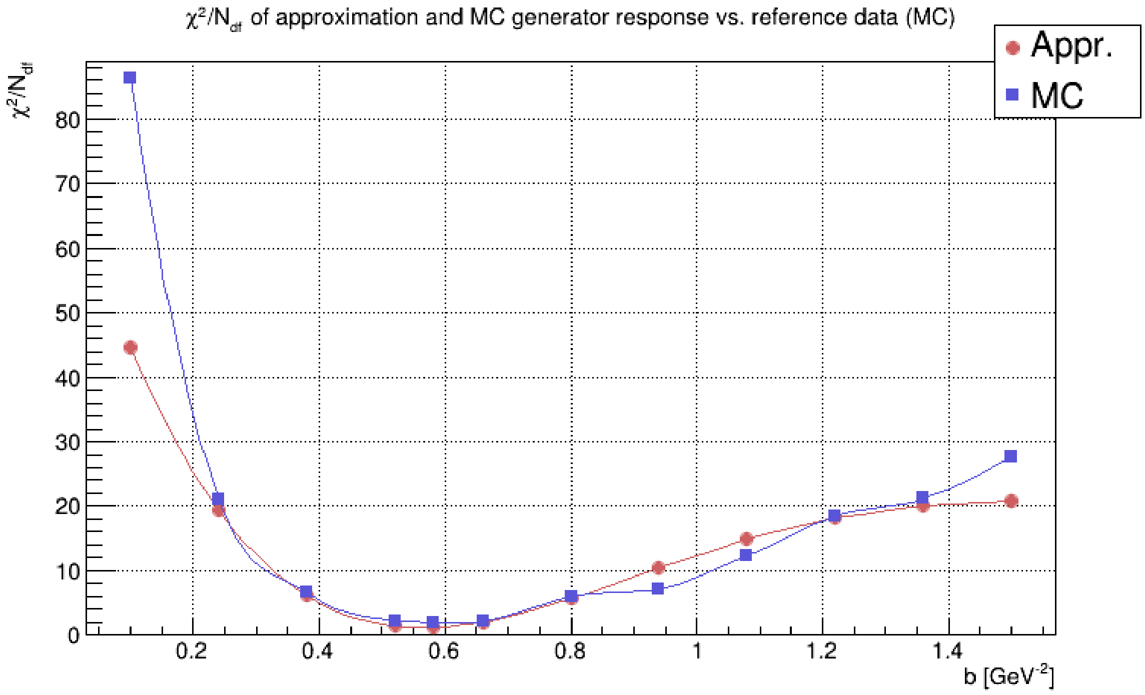


Figure 8.17: χ^2/N_{df} plot of the sensitivity test of parameter `PARJ(42)`.

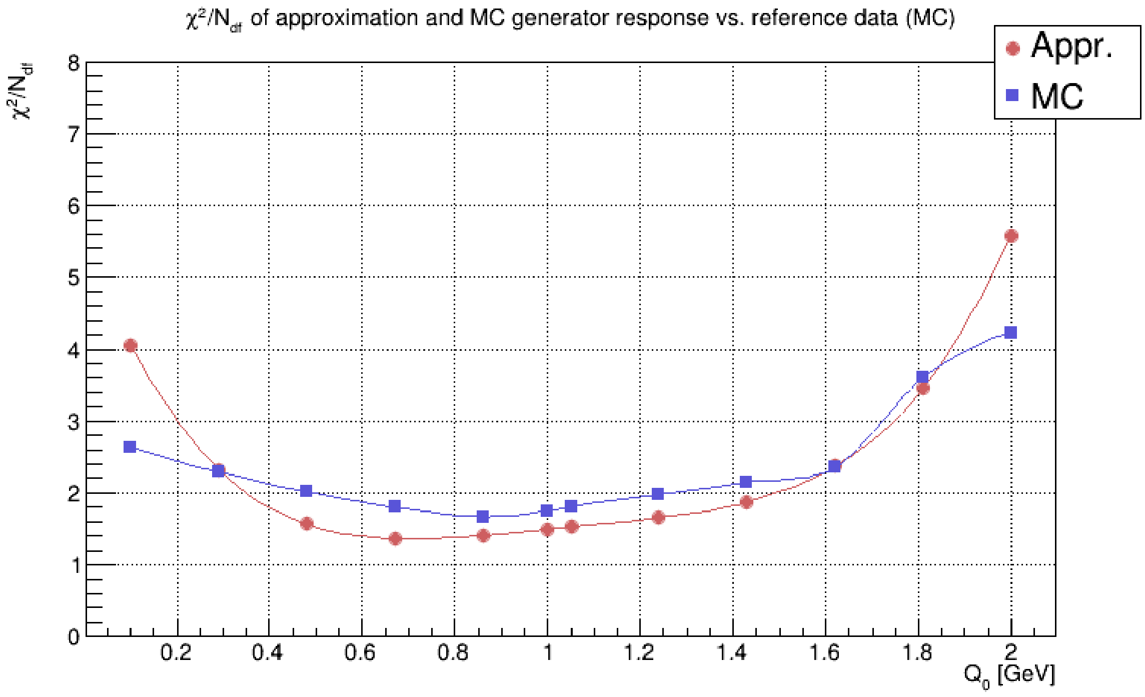


Figure 8.18: $\chi^2/N_{\rm df}$ plot of the sensitivity test of parameter PARJ(82).

In the case of PARJ(41), there is a discrepancy between the approximation and the Monte Carlo generator in the region of the minimum, which is due to statistical fluctuations, and the difference is within statistical errors. For PARJ(42) the agreement is better, especially in the region of the minimum.

8.4.3 Parameter PARJ(82) - shower cut-off

In the PYTHIA parton shower model, the shower evolution is cut off at some lower scale $Q_0 = Q_{\text{shower cut−off}}$. This value is defined by parameter PARJ(82)='shower cut-off', with its default value set to PARJ(82)= 1 GeV.

The result for the PARJ(82) test within the limits presented in Tab. 8.2 is shown in Fig. 8.18.

It can be seen that in comparison to the other three tested parameters, PARJ(82) is the least sensitive one: it remains practically constant across a large part of the scanning interval. The discrepancy between the approximation and Monte Carlo generator response is within statistical errors.

8.4.4 Variation of two parameters PARJ(21) and PARJ(42)

The tuning of single parameters showed that parameter PARJ(21) seems to be the most sensitive out of the four chosen parameters, followed by PARJ(42) and PARJ(41), while PARJ(82) remains practically constant over the wide part of the variation interval.

Figure 8.19 presents the results of the simultaneous variation of the two more sensitive parameters PARJ(21) and PARJ(41). Since in this case there are two parameters being varied simultaneously, the scan is done along a diagonal of a rectangle in the 2D parameter space. The coordinates of the vertices of the rectangle are defined by the corresponding parameter limiting value. The main diagonal is divided by a user-defined number of steps, which in this case was

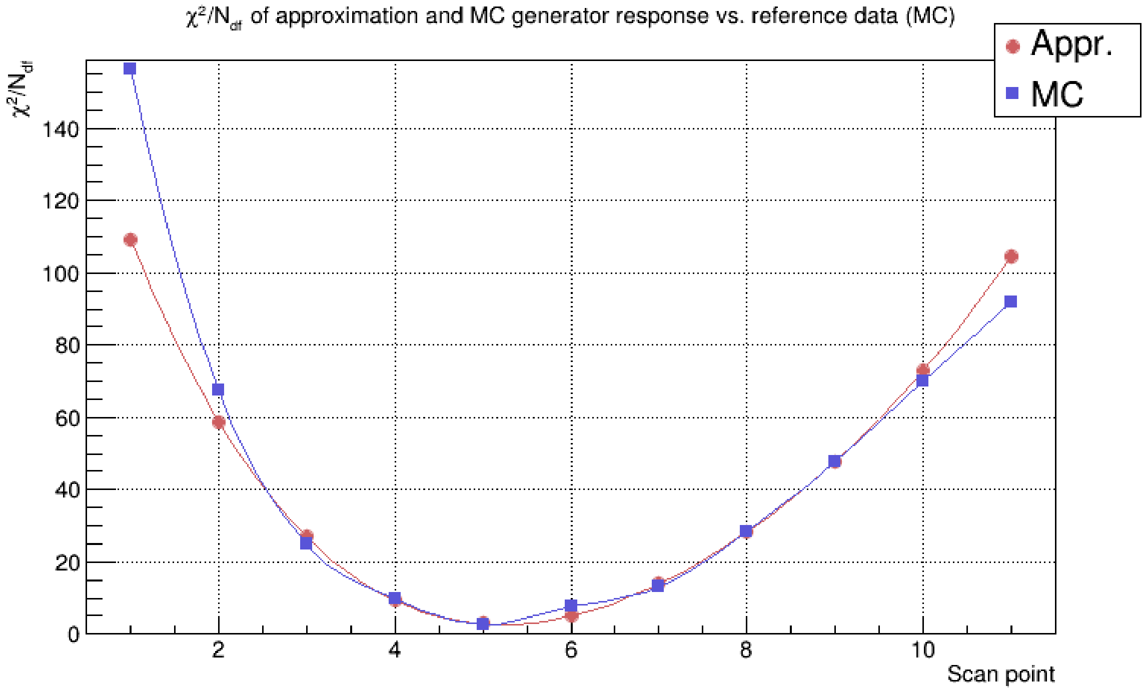


Figure 8.19: $\chi^2/N_{\rm df}$ plot of the sensitivity test of simultaneous tuning of parameters PARJ(21) and PARJ(42).

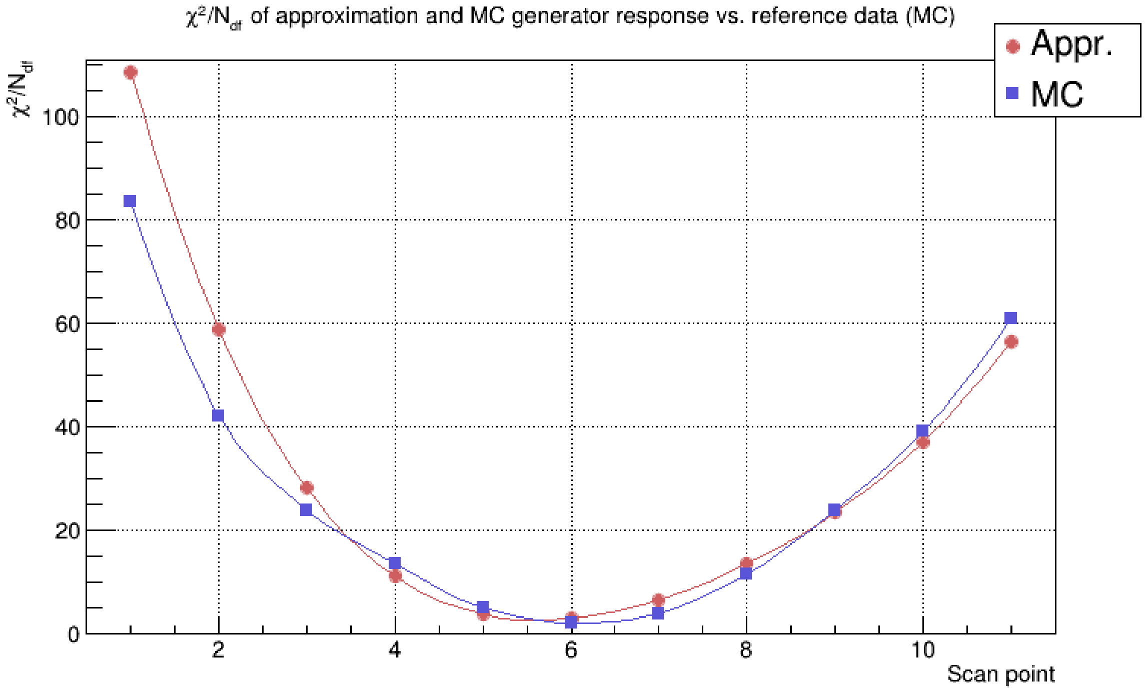


Figure 8.20: $\chi^2/N_{\rm df}$ plot of the sensitivity test of simultaneous tuning of parameters PARJ(21), PARJ(42) and PARJ(41).

equal to 10 and resulted in 11 sampling points. In each point a vector of corresponding parameter values is formed and for it the responses of the approximation and Monte Carlo generated are obtained. Thus, for the cases of 2 or more parameters being varied simultaneously, the horizontal axis on the reduced χ^2 plots corresponds to the scanning points in the parameter space.

The approximation agrees very well with the Monte Carlo generator in the interval between points 3 and 10, with a minimum in point 5.

8.4.5 Variation of three parameters PARJ(21), PARJ(42) and PARJ(41)

Figure 8.20 shows the simultaneous tuning of three parameters PARJ(21), PARJ(42) and PARJ(41). The approximation is best in the interval from point 4 to 10, with a minimum between points 5 and 6.

8.4.6 Variation of PARJ(21), PARJ(42), PARJ(41) and PARJ(82)

Figure 8.21 presents the result of the simultaneous tuning of all four parameters PARJ(21), PARJ(42), PARJ(41) and PARJ(82) in the limits defined in Tab. 8.2. Points on the horizontal axis correspond to points in parameter space.

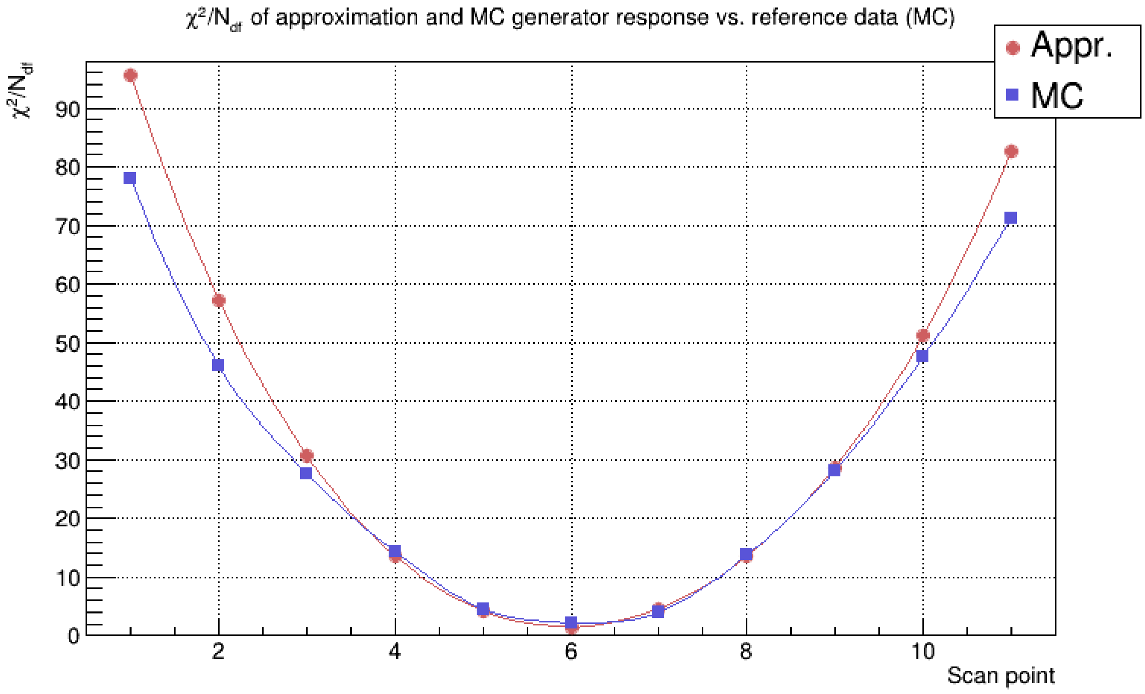


Figure 8.21: $\chi^2/N_{\rm df}$ plot of the sensitivity test of simultaneous tuning of four parameters PARJ(21), PARJ(42), PARJ(41) and PARJ(82).

The plot shows that the approximation agrees very well with the Monte Carlo generator response in the region of the minimum, between points 4 and 9.

Summary. Figures 8.15 to 8.17 and 8.19 to 8.21 show that the selected observables are sensitive to the tuned parameters. Fig. 8.18 proves that the observables weakly depend on parameter PARJ(82). In general, the studies show that the approximation agrees very well with the Monte Carlo generator response.

8.5 Overdetermination influence

The tuning application approximates the number of events n_b in every bin b of the compared histograms by a 2nd order polynomial, where the number of the initial anchor points (sampling points in the parameter space) influences the quality of the approximation. The purpose of the overdetermination test was to study this effect and select an optimal number of anchor points.

The tunes were done for a single parameter PARJ(21). The generated data samples were 10^4 events large and a sample of data of the same size, recorded by LHCb detector, was used as a reference. In the case of one parameter, from the Tab. 5.1 it follows that the number of unknown coefficients for the 2nd order polynomial equals to $N_{\mathrm{min}} = 3$.

Figure 8.22 to 8.26 present dependencies of χ^2/N_{df} of approximation and Monte Carlo generator response vs. reference data for different numbers of the initial sampling points. Values on the horizontal axis correspond to the values of parameter PARJ(21), tuned within the limits defined in Tab. 8.2.

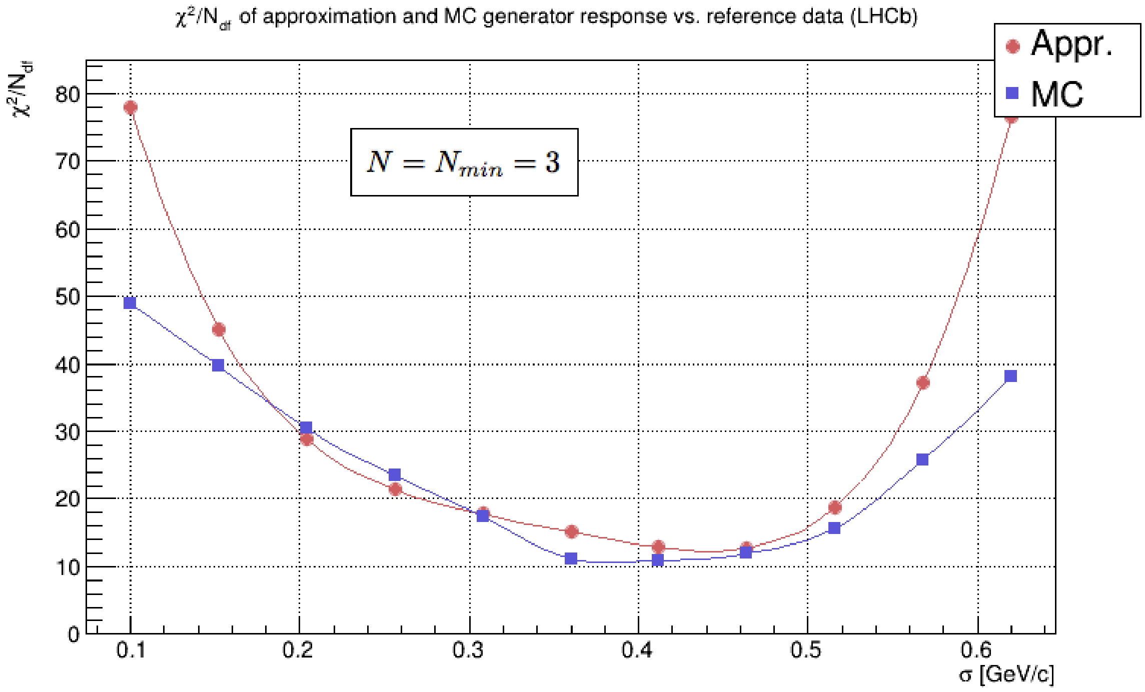


Figure 8.22: χ^2/N_{df} plot of tuning a single parameter PARJ(21) with the number of the initial sampling points equal to $N_{\mathrm{min}} = 3$.

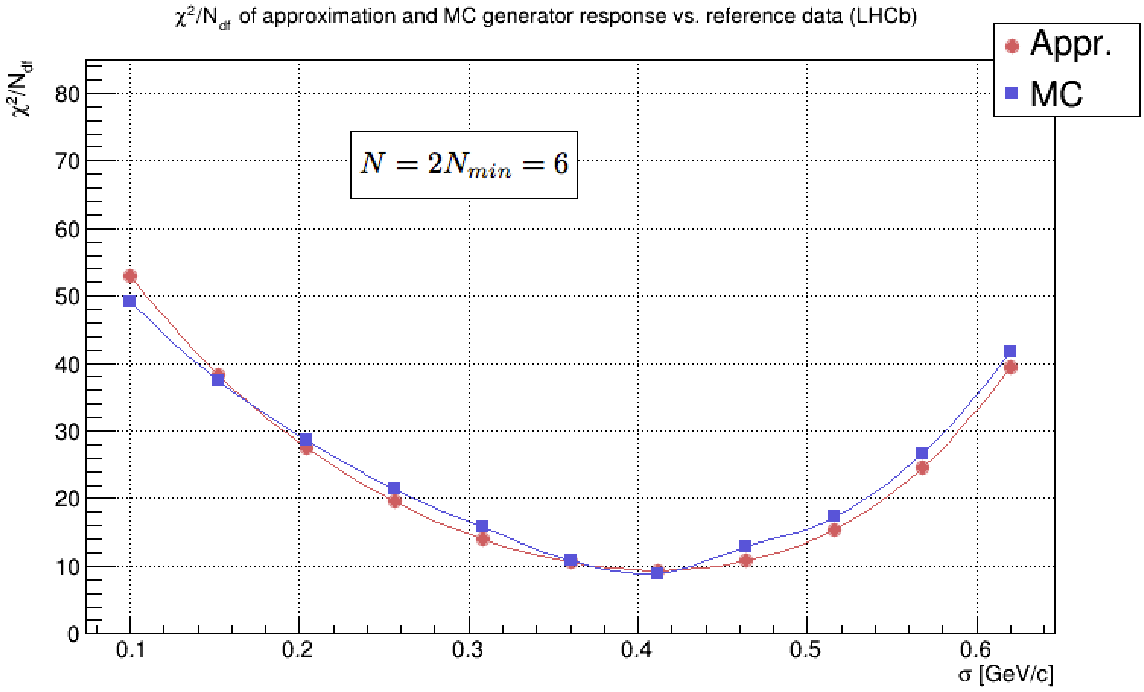


Figure 8.23: χ^2/N_{df} plot of tuning a single parameter `PARJ(21)` with the number of the initial sampling points equal to $2N_{\text{min}} = 6$.

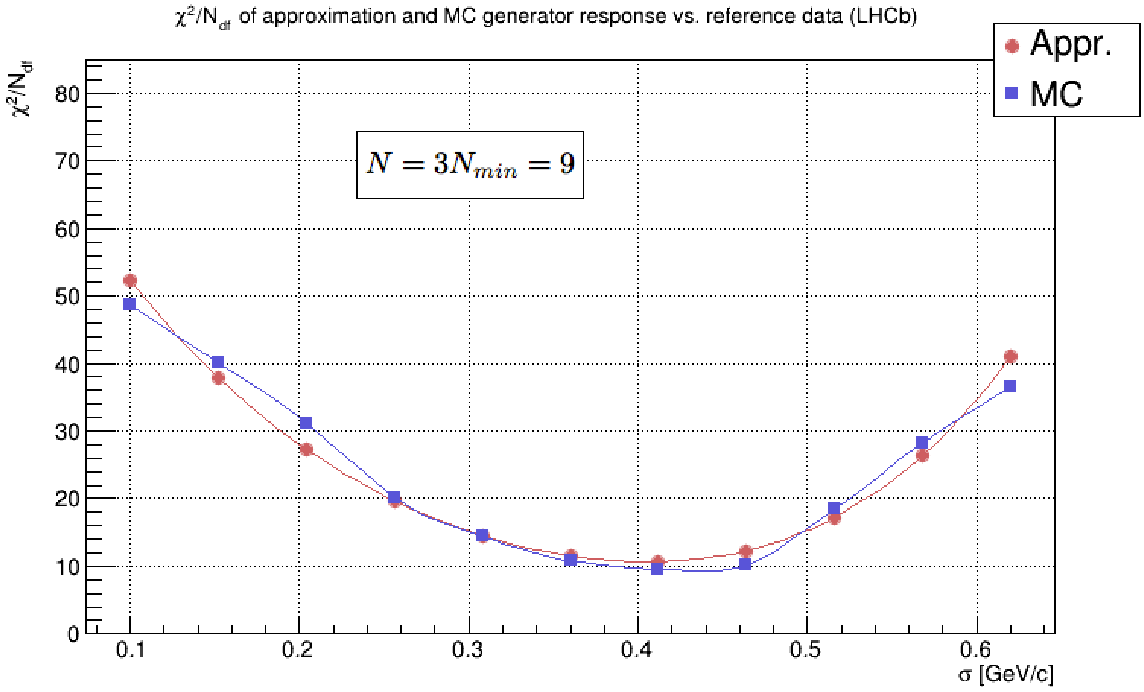


Figure 8.24: χ^2/N_{df} plot of tuning a single parameter `PARJ(21)` with the number of the initial sampling points equal to $3N_{\text{min}} = 9$.

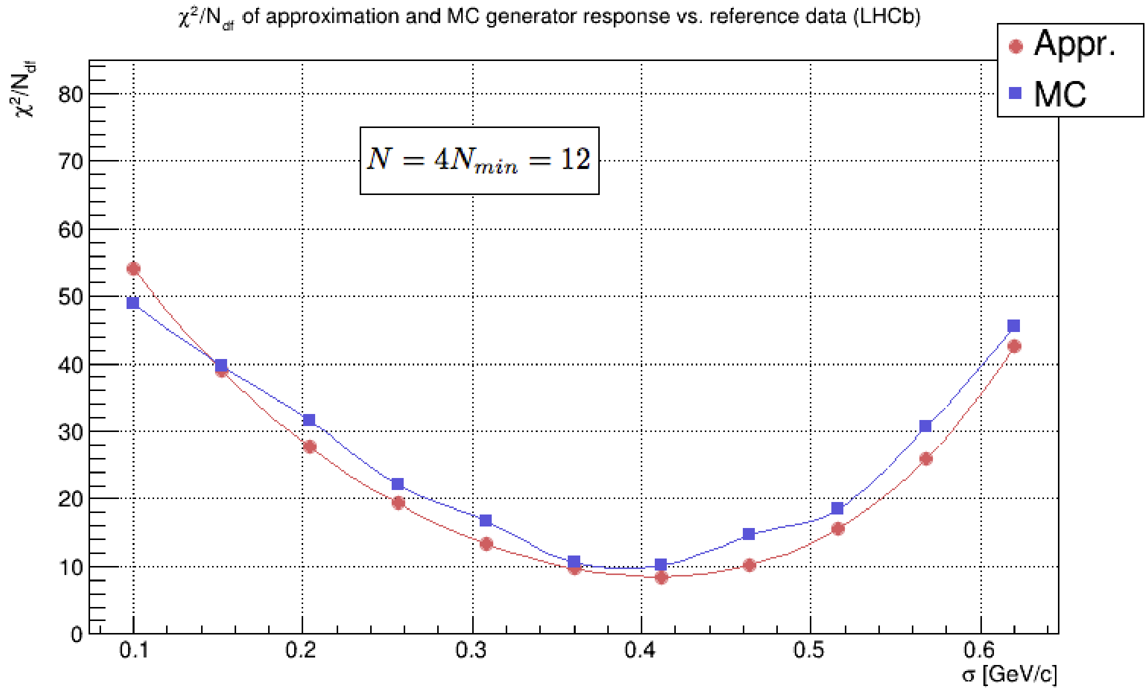


Figure 8.25: χ^2/N_{df} plot of tuning a single parameter PARJ(21) with the number of the initial sampling points equal to $4N_{\mathrm{min}} = 12$.

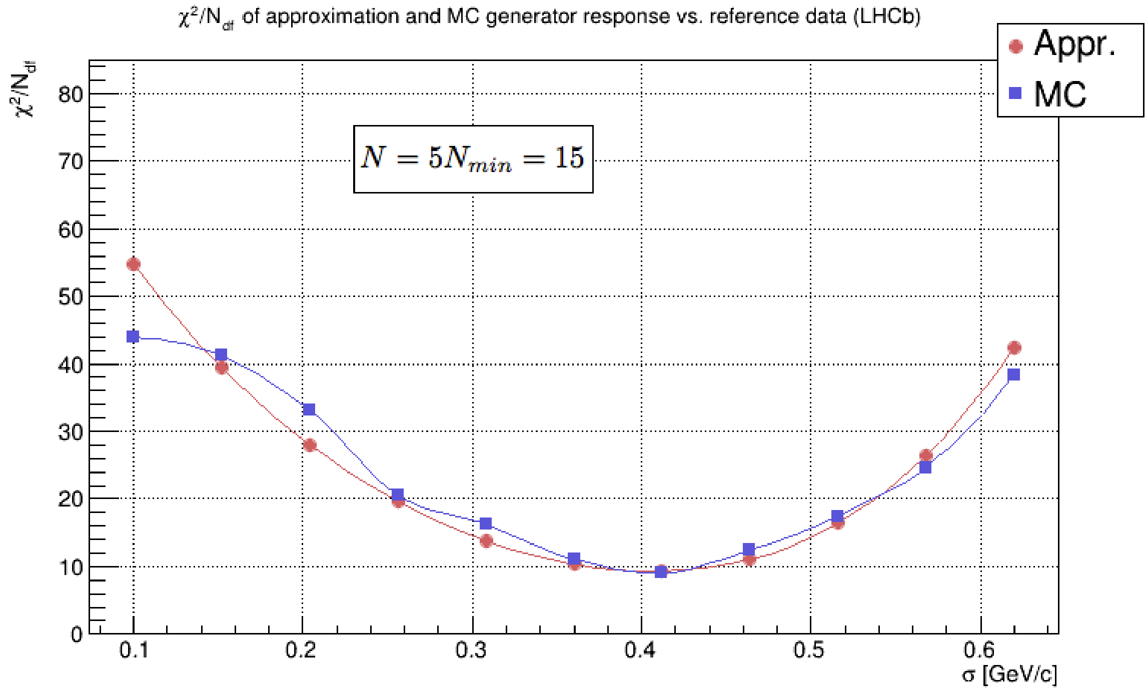


Figure 8.26: χ^2/N_{df} plot of tuning a single parameter PARJ(21) with the number of the initial sampling points equal to $5N_{\mathrm{min}} = 15$.

The test shows that the increase in the number of initial sampling points from N_{min} to $2N_{\text{min}}$ significantly improves the approximation quality. Increasing it further becomes less beneficial since a relatively smaller improvement in the approximation is accompanied by a significant increase in computational time. Thus $2N_{\text{min}}$ is good compromise between the accuracy of the fit and the computational time and it is used for the main tuning. This is consistent with similar studies and the recommendations in reference [51].

8.6 Main tuning

The main tuning was done with the help of the `Lagrange` application package for the four parameters `PARJ(21)`, `PARJ(41)`, `PARJ(42)` and `PARJ(82)`. Their default values and limits are presented in Tab. 8.2. The approximation was done with a 2nd order polynomial and as it follows from Tab. 5.1 of Chapter 5, the number of unknown coefficients in this case is $N_{\text{min}} = 15$. The number of initial sampling points was chosen to be $N = 2N_{\text{min}} = 30$.

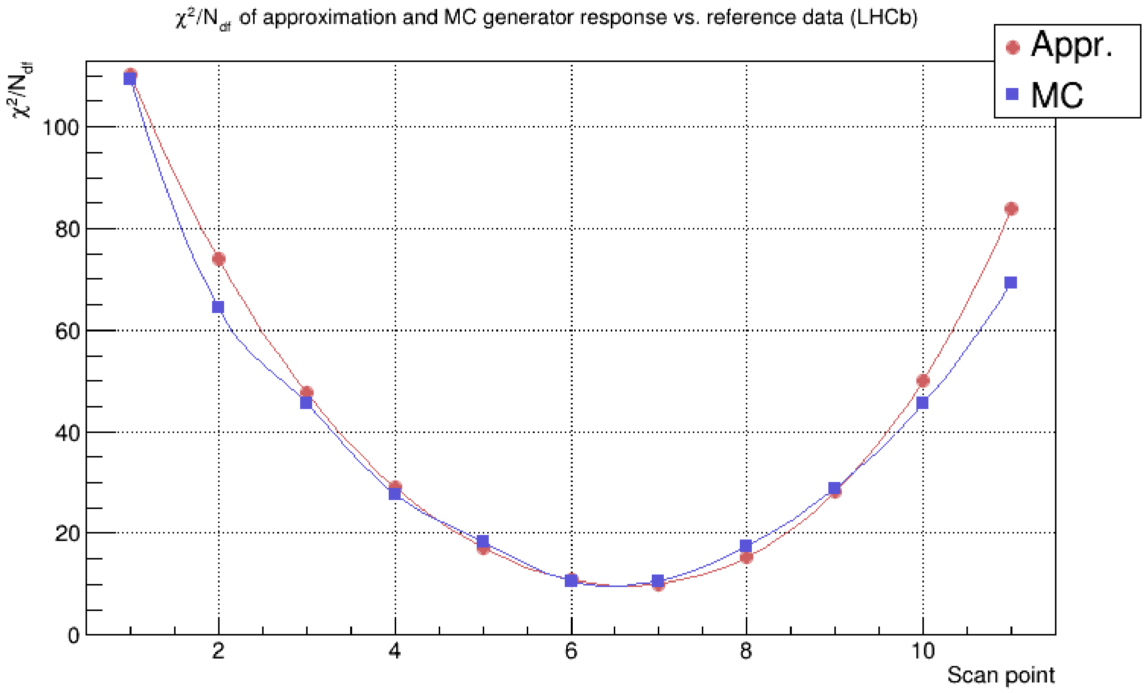


Figure 8.27: χ^2/N_{df} plot of simultaneous tuning of four parameters `PARJ(21)`, `PARJ(42)`, `PARJ(41)` and `PARJ(82)`. Red points correspond to the approximation, blue points to the Monte Carlo generator response. The approximation was done with a 2nd order polynomial and the number of initial sampling points is equal to $2N_{\text{min}} = 30$. The simulated datasets size is 10^4 events. The χ^2 values were calculated taking into account only statistical uncertainties of the reference data, excluding statistical errors of the approximation and Monte Carlo generator for the corresponding points.

Figure 8.27 demonstrates the result of an arbitrary scan, conducted after all the coefficients of the approximation polynomial have been estimated. The scan was done along one of the diagonals of the hypercube in the parameter space. The plotted dependencies are the χ^2/N_{df} of the approximation (red points) and Monte Carlo generator response (blue points) vs. reference data. The vertical axis corresponds to the $\chi^2/N_{\rm df}$ value and the horizontal one represents the scanning points along the line.

This scan test shows that the parameterization and the Monte Carlo generator response are in a very good agreement, especially in the region of the minimum. The high $\chi^2/N_{\rm df}$ values at the main minimum are due to the fact that in the process of their calculation only the statistical uncertainties of the reference data were used, and therefore the errors were underestimated.

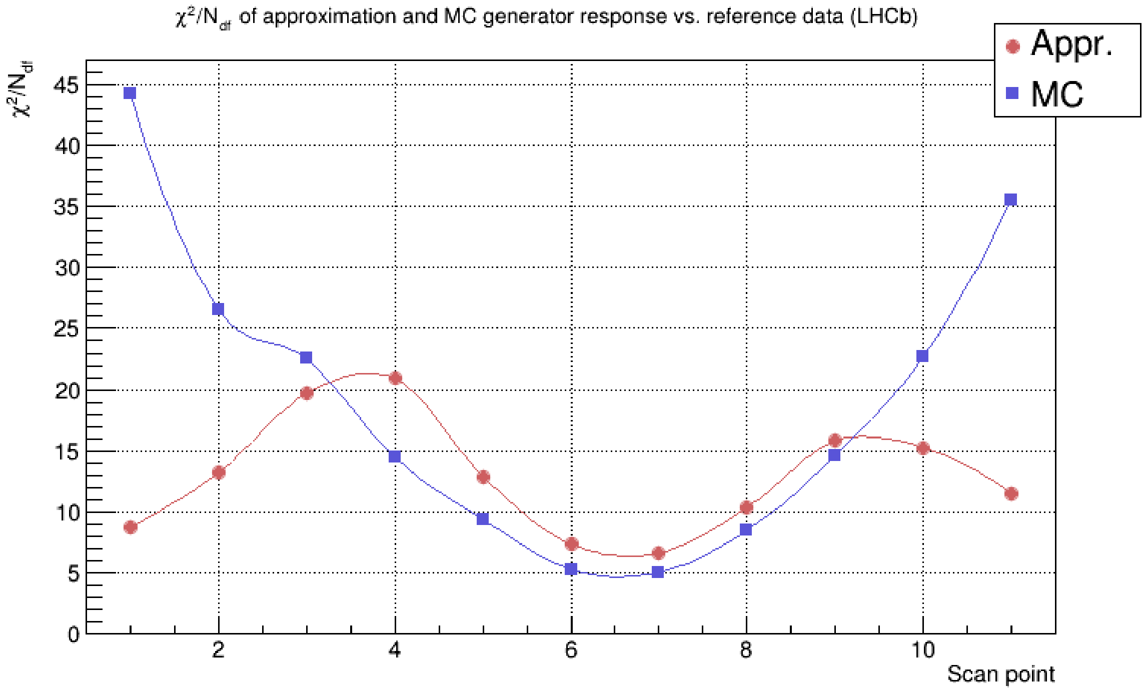


Figure 8.28: $\chi^2/N_{\rm df}$ plot of simultaneous tuning of four parameters `PARJ(21)`, `PARJ(42)`, `PARJ(41)` and `PARJ(82)`. Red points correspond to the approximation, blue points to the Monte Carlo generator response. The χ^2 values were calculated taking into account statistical uncertainties, which were estimated the way it was described in Section 6.3 of Chapter 6.

Figure 8.28 presents the results of the same scan test, with account for the statistical uncertainties of the approximation. The decline of the $\chi^2/N_{\rm df}$ values for the parameterization, plotted in red, towards the boundaries is due to the growth of the statistical errors of the approximation. If in the central region of the minimum the statistical uncertainties of the approximation are comparable to those of the Monte Carlo generator response, then past point 4 towards the left boundary and past point 9 towards the right limit the difference starts to grow. This behavior is explained by the comparably small number of generated events per data sample and the growth of the errors of the parameterization towards the boundaries of the scanned volume.

The difference between the $\chi^2/N_{\rm df}$ values in the region of the minimum is explained by the difference in statistical uncertainties: the statistical errors of the approximation are smaller due to the higher statistics. The blue curve corresponds to the MC data set with 10^4 events, the red one is the approximation based on a fit to $2 \times N_{\rm min}$ number of anchor points, corresponding to $2 \times N_{\rm min} \times 10^4$ MC events.

The statistical uncertainties of the approximation make the χ^2 a badly-behaved function. Minimization without defining parameter boundaries yields unacceptable results with `PARJ(82)` ending up having negative values. In the case of an explicit definition of parameter limits, `Minuit` prefers values outside of the designated boundaries for `PARJ(41)` and `PARJ(82)`, therefore it was decided to ignore Monte Carlo statistical errors in the parameterization.

Minimization without accounting for the statistical uncertainties of the approximation results in a minimum of $\chi^2/N_{\mathrm{df}} = 6.49$ for the parameter values presented in Tab. 8.3. It is, however, a visible improvement over the Monte Carlo with default parameter values, which has $\chi^2/N_{\mathrm{df}} = 12.14$.

Parameter	default value	new value	description
`PARJ(21)`	0.36	0.427 ± 0.003	σ
`PARJ(41)`	0.3	0.229 ± 0.016	a
`PARJ(42)`	0.58	0.402 ± 0.014	b
`PARJ(82)`	1	1.735 ± 0.053	shower cut-off

Table 8.3: Default and new values of the tuned parameters, estimated by minimizing the approximation χ^2 function without consideration of statistical uncertainties of the parameterization. The symmetrical uncertainties for the values of the minimized parameters are the parabolic errors determined from the covariance matrix by `Minuit`.

The validation scan around the minimized parameter values, presented in Fig. 8.29, shows a good agreement between the approximation and the Monte Carlo generator. The limits for the parameters values for the scan were set to ±0.4 of the corresponding best value and are presented in Tab. 8.4. The high χ^2 value in the minimum is explained by the fact that the statistical errors of the parameterization were ignored.

Parameter	new value	lower limit	upper limit	description
`PARJ(21)`	0.427	0.36	0.516	σ
`PARJ(41)`	0.229	0.58	0.494	a
`PARJ(42)`	0.402	0.3	0.44	b
`PARJ(82)`	1.735	1.0	2.05	shower cut-off

Table 8.4: Parameter boundaries for the validation scan.

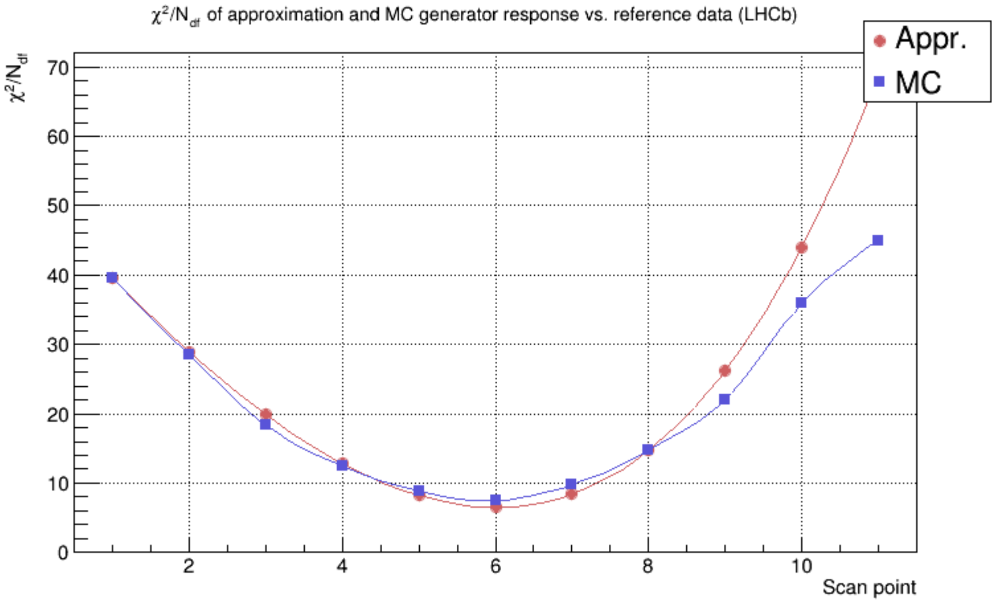


Figure 8.29: $\chi^2/N_{\rm df}$ plot for the best tuning validation test of parameters `PARJ(21)`, `PARJ(42)`, `PARJ(41)` and `PARJ(82)`. Red points correspond to the approximation, blue points to the Monte Carlo generator response. Point 6 corresponds to the new parameter values presented in Tab. 8.3. The scan was done within ±0.4 of the corresponding best value. The χ^2 values were calculated taking into account only statistical uncertainties of the reference data.

Figures 8.30 through 8.42 present the comparison of the criteria distributions with the new tune (red) plotted alongside the default Monte Carlo generator configuration (blue) and the reference data (black).

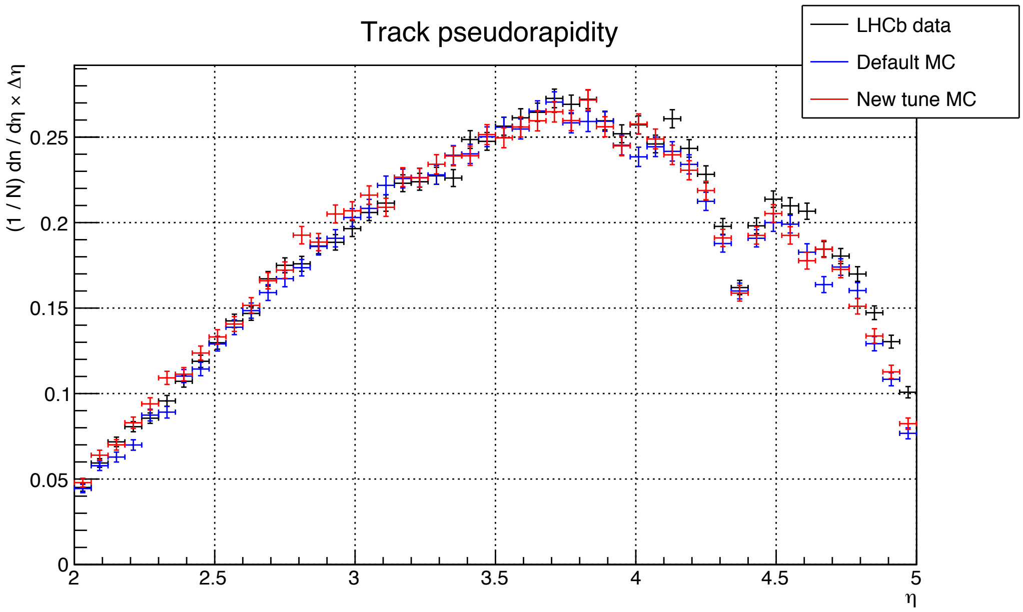


Figure 8.30: Track pseudorapidity distributions, between the reference data (LHCb recorded, plotted in black), default (blue) and new (red) Monte Carlo tunes.

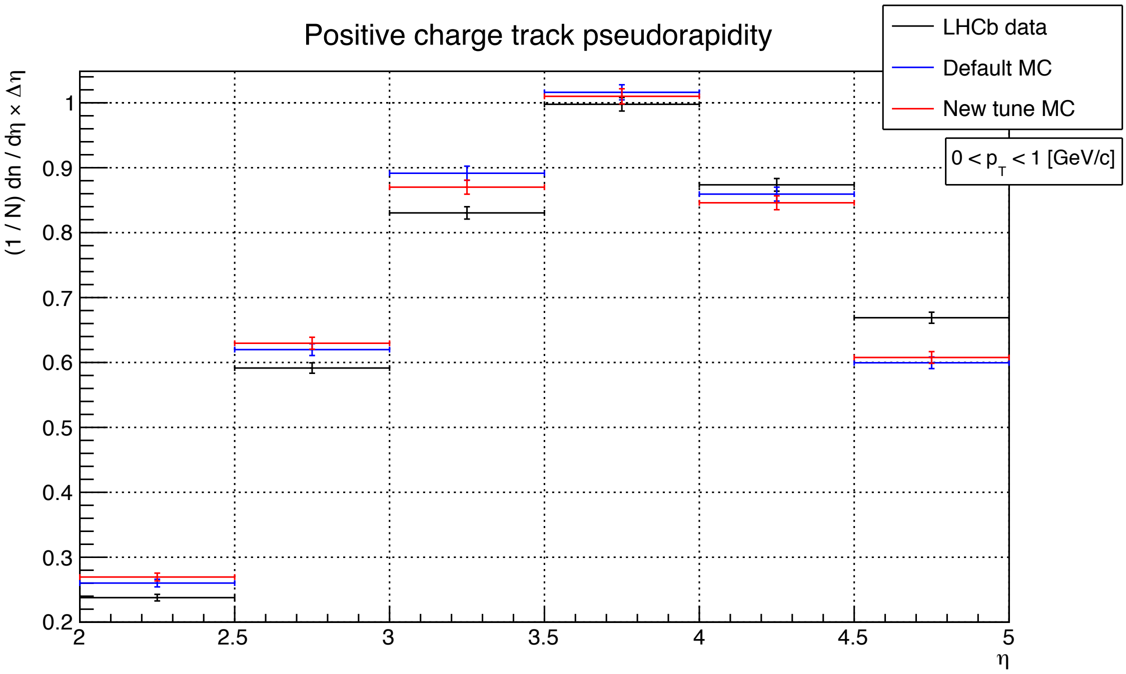


Figure 8.31: Pseudorapidity distribution for positively charged tracks with $0 < p_{\rm T} < 1$ GeV/c for $2 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

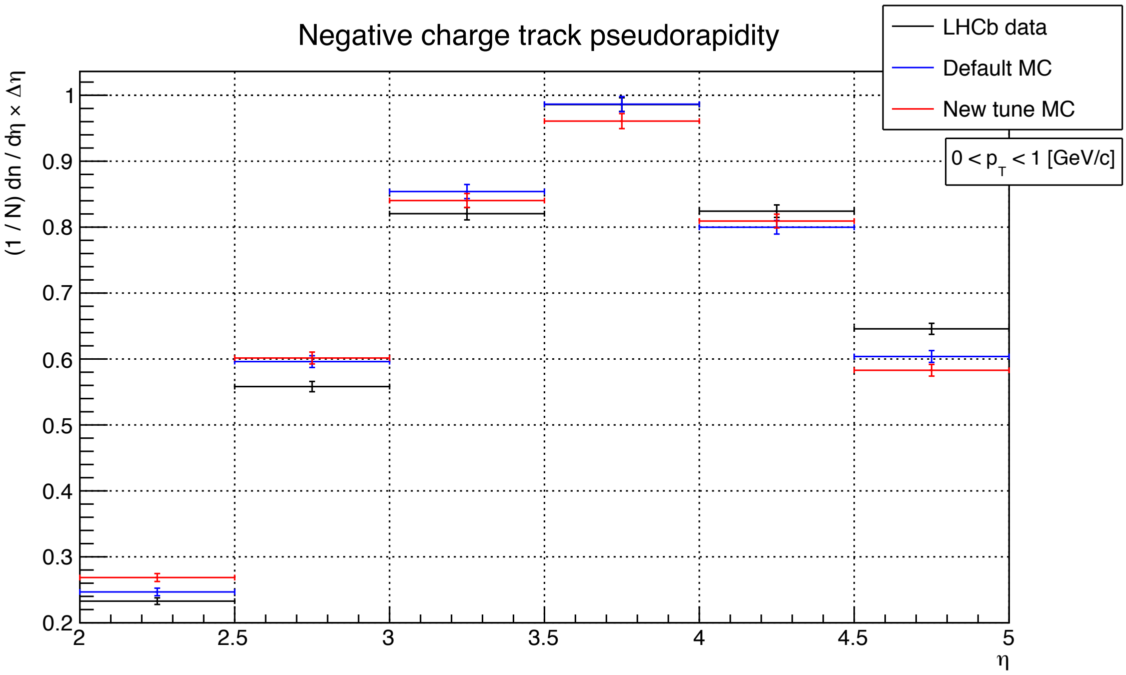


Figure 8.32: Pseudorapidity distribution for negatively charged tracks with $0 < p_{\rm T} < 1$ GeV/c for $2 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

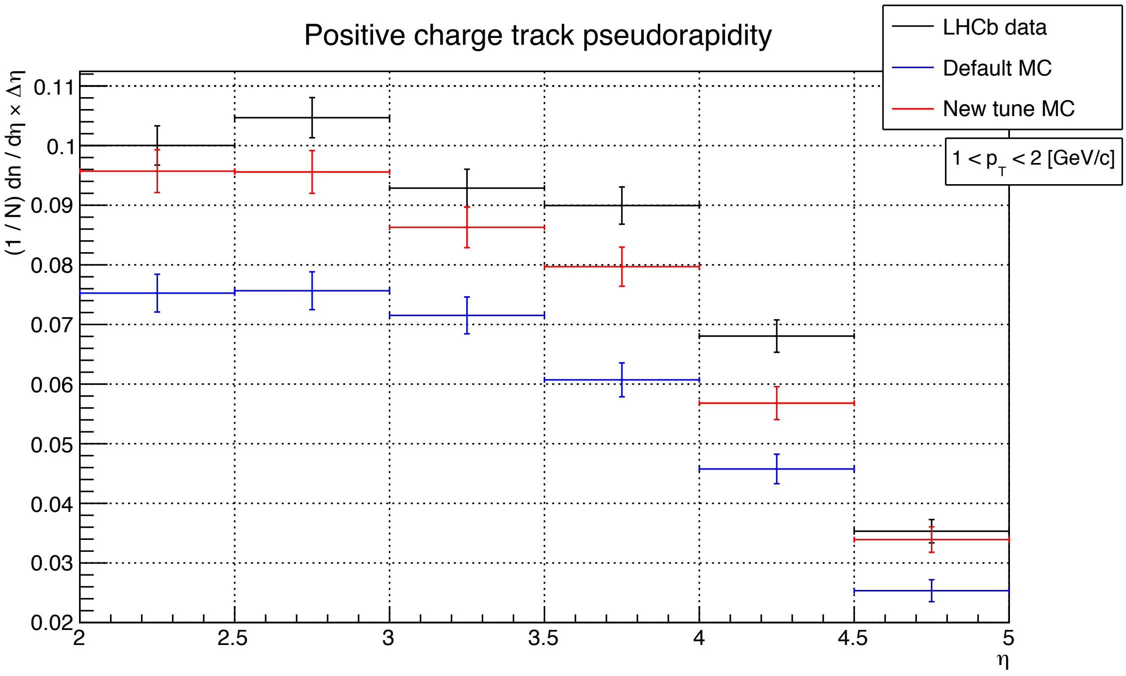


Figure 8.33: Pseudorapidity distribution for positively charged tracks with $1 < p_{\mathrm{T}} < 2$ GeV$/c$ for $2 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

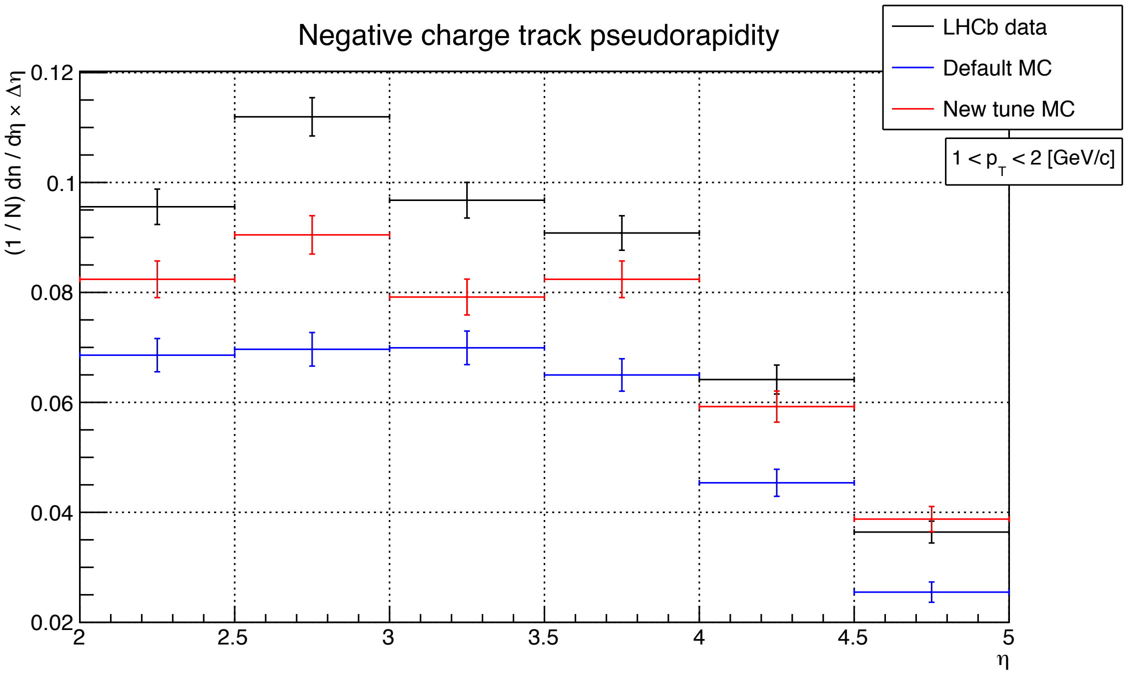


Figure 8.34: Pseudorapidity distribution for negatively charged tracks with $1 < p_{\mathrm{T}} < 2$ GeV$/c$ for $2 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

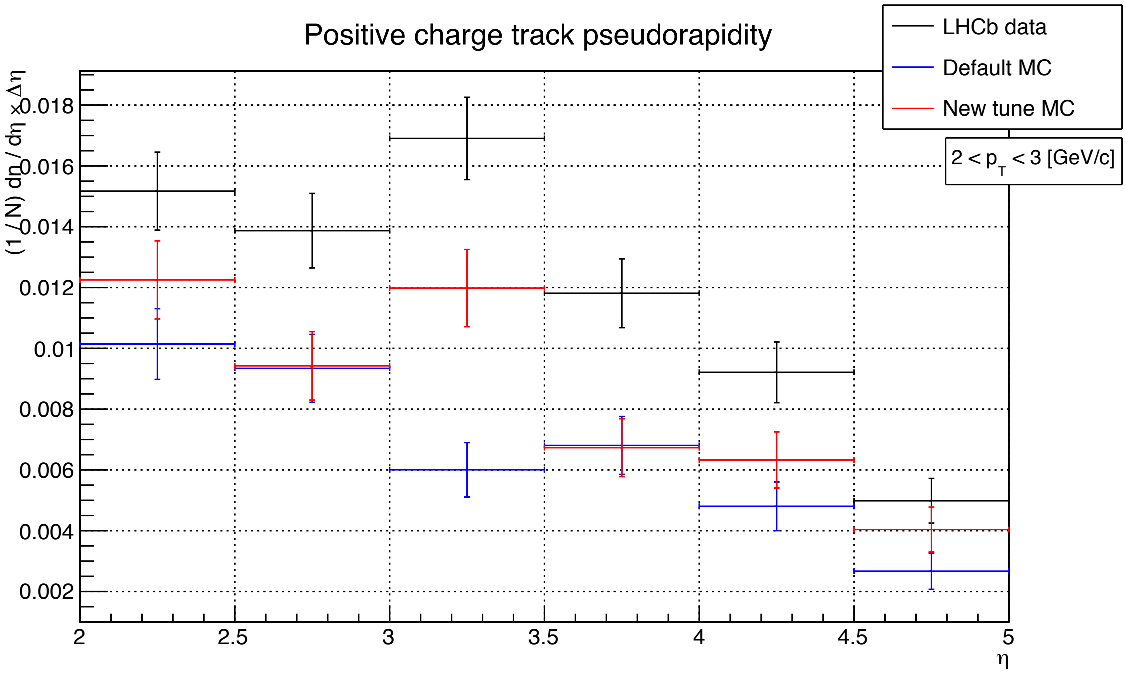


Figure 8.35: Pseudorapidity distribution for positively charged tracks with $2 < p_{\mathrm{T}} < 3$ GeV/c for $2 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

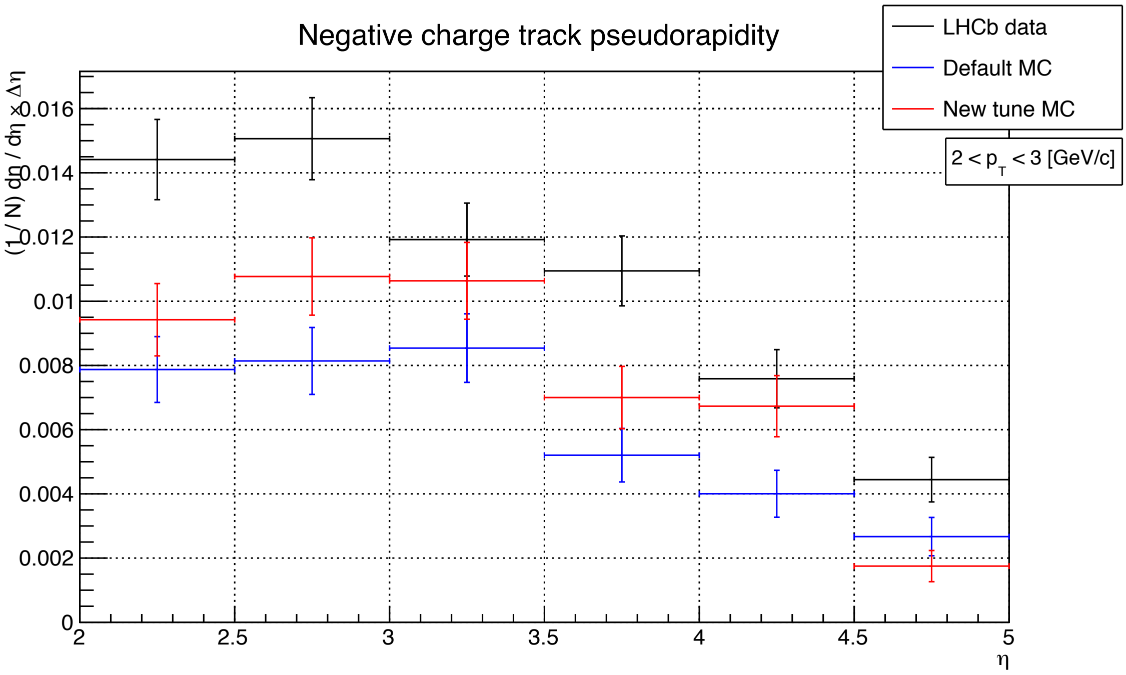


Figure 8.36: Pseudorapidity distribution for negatively charged tracks with $2 < p_{\mathrm{T}} < 3$ GeV/c for $2 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

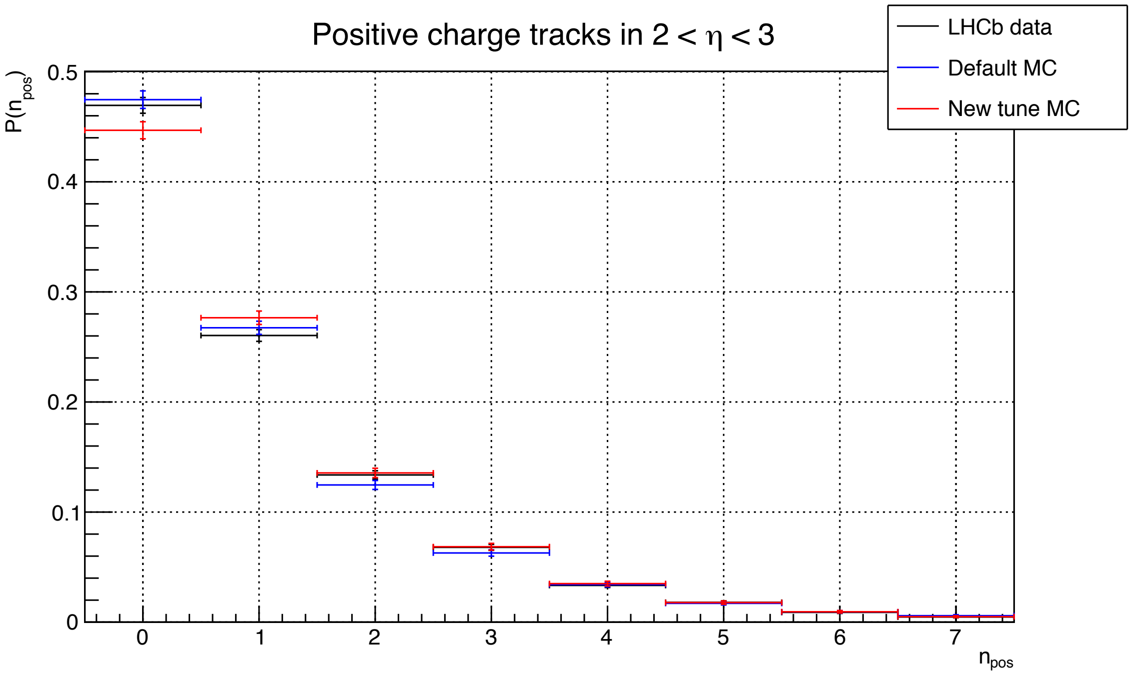


Figure 8.37: Multiplicities of positively charged tracks for $2 < \eta < 3$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

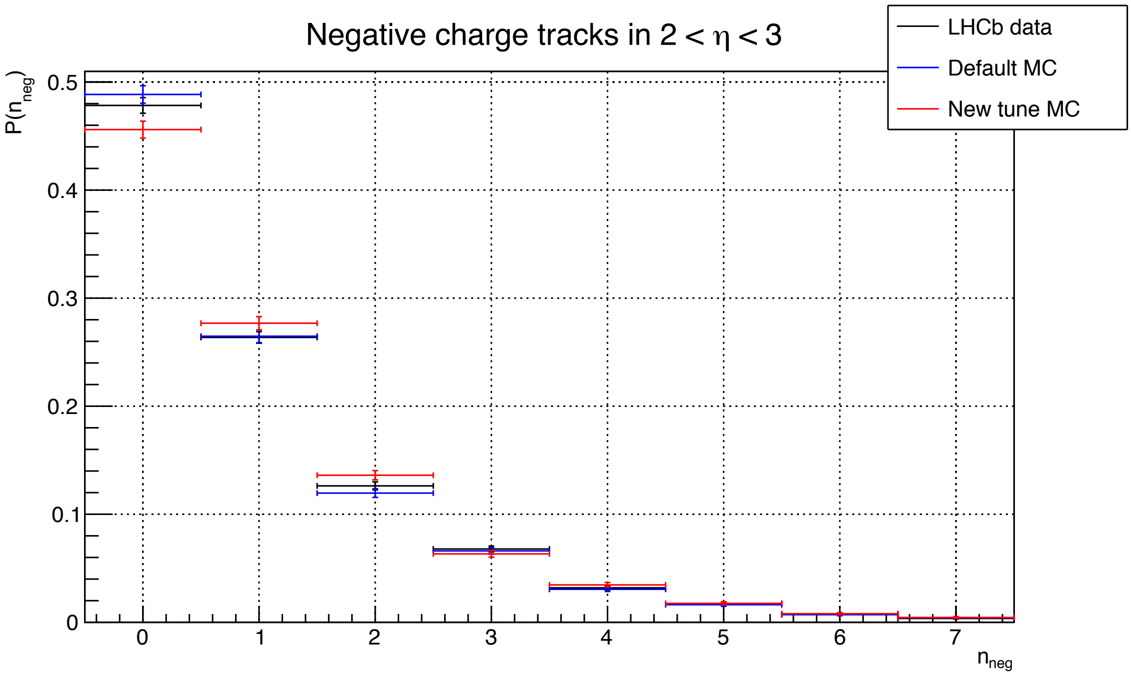


Figure 8.38: Multiplicities of negatively charged tracks for $2 < \eta < 3$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

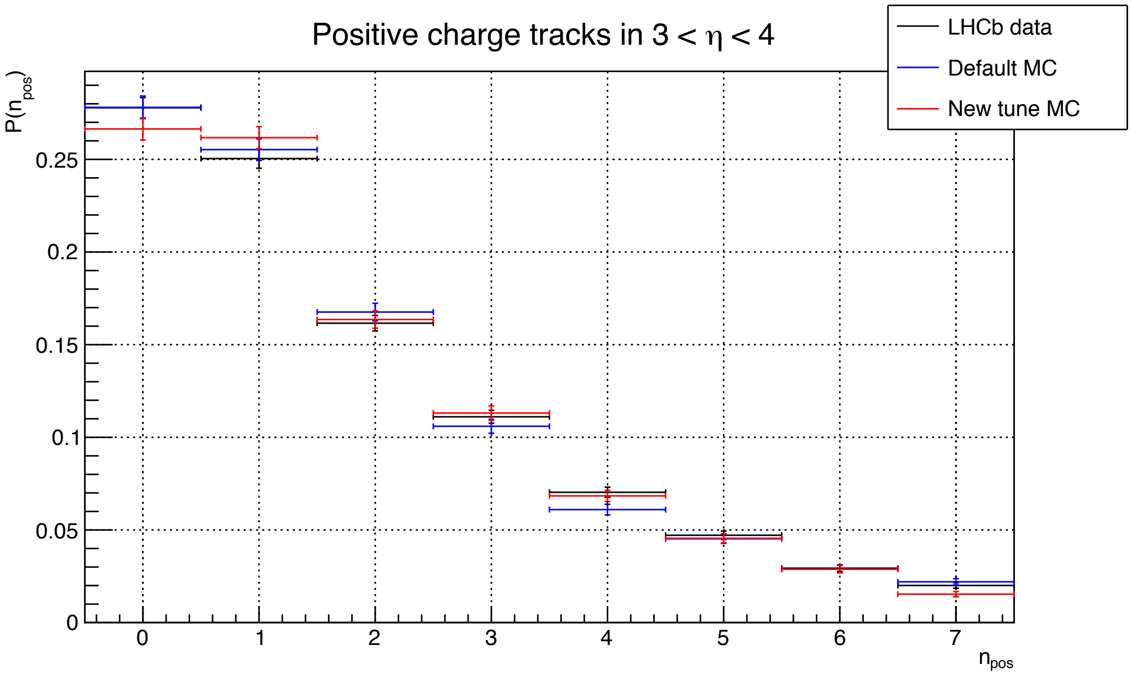


Figure 8.39: Multiplicities of positively charged tracks for $3 < \eta < 4$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

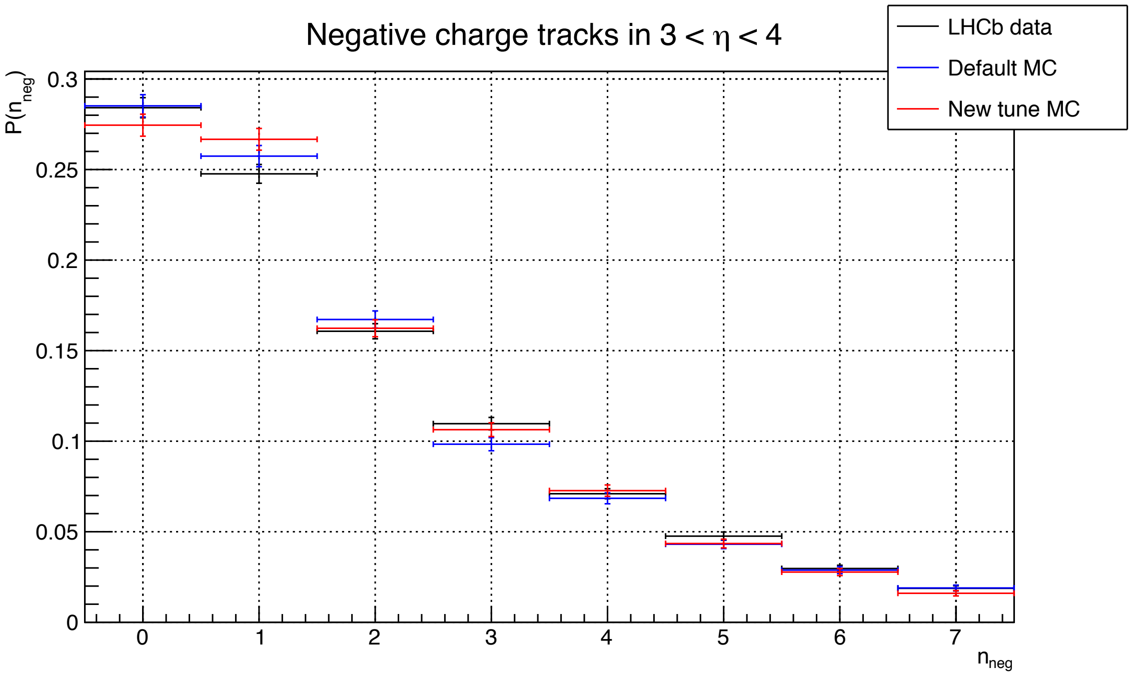


Figure 8.40: Multiplicities of negatively charged tracks for $3 < \eta < 4$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

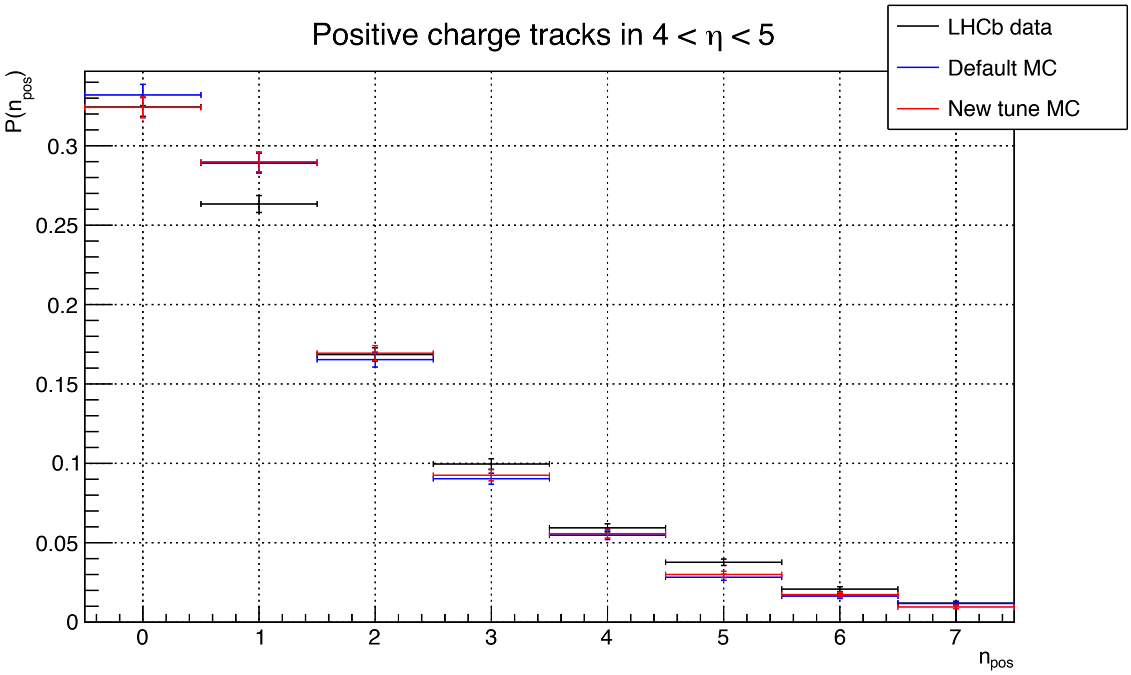


Figure 8.41: Multiplicities of positively charged tracks for $4 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

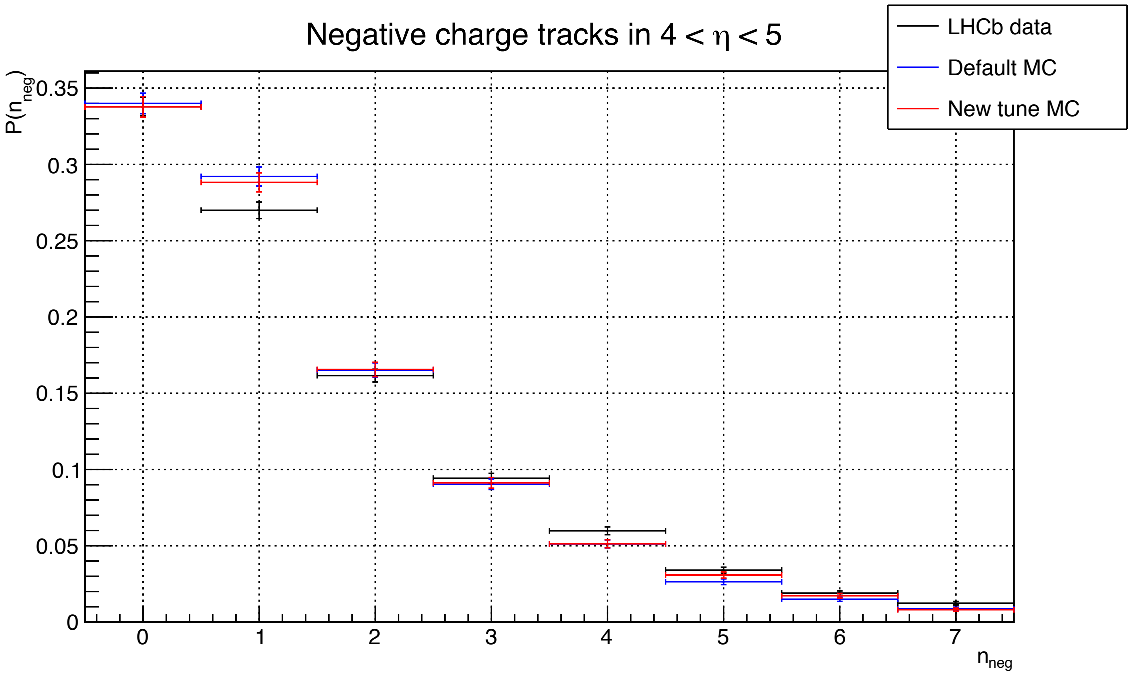


Figure 8.42: Multiplicities of negatively charged tracks for $4 < \eta < 5$. The reference data (LHCb recorded) is plotted in black. Default and new Monte Carlo tunes are plotted in blue and red respectively.

It appears that for the most of the figures, the default and new Monte Carlo tunes agree well within the statistical uncertainties and tend to describe the reference data equally good or with a similar level of discrepancy. However, the Monte Carlo data sample, simulated with the newly found parameter values, improves the reference data description in the case of the η distribution for tracks with $1 < p_{\mathrm{T}} < 3$ GeV/c. This effect is also seen in Fig. 8.43 (in the central part), which shows the comparison of the tuning histogram for the two Monte Carlo tunes and reference data.

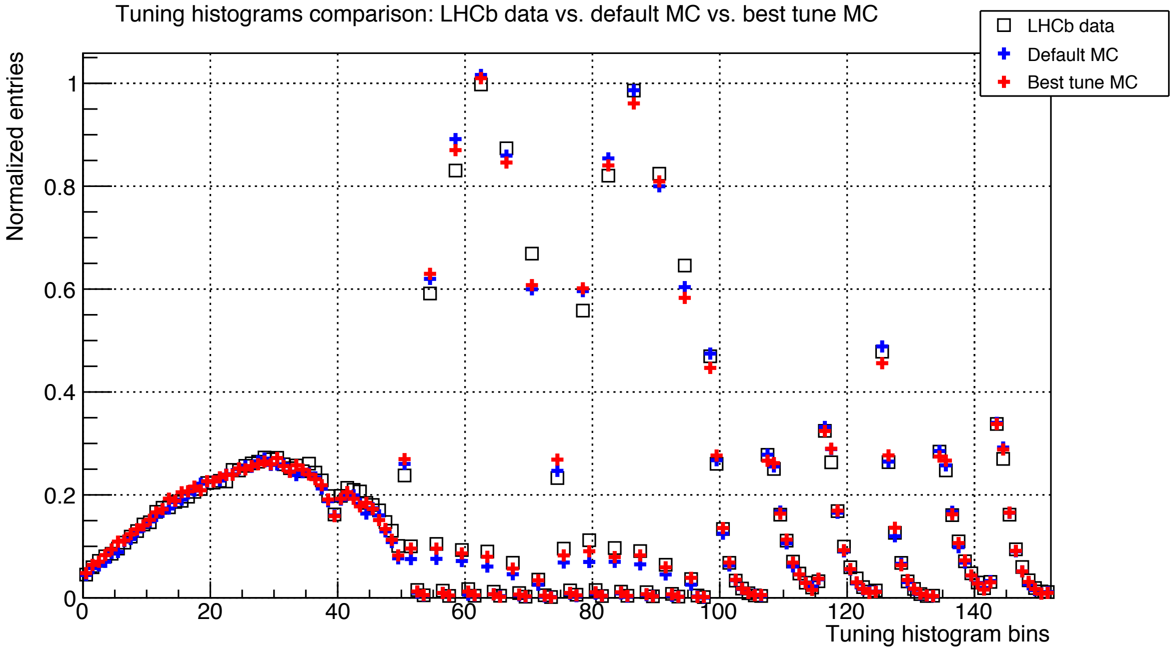


Figure 8.43: Comparison plot of criteria histograms for the reference data (LCHb recorded) and two Monte Carlo data samples (default and new tune), as they are represented in the tuning application.

Table 8.5 presents the χ^2 values of the comparison criteria for the default and the new Monte Carlo tunes versus the reference data. It is seen that the improvement of the χ^2 of the new Monte Carlo tune mainly comes from the η distributions of higher-p_{T} tracks.

Distribution	bins	default MC χ^2	overflow bin χ^2	new MC χ^2	overflow bin χ^2
η	50	291.24	–	270.89	–
η of n_{pos} $(0 < p_{\mathrm{T}} < 1$ GeV/c)	5	145.34	–	140.61	–
η of n_{neg} $(0 < p_{\mathrm{T}} < 1$ GeV/c)	5	75.93	–	151.11	–
η of n_{pos} $(1 < p_{\mathrm{T}} < 2$ GeV/c)	5	357.05	–	41.71	–
η of n_{neg} $(1 < p_{\mathrm{T}} < 2$ GeV/c)	5	435.24	–	96.47	–
η of n_{pos} $(2 < p_{\mathrm{T}} < 3$ GeV/c)	5	142.86	–	61.72	–
η of n_{neg} $(2 < p_{\mathrm{T}} < 3$ GeV/c)	5	115.52	–	57.56	–
η of n_{pos} $(p_{\mathrm{T}} > 3$ GeV/c)	5	3.39	–	17.63	–
η of n_{neg} $(p_{\mathrm{T}} > 3$ GeV/c)	5	26.19	–	3.35	–
n_{pos} multiplicity $(2 < \eta < 3)$	8	15.21	0.47	20.8	8.44
n_{neg} multiplicity $(2 < \eta < 3)$	8	7.57	2.03	30.35	0.95
n_{pos} multiplicity $(3 < \eta < 4)$	8	19.13	3.94	20.87	7.08
n_{neg} multiplicity $(3 < \eta < 4)$	8	21.77	0.83	26.13	0.15
n_{pos} multiplicity $(4 < \eta < 5)$	8	66.9	2.73	55.82	4.92
n_{neg} multiplicity $(4 < \eta < 5)$	8	64.19	0.01	42.34	1.78

Table 8.5: The per-distribution χ^2 values for the default and new Monte Carlo tunes, compared to the reference data. The "bins" column states the numbers of the bins in the histograms of the corresponding distributions, which are the same as on the plots presented before. For the several distributions for which the tuning application also fits the overflow bin, its contribution is shown in the "overflow bin χ^2" column.

The errors on the new parameter values (best tune values) were estimated with the help of the likelihood-based method, called *profile likelihood*. The scan interval for each parameter p_i was defined by the two values $p_i^{\mathrm{upper}} = p_i^{best} + \Delta p_i$ and $p_i^{\mathrm{lower}} = p_i^{best} - \Delta p_i$ (upper and lower boundaries respectively), for which

$$\min_{p_j, j\neq i} \chi^2(\vec{p}, p_i = p_i^{\mathrm{upper}} \text{ or } p_i^{\mathrm{lower}}) = \chi^2_{\mathrm{min}} \begin{array}{l} +1 \ \rightarrow \ \text{nominal} \\ +6.49 \ \rightarrow \ \text{used here} \end{array} ,$$

and $\Delta p_i = \alpha \times p^{\mathrm{best}}_{i_{\mathrm{Minuit\ error}}}$, where $p^{\mathrm{best}}_{i_{\mathrm{Minuit\ error}}}$ is the error on the minimized parameter value estimated by `Minuit`, and α is a constant number which was arbitrary set to $\alpha = 2.75$. Since in the case of the study the errors were underestimated by ignoring statistical uncertainties of the approximation, the 68% confidence interval must be adjusted accordingly, by increasing the χ^2_{min} by $\Delta\chi^2$, which is expected to be $\Delta\chi^2 = \langle \frac{\chi^2}{N_{\mathrm{df}}} \rangle = 1$. From the underestimated errors it follows that the uncertainty of the fitted parameters is also underestimated. To account for this, $\Delta\chi^2$ was chosen as $\Delta\chi^2 = \chi^2_{\mathrm{min}}/N_{\mathrm{df}} = 6.49$.

Figures 8.44 to 8.47 present the results of the profiling procedure for the tuned parameters. In each point on the separate plot, the profiled parameter had its value fixed and the χ^2 function was minimized against the other three free parameters. The minima on the plots match the best tune values for the corresponding profiled parameters.

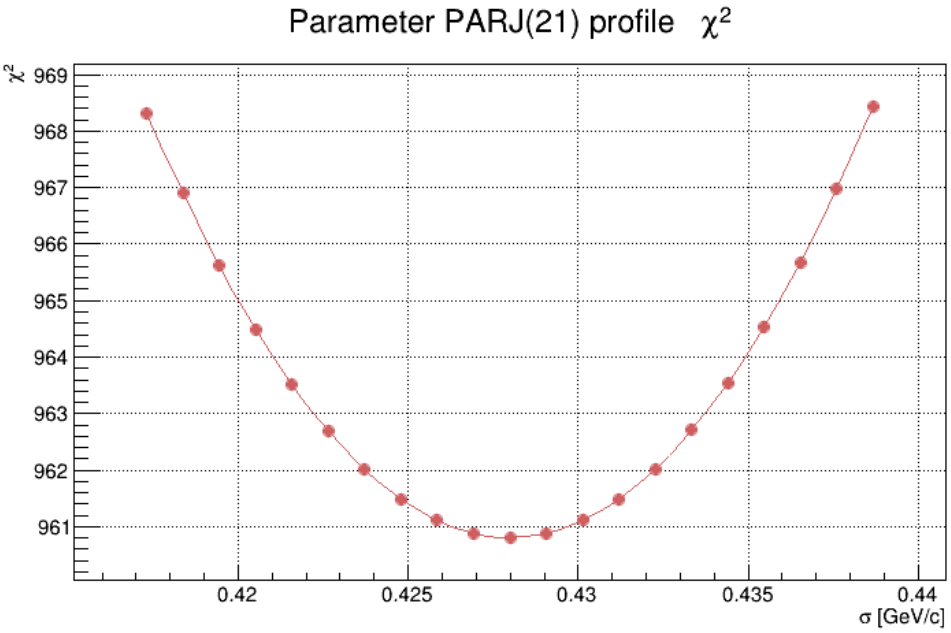


Figure 8.44: Profile likelihood χ^2 plot of parameter PARJ(21). The vertical axis corresponds to the χ^2 value, and the horizontal one to the parameter value.

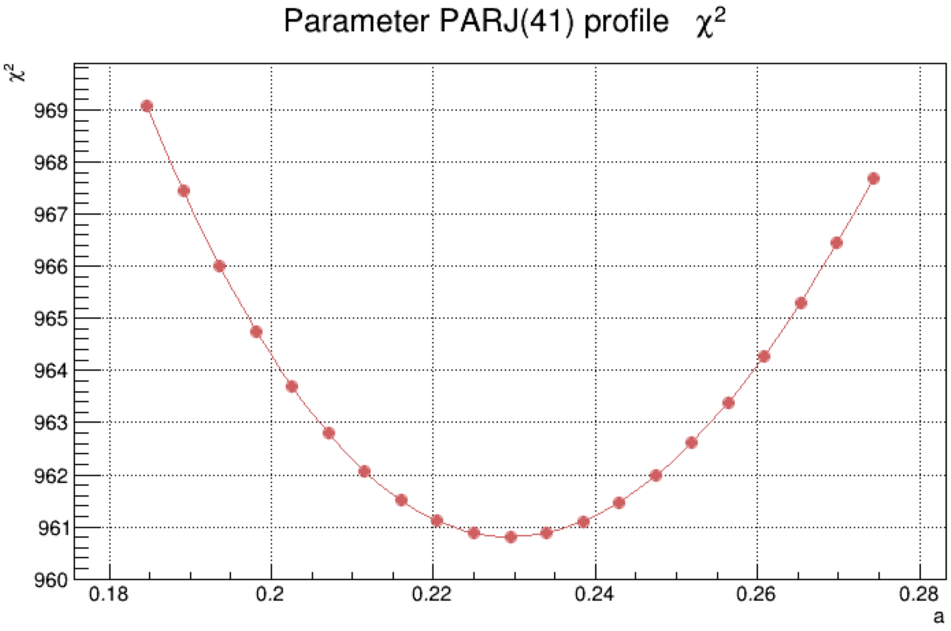


Figure 8.45: Profile likelihood χ^2 plot of parameter PARJ(41). The vertical axis corresponds to the χ^2 value, and the horizontal one to the parameter value.

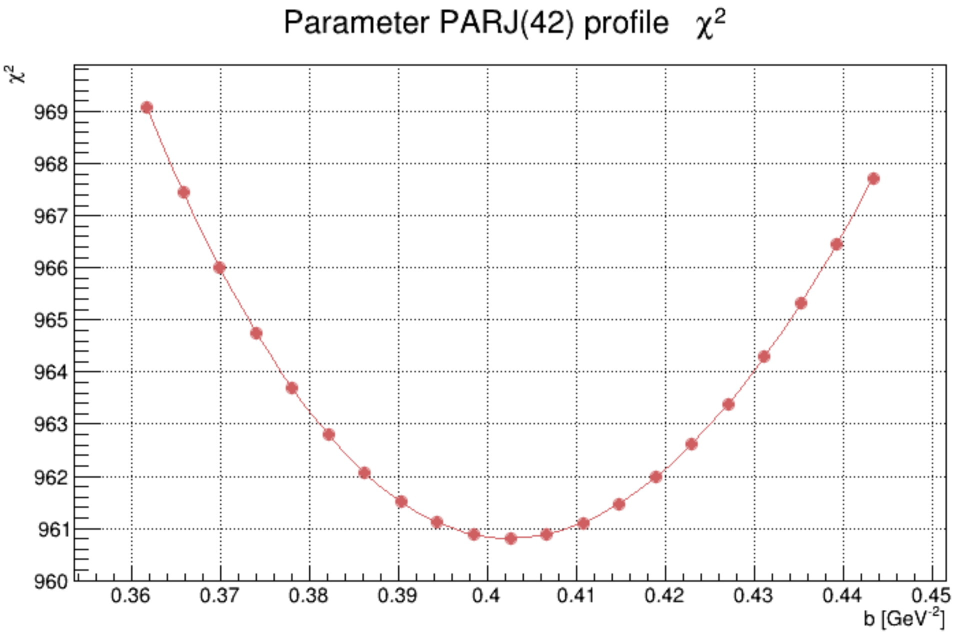


Figure 8.46: Profile likelihood χ^2 plot of parameter `PARJ(42)`. The vertical axis corresponds to the χ^2 value, and the horizontal one to the parameter value.

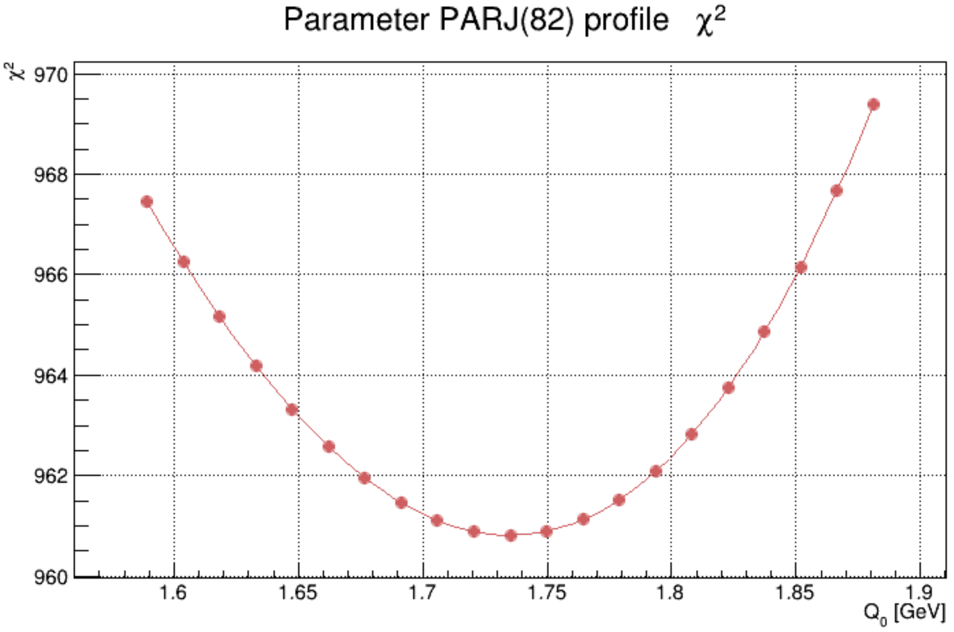


Figure 8.47: Profile likelihood χ^2 plot of parameter `PARJ(82)`. The vertical axis corresponds to the χ^2 value, and the horizontal one to the parameter value.

The resulting new values and their uncertainties, estimated from the profiling procedure, for the four tuned parameters are presented in Tab. 8.6.

Parameter	default value	new value	uncertainty	description
`PARJ(21)`	0.36	0.427	$+0.009$ -0.009	σ
`PARJ(41)`	0.3	0.229	$+0.043$ -0.039	a
`PARJ(42)`	0.58	0.402	$+0.039$ -0.036	b
`PARJ(82)`	1	1.735	$+0.127$ -0.144	shower cut-off

Table 8.6: The default values of the parameters, their new tuned values and their uncertainties.

Chapter 9

Conclusions

Manum de tabula!
Lat.: Hand off the board!

Simulations play a very important role in modern physics. Recently, numerical modeling has become a rapidly growing area of science and the Monte Carlo tuning technics, considered in this work, have entered a "boom era". Monte Carlo predictions are used in many fields and have a lot of applications: from estimating experimental efficiencies and uncertainties to testing theories. The Monte Carlo tuning helps to achieve a better agreement between simulated and reference data.

The LHCb software framework models physics processes in the LHCb detector, and makes it possible to predict results of the experiment. In PYTHIA, Monte Carlo event generator simulates processes in collisions of protons (hard processes, initial- and final-state radiation, parton showers, fragmentation, decays). The detector simulation (GEANT) describes particles, traversing through the detector material (ionization, multiple scattering, nuclear interactions). But as has been described in this work, many models which describe physics processes, contain weakly constrained free parameters. In order to obtain Monte Carlo event generator predictions which will agree with the reference experimental data, it is required to simultaneously tune many phenomenological parameters in a multidimensional parameter space.

This work focuses on the creation of a fast analytical model of the Monte Carlo event generator, on the development of mathematical methods and creation of their software application, followed by its tuning with respect to the data collected by the LHCb detector. Herewith, the fitting algorithm is integrated with the LHCb detector software model.

The following results were obtained in this work:

I The tuning procedure:

1. Further development of the methodology of parameterization of Monte Carlo generator behavior:
   a) expansion of the interpolation approach: interpolation by a 2$^{\mathrm{nd}}$ or 3$^{\mathrm{rd}}$ order polynomial,

   b) computation of the unknown polynomial coefficients by two different methods using the method of least-squares:
      - a fast method of solving an overdetermined system via the normal system of equations, which requires an estimate of computational errors;
      - a precise method of solving an overdetermined system of equations via singular value decomposition of the system matrix, which is more computationally heavy than the first method;

   c) use of per-bin weights (ability to emphasise or to exclude individual bins or distributions, used as comparison criteria),

   d) use of multiple distributions in comparison criteria,

   e) implementation in the object-oriented C++ environment (use of ROOT object-oriented data analysis framework), that is more flexible and extensible.

2. A detailed step by step fitting algorithm is presented, which estimates the parameterization coefficients and delivers the optimal, best set of parameter values by using numerical minimization of the χ^2 function.

3. A methodology of estimation of uncertainties of the parameterization result is implemented.

4. A tuning application package implemented in C++.

   The `JobManager` application, which is responsible for transparent parallelization of simulation jobs on a local computational cluster.

   The Analytical application, which implements a physics analysis and produces histograms of selected distributions, used as comparison criteria by the main tuning application.

   The main tuning application `Lagrange`, which implements the fitting algorithm.

   The source code of all the applications is well commented.

5. The developed methodology and applications were tested with simple numerical examples.

6. The `Lagrange` application implements a method of guaranteed estimation of computational errors, based on the developed methodology, and estimation of statistical uncertainties of the parameterization result.

7. The tuning application provides a way to test the obtained parameterization by scanning the parameters space along an arbitrary straight line.

8. The estimated best set of parameter values

II The developed application package is a tool for systematic tuning of Monte Carlo event generator parameters to the experimental data recorded by LHCb detector.

The following four major non-flavor hadronization parameters were selected for the studies: Lund string parameters a `PARJ(41)` and b `PARJ(42)`, Gaussian transverse momentum distributions σ `PARJ(21)` and the shower parameter (shower cut-off) `PARJ(82)`.

Chapter 8 describes in detail:

a) Influence of the number of the initial sampling points on the approximation of the Monte Carlo generator response;

b) Analysis of the sensitivity of the selected distributions to the shifts of the tuned parameters in the parameter space;

c) Analysis of the parameterization;

d) Numerical experiments, likelihood profiling procedure and the results of the main tuning of the four selected parameters, presented in Table 8.6.

The following conclusions are drawn from the numerical experiments:

- Parameterization of the Monte Carlo generator response gives a robust analytical model, which allows to make fast predictions of generator distributions;
- The fitting algorithm finds the best set of parameter values which make the Monte Carlo generator better describe the reference data, and such tuning is very important for any physics analysis which relies on Monte Carlo data;
- The developed tuning package:
  - is a tool which can handle such tuning tasks systematically, it is fully automatized and once a tuning has been configured - all the procedures such as parallel computational jobs execution, the main tuning, scan and validation tests, likelihood profiling, etc. are done in completely automatic manner;
  - uses the LHCb software framework to fully recreate the environment in which the reference data was recorded;
  - allows accounting not only for statistical, but also computational uncertainties;
  - makes it possible to produce tunings for different purposes (different user-defined comparison distributions);
- A full tuning of four selected parameters of Pythia 6.4 was done with the help of the developed software package, using a sample of data recorded by LHCb detector in 2010.

Therefore, the further development and adoption by the LHCb community of the tools, described in the thesis, can improve the description of the data recorded by the LCHb detector by the Monte Carlo event generators, and refine the precision of measurements which rely on them.

Bibliography

[1] CERN, CERN Document Server, http://cds.cern.ch/

[2] ATLAS Collaboration, W.W. Armstrong et al., *ATLAS: Technical Proposal for a General-Purpose pp Experiment at the Large Hadron Collider at CERN*, CERN-LHCC/94-43, 1994.

[3] CMS Collaboration, *CMS: Technical Proposal*, CERN-LHCC/94-38, 1994.

[4] ALICE Collaboration, N. Ahmad et al., *A Large Ion Collider Experiment – Technical Proposal*, CERN-LHCC/95-71, 1995.

[5] O. Adriani et al., *Technical Proposal for the CERN LHCf Experiment: Measurement of Photons and Neutral Pions in the Very Forward Region of LHC*, Technical Report, CERN-LHCC/2005-032, LHCC-P-007, 1999.

[6] W. Kienzle et al., *TOTEM, Total Cross Section, Elastic Scattering and Diffraction Dissosiation at the LHC.*
Technical Proposal, Technical Report, CERN-LHCC/99-007, LHCC-P-5, 1999.

[7] J. van Tilburg, *Track simulation and reconstruction in LHCb*, Thesis, NIKHEF, 2005.

[8] O. Schneider, T. Nakada. *LHCb trigger. Prepared for the 4th International Conference on B Physics and CP violation (BCP4)*, Ago Town, Mie Prefecture, Japan, 19-23 Feb. 2001.

[9] T. Sjostrand et al., *PYTHIA 6.4 Physics and manual*, JHEP 0605 (2006) 026.

[10] LHCb Collaboration, S. Amato et al., *LHCb Technical Proposal*, CERN-LHCC/98-4, 1998.

[11] LHCb Collaboration, *The LHCb Detector at the LHC*, JINST 3, 2008.

[12] *LHCb Magnet: Technical Design Report*, CERN-LHCC-2000-007.

[13] *LHCb VELO TDR: Vertex Locator. Technical Design Report*, CERN-LHCC-2001-011.

[14] Aras Papadelis, *Characterisation and commissioning of the LHCb VELO detector*, Thesis, 2009.

[15] *The LHCb Silicon Tracker, first operational results*, CERN-LHCb-CONF-2010-003.

[16] LHCb Collaboration, P. Barbosa-Marinho et al., *LHCb Inner Tracker Technical Design Report*, CERN-LHCC/2002-029, 2003.

[17] LHCb Collaboration, P. Barbosa-Marinho et al., *LHCb Outer Tracker Technical Design Report*, CERN-LHCC/2001-024, 2001.

[18] *LHCb RICH: Technical Design Report*, CERN-LHCC-2000-037.

[19] *LHCb Calorimeters: Technical Design Report*, CERN-LHCC-2000-036.

[20] LHCb Collaboration, P. Barbosa-Marinho et al., *LHCb Muon Technical Design Report*, CERN-LHCC/2001-010, 2001.

[21] *The LHCb trigger and its performance in 2011*, CERN-LHCb-DP-2012-004, 2012.

[22] Particle Data Group, *Review of Particle Physics*, Journal of Physics G, Nuclear and Particle Physics, Volume 37, Number 7A (2010).

[23] M. Kobayashi and T. Maskawa, Prog. Theor. Phys. 49 , 652 (1973).

[24] L. Wolfenstein, Phys. Rev. Lett. 51, 1945 (1983).

[25] I. Belayev, *Rare decays of B mesons at the LHCb*, Thesis, Moscow, 2005.

[26] A. J. Buras et al., Phys. Rev. D. 50, 3433 (1994).

[27] *The BABAR physics book: Physics at an asymmetric B factory*, (P. F. Harrison and H. R. Quinn, eds.), SLAC-R-0504, 1998.

[28] BABAR Collaboration, B. Aubert et al., *Measurement of the CP-violating asymmetry amplitude* $\sin 2\beta$, Phys. Rev. Lett. 89 (2002) 201802.

[29] BELLE Collaboration, K. Abe et al., *Measurement of the CP-violation parameter* $\sin(2\phi_1)$ *with 152 million B anti-B pairs*, hep-ex/0308036, 2003.

[30] K. Anikeev et al., *B physics at the Tevatron. RunII and beyond*, hep-ph/0201071.

[31] C. Jarlskog, Phys. Rev. Lett. 55, 1039 (1985).

[32] G. Dissetori, I.K. Knowles, M. Schmelling. *Quantum Chromodynamics: Hight Energy and Theory.* Oxford University Press, 2005.

[33] L.I. Saracheva. *Introduction into physics of microcosm. Physics of nucleus and particles.* Moscow: URSS, 2009.

[34] B.S. Ishhanov, I.M. Kapitonov, N.P.Udin. *Particles and atomic nucleus.* Moscow: URSS, 2007.

[35] Ta-Pei Cheng, Ling-Fong Li. *Gauge Theory of Elementary Particle Physics.* Clarendon Press, Oxford, 1984.

[36] A. Donnachie, P.V. Landshoff, *Total cross sections*, Phys. Lett., B296 (1992) 227-232.

[37] P. Nason et al., *Bottom production*, hep-ph/0003142, 1999.

[38] L. Smirnova, *Expected b-production at the LHC*, presentation at ICMD' 2009; access3.pdf.

[39] D. J. Lange, *The EvtGen Particle Decay Simulation Package*, Nucl. Instrum. Meth. A 462 (2001) 152.

[40] GEANT4 Collaboration, S. Agostinelli et al., *GEANT4: A simulation toolkit*, Nucl. Instrum. Meth. A 506 (2003) 250.
See also: http://geant4.web.cern.ch/geant4/

[41] *GEANT – Detector description and simulation tool*, CERN Program Library Long Write-up W5013, CERN Geneva.

[42] G. Barrand et al., *GAUDI - A software Architecture and Framework for Building HEP Data Processing Applications*, Comput. Phys. Commun. 140 (2001) 45
http://proj-gaudi.web.cern.ch/proj-gaudi/

[43] G. Corti, *Overview of Monte Carlo simulation(s) in LHCb*, LHCb-UK students meeting-CERN, 13 Oct 2009.

[44] I. Belayev et al., *Simulation Application for the LHCb Experiment*, 2003, [physics/0306035]
http://lncb-release-area.web.cern.ch/LHCb-release-area/DOC/gauss/

[45] Ryd A., et al., *EvtGen: A Monte Carlo Generator for B-Physics*, EVTGEN-V00-11-07(2005).
See also: http://www.slac.stanford.edu/lange/EvtGen/

[46] The LHCb collaboration, *The BOOLE project*,
http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/boole/

[47] The LHCb collaboration, *The MOORE project*,
http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/moore/

[48] The LHCb collaboration, *The BRUNEL project*,
http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/brunel/

[49] The LHCb collaboration, *The DAVINCI project*,
http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/davinci/

[50] G. Barrand. *Data Visualization*, 2003,
http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/Visualization/

[51] A. Buckley et al. *Systematic event generator tuning for the LHC.* (2009), arXiv: 0907.2973v1 [hep-ph].

[52] Andy Buckley, Hendrik Hoeth, Holger Schulz, Jan Eitee von Seggern. *Monte Carlo event generator validation and tuning for the LHC.* (2009), 0902.4403.

[53] DELPHI Collaboration, K. Hamacher et al., Z. Phys. C 73 (1996) 11.

[54] K. Hamacher and M. Weierstall, *The Next Round of Hadronic Generator Tuning Heavily Based on Identified Particle Data*, (1995), arXiv:hep-ex/9511011.

[55] A. Buckley. *Tools for event generator tuning and validation.* (2008),0809.4638.

[56] G. A. Korn, T. M. Korn. *Mathematical handbook for scientists and engineers. Definitions, theorems and formulas for reference and review* McGraw-Hill Company, New York Toronto London 1961.

[57] Jamse W. Demmel. *Applied Numerical Linear Algebra.* University of California, Berkley, California, 1997.

[58] R. Penrose. *On best approximate solution of linear matrix equations*, in Proceedings of the Cambridge Philosophical Society 52 (1956) 17-19.

[59] S. Brandt. *Data Analysis. Statistical and Computational Methods for Scientists and Engineers.* Springer-Verlag, New York, 1998.

[60] Derek J. Hudson. *Statistics. Lectures on Elementary Statistics and Probability.* Geneva, 1964.

[61] J. James and M.Roos. *Minuit: A System for Function Minimization and Analysis of the Parameter Errors and Correlations.* Comput. Phys. Commun. 10 (1975) 343-367.

[62] W.T. Eadie, D. Drijard, F.E. James, M. Roos, B. Sadoulet. *Statistical Methods in Experimental Physics.* Moscow: Atomizdat, 1976.

[63] G. Cowan. *Statistics.* Journal of Physics G. Nuclear and Particle Physics. Review of Particle Physics (PDG). 37, 7A (2010), 350-360.

[64] A. E. Albert. *Regression and the Moore-Penrose pseudoinverse.* Academic press, New York, 1972.

[65] S.K. Godunov, A.G. Antonov, O.P. Kiriluk, V.I. Kostin. *Guaranteed Accuracy of Solving Systems of Linear Equations in Euclidean Spaces.* Novosibirsk: Nauka, 1992.

[66] S.K. Godunov. *Lectures on Modern Aspects of Linear Algebra* . Novosibirsk: Nauchnay kniga, 2002.

[67] I.M. Sobol. *Distribution of points in a cube and approximate evaluation of integrals*, U.S.S.R Comput. Maths. Math. Phys. 7: 86–112, 1967.