Particle Acceleration and Detection

Rudolf Frühwirth
Are Strandlie

Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors

OPEN ACCESS

Springer

Particle Acceleration and Detection

The series “Particle Acceleration and Detection” is devoted to monograph texts dealing with all aspects of particle acceleration and detection research and advanced teaching. The scope also includes topics such as beam physics and instrumentation as well as applications. Presentations should strongly emphasize the underlying physical and engineering sciences. Of particular interest are

- contributions which relate fundamental research to new applications beyond the immediate realm of the original field of research
- contributions which connect fundamental research in the aforementioned fields to fundamental research in related physical or engineering sciences
- concise accounts of newly emerging important topics that are embedded in a broader framework in order to provide quick but readable access of very new material to a larger audience.

The books forming this collection will be of importance to graduate students and active researchers alike.

More information about this series at http://www.springer.com/series/5267

Rudolf Frühwirth • Are Strandlie

Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors

Springer

Rudolf Frühwirth
Institute of High Energy Physics
Austrian Academy of Sciences
Wien, Austria

Are Strandlie
Norwegian University of Science
and Technology
Gjøvik, Norway

Published with the support of the Austrian Science Fund (FWF): PUB 733-Z

ISSN 1611-1052 ISSN 2365-0877 (electronic)
Particle Acceleration and Detection
ISBN 978-3-030-65770-3 ISBN 978-3-030-65771-0 (eBook)
https://doi.org/10.1007/978-3-030-65771-0

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my teacher Meinhard Regler, my wife Sylvia Frühwirth-Schnatter, and my sons Stephan, Matthias, and Felix. — R.F.

To my wife Hanne Kari and my children Kristine, Harald, and Karen. — A.S.

Preface

Scope

Track and vertex reconstruction is an important part of the data analysis chain in experiments at particle accelerators. This includes fixed-target experiments, experiments at lepton and hadron colliders, and neutrino experiments where the detector is the target. This book deals almost exclusively with the reconstruction of charged particles and their production vertices in tracking detectors. The reconstruction of neutral particles in calorimeters and particle identification are subjects that are outside the scope of the present book; excellent treatments of these topics can be found elsewhere.

The methods presented here are also largely agnostic to the detector technology that produces the observations that are the input to the algorithmic chain of track finding $\rightarrow$ track fitting $\rightarrow$ vertex finding $\rightarrow$ vertex fitting. The problems that arise in producing hit positions with well-calibrated standard errors in the various types of tracking detectors are therefore described only very briefly; otherwise, it is simply assumed that such positions with correct standard errors are available. Nevertheless, an important part of the book deals with methods not sensitive to deviations from this ideal case.

Although track and vertex reconstruction can be formulated largely in geometrical terms, a statistically sound treatment of the stochastic processes that disturb the trajectories of charged particles by interactions with the detector material is mandatory. The modeling of these processes and their incorporation into the reconstruction algorithms is therefore a significant topic that is treated in considerable detail.

The examples in the last part of the book come from the four LHC experiments, complemented by two experiments at SuperKEKB and FAIR. Nonetheless, it was our aim to present the algorithmic solutions in as general a context as possible. Inevitably, some selection of the material in the extensive literature was necessary, driven by the available space and potentially somewhat influenced by our own experiences and predilections. Still, we aspired to describe in sufficient detail as many important contributions as possible; wherever this was not feasible, there are pointers to the relevant publications.

Content

The first part of the book is conceived as an introduction. Chapter 1 gives a very brief outline of tracking detectors, their basic principles of operation, and the challenges they pose for calibration and alignment. Chapter 2 shows how track and vertex reconstruction are embedded in the entire event reconstruction chain, from the trigger to the physics object reconstruction, and how important they are for the final physics analysis. Chapter 3 introduces basic notions from applied mathematics and statistics relevant for the core topics of the book: function minimization, regression and state space models, and clustering.

Part II covers the first core topic, track reconstruction. Chapter 4 describes track models, starting with the equations of motion of charged particles, followed by describing different ways of parametrizing the state of a particle and exhibiting algorithms for track and error propagation in various types of magnetic fields. It concludes with modeling of material effects and their inclusion in the track reconstruction. Chapter 5 is dedicated to track finding. As there is no systematic theory of track finding yet, we describe a variety of algorithms that have been and are presently successfully deployed in many experiments, including fast track finding in real time at the trigger level. Chapter 6 presents established methods for the estimation of track parameters, both traditional least-squares estimators and more recent robust and adaptive estimators. The special cases of circle and helix fitting are given a separate section as is the assessment of track quality. The detection of outliers and the finding of kinks in tracks concludes the chapter and Part II.

Part III is dedicated to the second core topic, vertex reconstruction. Chapter 7 first introduces the distinction between primary and secondary vertices, then goes on to discuss search strategies for finding primary vertices, both in one dimension and in 3D space. Various clustering algorithms are presented. Chapter 8 showcases methods for vertex fitting, which are very similar to the ones used for track fitting on a mathematical level. The concluding section shows how to extend the vertex fit to a kinematic fit by imposing additional geometric constraints and conservation laws. Part III concludes with Chap. 9, which deals with the reconstruction of secondary vertices. As the methodology of the search for secondary vertices is strongly influenced by the location of the vertex and the properties of the emerging particles, the four most important types are treated in four separate sections.

Part IV of the book presents case studies of approaches to tracking and vertexing from current and future experiments. Given our background in two of the LHC experiments and the enormous challenges in all of the LHC experiments, it is maybe not surprising that the four of them are our prime examples. They are complemented by two experiments not at the LHC, Belle II, and CBM. They have to solve their own specific tracking and vertexing problems, somewhat different from the ones typical for the LHC, but not less difficult all the same. We have to warn the reader, however, that at least some of these examples come with an expiration date as it were. In 2019, the LHC experiments have already started preparations for Run 3 of the LHC, which is scheduled to start in 2021, and for the high-luminosity phase of the LHC or HL-

LHC, planned for operation starting in 2026. It is to be expected that the conditions at the HL-LHC require substantial changes in the reconstruction algorithms of ATLAS and CMS, especially in the track finding part. Belle II has started operation recently and will certainly adapt and optimize its current algorithms with the rise of the luminosity of the SuperKEKB B factory. CBM was still in the preparation phase in 2019, and although it has found very convincing solutions to its tracking and vertexing challenges, it too will profit from experience and probably modify its approach if need arises. The reader should therefore keep in mind that the examples in Part IV are based on the published material at the time of writing, and that many exciting developments are yet to come or have already found their way into the track and vertex reconstruction software.

The appendices following Part IV contain supplementary material. Appendix A lists the Jacobian matrices of the parameter transformations treated in Chap. 4; Appendix B shows the regularization of the kinematic fit for singular covariance matrices; and Appendix C contains a list of software packages for track and vertex fitting, as well as entire frameworks with existing, but easily replaceable, modules for track and vertex reconstruction. These frameworks can serve as convenient testbeds for new ideas and algorithms, addressing users and developers alike.

Audience

The book is intended for a wide audience: PhD students who want to gain better understanding of the inner workings of track and vertex reconstruction; PhD students and postdocs who want to enrich their experience by participating in projects that require knowledge of the topic; and researchers of all ages who want to contribute to the progress of the field by becoming algorithm and software developers. In all cases, the reader is expected to have some basic knowledge of linear algebra and statistics.

Acknowledgements

We first wish to thank the series editor Werner Riegler (CERN), who has inspired and encouraged us to write this book and given us valuable feedback on all things related to detectors. We thank the anonymous reviewers for their valuable suggestions and corrections, as well as the colleagues from the experiments for their corrections and additions to Part IV: David Rohr for ALICE; Markus Elsing and Andreas Salzburger for ATLAS; Marco Rovere and Erica Brondolin for CMS; Agnes Dziurda for LHCb; Nils Braun and Felix Metzner for Belle II; and Ivan Kisel for CBM. Any errors that may remain are our own responsibility.

We owe a great deal of thanks to our colleagues and students with whom we have collaborated on track and vertex reconstruction in the course of our professional

lives, always teaching and learning at the same time. At CERN: Pierre Billoir, Teddy Todorov, Thomas Speer, Pascal Vanlaer, Wolfgang Adam, Matthias Winkler, Martin Liendl, Andreas Salzburger, Wolfgang Liebig, and Tom Atkinson; in Gjøvik and Oslo: Esben Lund, Lars Bugge, Jørn Wroldsen, Bjørn Lillekjendlie, Steinar Stapnes, and Håvard Gjersdal; and in Vienna: Meinhard Regler, Sylvia Frühwirth-Schnatter, Winfried Mitaroff, Peter Kubinec, Dieter Stampfer, Wolfgang Waltenberger, Josef Scherzer, Edmund Widl, Manfred Valentan, Moritz Nadler, Jakob Lettenbichler, and Erica Brondolin.

R.F. thanks his former director Jochen Schieck for allowing him to use the infrastructure at the Institute for High Energy Physics in Vienna for a year after retirement. He also thanks Eugenio Paoloni for his essential contributions to the development of the track finder for the Belle II SVD.

The open access version of this book has been made possible by a grant from the Austrian Science Fund (FWF). We thank the FWF for its generous support.

Vienna, Austria — Rudi Frühwirth
Gjøvik, Norway — Are Strandlie
May 2020

A Note on the References

WWW addresses (URLs) are given for web resources, technical reports, articles in arXiv and open access journals, and material that is hard to find otherwise.

Typesetting and Notation

Vectors are typeset in small bold italic letters, for example, $\boldsymbol{a}$. Unless specified otherwise, all vectors are column vectors. The length of vector $\boldsymbol{a}$ is denoted by $\dim(\boldsymbol{a})$, its transpose by $\boldsymbol{a}^{\mathsf{T}}$, its norm by $|\boldsymbol{a}|$. The scalar product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted by $\boldsymbol{a} \cdot \boldsymbol{b}$, and their cross product by $\boldsymbol{a} \times \boldsymbol{b}$. Matrices are typeset in capital bold italic letters, for example, $\boldsymbol{A}$. The rank of matrix $\boldsymbol{A}$ is denoted by $\mathrm{rank}(\boldsymbol{A})$, its diagonal by $\mathrm{diag}(\boldsymbol{A})$, its inverse by $\boldsymbol{A}^{-1}$, and its transpose by $\boldsymbol{A}^{\mathsf{T}}$. The block-diagonal matrix with blocks $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$ is denoted by $\boldsymbol{A} = \mathrm{blkdiag}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n)$. The gradient of a multivariate function $F(\boldsymbol{x})$ is a row vector and denoted by ∇F; the Hessian matrix is denoted by $\nabla^2 F$. The expectation of a random variable z is denoted by $\mathsf{E}[z]$ and its variance by $\mathsf{var}[z]$. The expectation of a random vector $\boldsymbol{\varepsilon}$ is denoted by $\mathsf{E}[\boldsymbol{\varepsilon}]$ and its covariance matrix by $\mathsf{Var}[\boldsymbol{\varepsilon}]$. The cross-covariance matrix of two random vectors $\boldsymbol{\varepsilon}$ and $\boldsymbol{\delta}$ is denoted by $\mathsf{Cov}[\boldsymbol{\varepsilon}, \boldsymbol{\delta}]$. As usual, δ_{ij} is the Kronecker delta.

Contents

List of Figures

List of Tables

Part I
Introduction

Chapter 1
Tracking Detectors

Abstract The chapter gives an overview of particle detectors, with the emphasis on tracking detectors. The working principles and the calibration of gaseous, semiconductor, and fiber detectors are explained, followed by a brief review of detector alignment. As an illustration, the tracking systems of the four experiments at the LHC and two non-LHC experiments, Belle II and CBM, are presented.

1.1 Introduction

Many high-energy physics experiments are performed by colliding two beams of high energy particles or one beam with a target. The particles produced in a collision are recorded by particle detectors. The collision is then studied by reconstructing most of the particles produced in the interaction and determining their properties. For a general review of particle detectors, see for instance [1, Chap. 7] and [2].

The present book concentrates on the reconstruction of charged tracks and interaction vertices using information collected by tracking detectors. A tracking detector can be a single device such as a wire chamber or a silicon strip sensor, or a full-blown tracking system such as a time projection chamber capable of stand-alone track reconstruction; see Sect. 1.6. For a review of non-tracking detectors, the reader is referred to [1, 3].

A charged particle crossing a tracking detector generates a single or a string of spatial observations in the local coordinate system of the detector. For track reconstruction, these have to be transformed to points in 3D space usually with different precisions in the three coordinates. Accordingly, the correct transformations are determined by the alignment procedure; see Sect. 1.5. The resulting space points or "hits" are collected into track candidates by the track finding. In the subsequent track fit, the track parameters are estimated and the track hypothesis is tested. Successfully reconstructed tracks are then clustered into production or decay vertices, followed by a vertex fit and a test of the vertex hypothesis.

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_1

There are three principal types of tracking detectors: gas-filled or gaseous tracking detectors; solid state tracking detectors, usually equipped with silicon sensors; and scintillating fiber trackers. They are described in the following three sections.

1.2 Gaseous Tracking Detectors

Gaseous tracking detectors utilize the ionizing effect of charged particles in a volume of gas. The simplest gaseous detector is the Geiger-Müller counter, a tube with a central wire. A potential difference of 1–3 kV between the tube wall and the wire causes the primary electrons and ions to move towards the anode (wire) and the cathode (tube wall), respectively. Because of the large field strength close to the wire, the primary electrons generate an avalanche of secondary electrons and ions, resulting in a detectable signal fed into an amplifier.

In order to use this principle for the measurement of the position of a charged particle, a structured array of many elements has to be designed. The following subsections describe a few common types of such arrays.

1.2.1 Multi-wire Proportional Chamber

The multi-wire proportional chamber (MWPC, [1, Sect. 7.1]) is a thin cuboid volume of gas, oriented approximately perpendicular to the passage of the particles to be measured. The volume is bounded by a pair of conductive plates acting as the cathode. Inside the volume an array of anode wires detects the passage of charged particles, see Fig. 1.1. With a typical wire spacing of 1 mm a spatial resolution of around 0.3 mm can be achieved in the v-coordinate orthogonal to the wires. A further improvement can be obtained by tilting the MWPC so that the probability of a particle giving signals on two adjacent wires gets larger or by rotating the chamber and measuring the drift times to the anode wires and estimating the point of passage from the drift distances [4, 5]. The resolution in the w-coordinate along the wire is poor, equal to the wire length divided by $\sqrt{12} \approx 3.46$.

1.2.2 Planar Drift Chamber

A planar drift chamber is similar in shape to an MWPC (see Fig. 1.1), but the electrical field is shaped by an alternating array of sense (anode) and field (cathode) wires; see [1, Sect. 7.2] and [6]. The electrons and ions from the primary ionization drift to the respective electrodes, and gas amplification in the strong field close to the sense wires gives a detectable signal that is amplified. In addition, the drift time

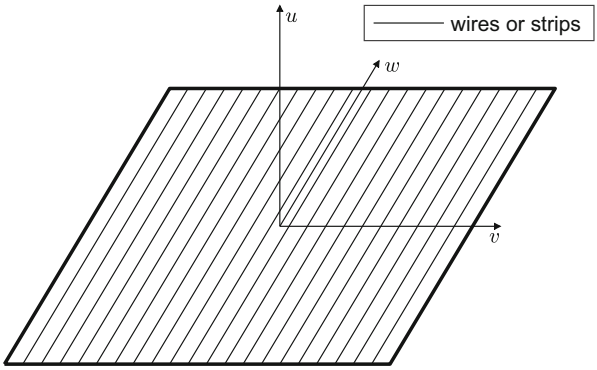

Fig. 1.1 Local coordinate system in a wire chamber, in a planar drift chamber, or in a silicon strip sensor

between the crossing time of the particle and the signal time on the sense wire is measured. The drift distance can then be computed from the drift time. Given a precise calibration of the drift distance, a spatial resolution below 0.1 mm in the coordinate v orthogonal to the wire can be achieved; see for instance [7].

Drift chambers need far fewer electronic channels than MWPCs, but have to be monitored and calibrated much more carefully. In addition, there is an inherent left-right ambiguity of the spatial position relative to the sense wire, so that each hit has a mirror hit. This ambiguity must be resolved in the track finding or track fitting stage. The spatial position w along the wire can be measured by comparing the charges at both ends of the wire. A resolution of a couple of millimeters can be achieved in this way [8].

1.2.3 Cylindrical Drift Chamber

Cylindrical drift chambers [1, Sects. 7.2 and 7.3] have been and still are widely used in collider experiments. Such a chamber consists of up to 60 cylindrical layers of alternating field and sense wires mostly parallel to the beams of the collider; see Fig. 1.2. In the local coordinate system of the chamber, the z-axis is the symmetry axis of the chamber. The measured point in the transverse plane can be given in polar coordinates $(R_\mathrm{m},\Phi_\mathrm{m},z_\mathrm{m})$, where R_m is the radial position of the sense wire, and Φ_m is the polar angle of the wire plus/minus the drift angle Φ_d, i.e., the drift distance d divided by R_m. The resolution of the drift distance is typically in the order of 0.1–0.2 mm; see for instance [9, 10]. Like in a planar drift chamber, each hit has a mirror hit, and the ambiguity must be resolved in the track finding or track fitting stage. The z-coordinate can be measured by charge division or by adding "stereo" layers of wires tilted with respect to the "axial" layers. The resulting

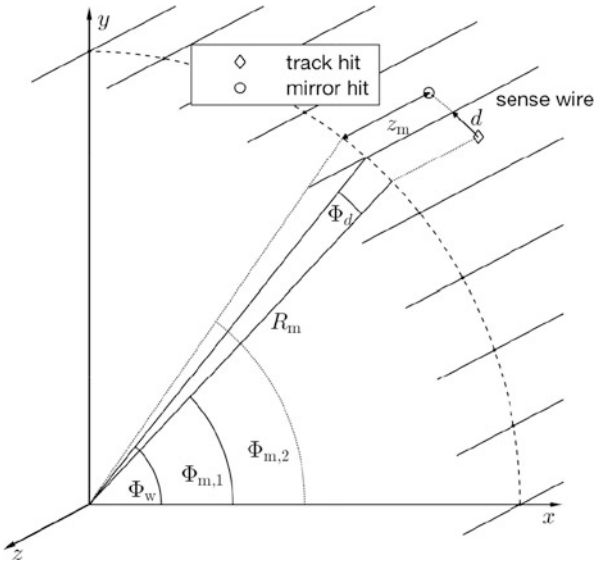

Fig. 1.2 Local coordinate system in a cylindrical drift chamber. Only one layer of sense wires is shown. R_m is the radial distance of the sense wire from the z axis; Φ_w is the azimuth angle of the sense wire. d is the drift distance; $\Phi_d = d/R_m$ is the angle spanned by d at radius R_m. The azimuth angles of the track hit and its mirror hit are $\Phi_{m,1} = \Phi_w - \Phi_d$, $\Phi_{m,2} = \Phi_w + \Phi_d$. The z-coordinate may or may not be measured directly

spatial resolution of z_m is equal to the drift distance resolution divided by the sine of the stereo angle, typically 2–3 mm.

1.2.4 Drift Tubes

Drift tubes are small drift chambers with a single sense wire. They can have rectangular or circular cross sections, and can be arranged in cylindrical or planar layers. In planar layers, wires can run in two or more directions, giving good spatial resolution in two orthogonal directions. The resolution of the drift distance is in the order of 0.1–0.2 mm.

1.2.5 Time Projection Chamber

The typical time projection chamber (TPC), as employed in many collider experiments, is a large gas-filled volume shaped as a hollow cylinder, the axis of which is aligned with the beams and the magnetic field [1, Sects. 7.3.3], see Fig. 1.3. There is a potential difference between the central cathode plane and the two anode end plates. The latter are equipped with position sensors. A charged particle traversing the chamber ionizes the gas, and the electrons travel along the field lines towards the end plates, where both the point and the time of arrival are measured and recorded. A track therefore generates a dense string of space points. The position

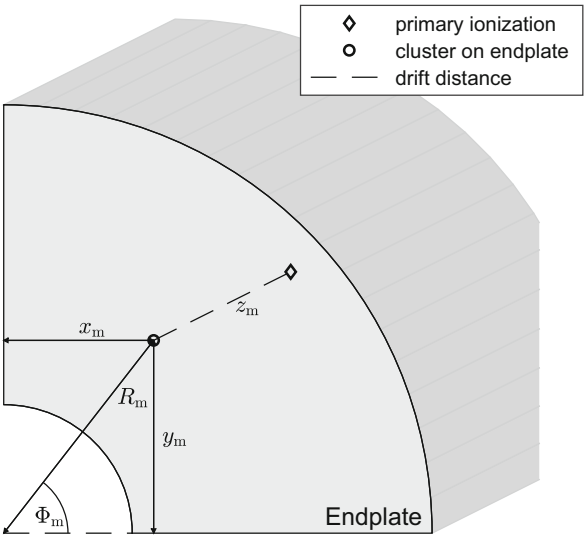

Fig. 1.3 Local coordinate system in a time projection chamber. The position in the endplate can be given in polar coordinates (R_m, Φ_m) or in Cartesian coordinates (x_m, y_m). z_m is the drift distance

sensors at the end plates can be wire chambers or micro-pattern gas detectors. In the drift direction z a spatial resolution in the order of 1 mm is possible; the transverse resolution depends on the technology of the endplate sensors and increases with the drift distance because of diffusion. Resolutions well below 0.1 mm can be achieved with GEM chambers on the end plates. Another important calibration issue is the correction of distortions arising from space charge effects, see for instance [11].

1.2.6 Micro-pattern Gas Detectors

The earliest micro-pattern gas detector is the micro-strip gas detector, in which wires are replaced by microscopic metal strip structures deposited on high-resistivity substrates [1, Sect. 7.4]. With typical strip distances of 75 µm, a spatial resolution below 20 µm can be achieved. Other developments are the Gas Electron Multiplier (GEM) and Micro-Mesh Gaseous Structure (Micromegas) chambers. Resolutions down to 10 µm can be attained by these devices. Due to their small size and fast collection of positive ions, they can be operated at high rates up to several MHz per mm^2.

1.3 Semiconductor Tracking Detectors

Semiconductor tracking detectors [1, Sect. 7.5] are mostly made of thin silicon wafers, approximately 0.3 mm thick. The n-type silicon is processed by photo-

lithographic methods to create p^+-doped implants on one side, either thin strips (see Sect. 1.3.1) or small pixels (see Sect. 1.3.2). Each strip or pixel is connected to a read-out channel. The silicon bulk is fully depleted by a bias voltage. A charged particle crossing the wafer creates electron-hole pairs along its path. The electrons and the holes drift towards the electrodes and induce signals on the read-out electrodes. Silicon drift detectors employ the measurement of the electron drift time for measuring the position of a crossing particle; see Sect. 1.3.3.

1.3.1 Silicon Strip Sensors

Large-area semiconductor tracking systems employ silicon strip sensors to keep construction costs affordable. The implants in the silicon wafers are narrow strips with a typical width of 20 µm and a typical inter-strip distance of 100 µm. The local coordinate system is shown in Fig. 1.1. The spatial resolution in v, orthogonal to the strip direction, depends on the track direction via the cluster size, i.e., the number of adjacent strips with a signal above threshold, and on the Lorentz angle [12]. Under optimal conditions the resolution is in the order of 10 µm or better. The resolution in w, parallel to the strip direction, is equal to the strip length divided by $\sqrt{12}$. A typical strip length is 5 cm; shorter strips with a length below a centimeter are called mini-strips. For an example of the calibration procedure, see [13].

Two-dimensional (2D) measurements can be achieved by implanting strips on the back side of the wafer, either orthogonal to the ones on the front side, or at a different stereo angle. Alternatively, two one-sided sensors can be glued on the same mechanical support, separated by a small gap. If such a sensor happens to be crossed by n particles at the same time, up to n strips on each side can give a signal, resulting in up to n^2 points of intersection. At most n of these correspond to the true particle positions; the remaining ones are spurious or ghost hits.

1.3.2 Hybrid Pixel Sensors

In a pixel sensor, the implants are small square or rectangular pixels in a high-resistivity silicon wafer. The pixels are connected to the read-out channel by bump bonding. Depending on the pixel size and the track direction, pixel sensors can have a position resolution below 10 µm in both coordinates, especially if the signal height is measured and used to interpolate between pixels in a cluster. For an interesting example of the calibration procedure, see [14]. Here, the position is estimated from precomputed cluster templates, considering the incident angle and the Lorentz angle. The templates can also be used to decide whether an observed cluster is compatible with a predicted incident angle.

Pixel sensors are mostly employed in the region close to the interaction point, as they can deal with the high track density and the high background radiation

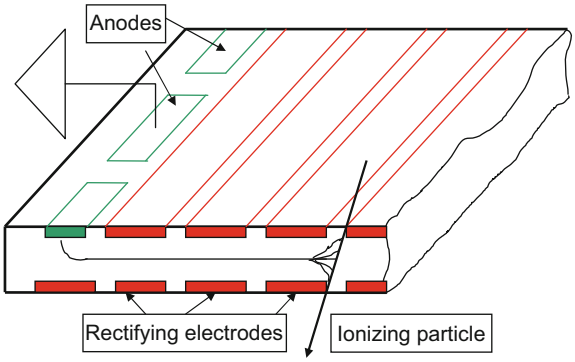

Fig. 1.4 Three-dimensional sketch of a multi-anode silicon drift detector. The trajectories of several signal electrons are indicated. The distance between the point where the ionizing particle crosses the middle plane of the detector and the array of anodes is obtained from measurement of electrons drift time. The second coordinate is given by the location of the anode pad where the signal electrons are collected. An improvement of resolution along this coordinate comes from charge sharing among several anodes. (From [16], by permission of Elsevier)

better than strip detectors. In addition, their excellent spatial resolution allows the separation of secondary decay vertices very close to the primary interaction vertex; see Chap. 9.

In monolithic pixel sensors, the sensitive volume and part of or the full read-out circuits are combined in one piece of silicon. The generated charge is collected on a dedicated collection electrode so that there is no need for delicate and expensive bump bonding. For an application of monolithic pixel sensors in a vertex detector, see Sect. 1.6.1.1.

1.3.3 Silicon Drift Sensors

In a silicon drift sensor, electrons are transported parallel to the surfaces of the sensor to an anode segmented into small pads [15, 16]. The position information along the drift direction is obtained from a measurement of the drift time of the electrons. The position in the second coordinate is obtained from charge sharing between adjacent pads; see Fig. 1.4. Silicon drift sensors have been deployed in STAR and ALICE, see [16, 17].

1.4 Scintillating Fiber Trackers

Scintillating fiber trackers combine the speed and efficiency of plastic scintillators with the geometric flexibility and hermeticity provided by fiber technology [18].

The fibers in such a tracking detector serve two functions: they convert the ionisation energy deposited by a passing charged particle into optical photons, and guide the optical signal to the devices that detect the generated light. In recent applications, these devices are silicon photomultipliers, which are fast, compact and sensitive to single photons [19].

The spatial resolution of a fiber is approximately equal to the diameter divided by $\sqrt{12}$, while the number of photons scales linearly with the diameter. The conflicting requirements on accuracy and light yield can be alleviated by staggering the fibers and by choosing a material with high intrinsic scintillation yield and long attenuation length. For an example of a large-scale scintillating fiber tracker designed for operation in the LHCb experiment from Run 3 of the LHC onward, see [20].

1.5 Alignment

The position measurements in the tracking detectors mentioned above are generated in the local coordinate system of the devices. In order to be useful for track reconstruction, they have to be converted to positions in the global coordinate system of the experiment along with the associated covariance matrices. As tracking detectors are very precise instruments, with position resolutions ranging from a couple of hundred micrometers down to about ten micrometers, their positions, orientations, and possible deformations have to be known with a similar or better precision. The importance of correct alignment, especially in the complex detectors of the LHC era, is attested by a series of workshops held at CERN in the past [21–23].

Misalignment or insufficient alignment has a deleterious effect on the efficiency of track and vertex reconstruction [21, p. 105]. Random misalignment also degrades the resolution of track and vertex parameters and subsequently of invariant masses. Moreover, systematic misalignment of larger substructures can cause a bias in the estimates of track momenta and vertex positions. This can be harmful in many of the physics analyses of the experiment.

Misalignment can have several sources: finite precision of the detector assembly, thermal and magnetic stresses on mechanical structures, sagging of wires or sensors because of gravity, changes in temperature and humidity, etc. Since misalignment can and does vary over time, constant monitoring is a necessity.

Alignment proceeds through several steps. The starting point is the ideal geometry, augmented by knowledge of the machining and assembly precision. The next step is alignment by hardware using lasers for measuring distances or proximity and tilt sensors. For instance, the ATLAS silicon tracker can be monitored optically by Frequency Scanning Interferometry to a precision of about 10 µm [24].

The final step is track-based alignment, either with tracks from cosmic muons, or from collisions, or both. Actually alignment profits from different types of tracks that cross different parts of the detector under different angles. For instance, tracks

from collisions hardly ever cross the entire central tracker of a collider experiment, but cosmic tracks do.

Track-based alignment can be split into internal alignment and external alignment. Internal alignment refers to the relative alignment of a tracking system, whereas external alignment refers to the alignment of the various tracking systems to the global frame of the experiment, which is usually tied to the beam pipe or some other part of the accelerator infrastructure. Even the internal alignment of a tracking system can be a big challenge. For instance, the current silicon tracker of the CMS detector has more than 10^4 sensors to be aligned, each with six degrees of freedom, not counting deformations of the sensors under gravity or thermal stresses. The estimation of $O(10^5)$ parameters is a highly non-trivial problem. A solution that has become a de-facto standard is the experiment-independent program MillepedeII [25, 26], which performs a simultaneous fit of (global) alignment parameters and (local) track parameters, allowing to include laser and survey data as well as equality constraints in the fit. For one of the alternative algorithms developed for track-based alignment, see [27].

1.6 Tracking Systems

Track reconstruction requires a minimal number of space points per track. A tracking system is a device that has enough information for stand-alone track reconstruction. A typical collider experiment has three tracking systems for momentum measurement: the vertex detector, the central tracker, and the muon tracking system. Fixed-target experiments frequently have vertex detectors as well, complemented by a magnetic spectrometer for momentum measurement.

The vertex detector is the tracking system closest to the beam, with the purpose to give very precise position and direction information of the tracks produced in a collision, so that decays very close to the interaction point can be detected with large efficiency; see Chap. 9. It therefore has the largest precision (smallest measurement errors) of all tracking systems. Vertex detectors are usually equipped with pixel sensors in order to achieve the required precision.

The central or inner tracker of a collider experiment is positioned between the vertex detector and the calorimeters. It is normally embedded in a solenoidal magnetic field with high bending power. A silicon tracker typically produces $O(10)$ hits per track, while a TPC produces $O(100)$ hits per track. The main requirements, not always easily satisfied, on the central tracker are: high single-hit precision; good capability to resolve two nearby tracks; precise momentum estimation by a long lever arm (large diameter); enough redundancy for high-quality track finding; hermetic coverage; as little material as possible. In some cases, especially at the future high-luminosity LHC (HL-LHC), fast readout is also essential, as trackers must be able to contribute to the trigger.

The muon system is situated behind the calorimeters which, in principle, absorb all particles with the exception of muons. Additional iron filters can be employed

as well. The muon system can provide an independent measurement of the muon momentum, especially for the purpose of triggering; see Sect. 2.1. If the muon system has to cover a large area, as is the case in the LHC experiments, it is typically equipped with proportional chambers, drift chambers or drift tubes.

The following subsections briefly describe the tracking systems of experiments at the LHC, the SuperKEKB B-factory, and the future Facility for Antiproton and Ion Research (FAIR).

1.6.1 Detectors at the LHC

1.6.1.1 ALICE

ALICE is a dedicated heavy-ion experiment at the LHC [28]. Its detector is designed to study the physics of strongly interacting matter at extreme energy densities.

Until the end of 2018, the Inner Tracking System (ITS, see Fig. 1.5) of the ALICE detector consisted of two barrel pixel layers [29], two layers of silicon drift detectors [17], and two layers of double-sided silicon strip detectors. After the upgrade in 2019–2020, the ITS consists of seven layers equipped with monolithic pixel chips [30].

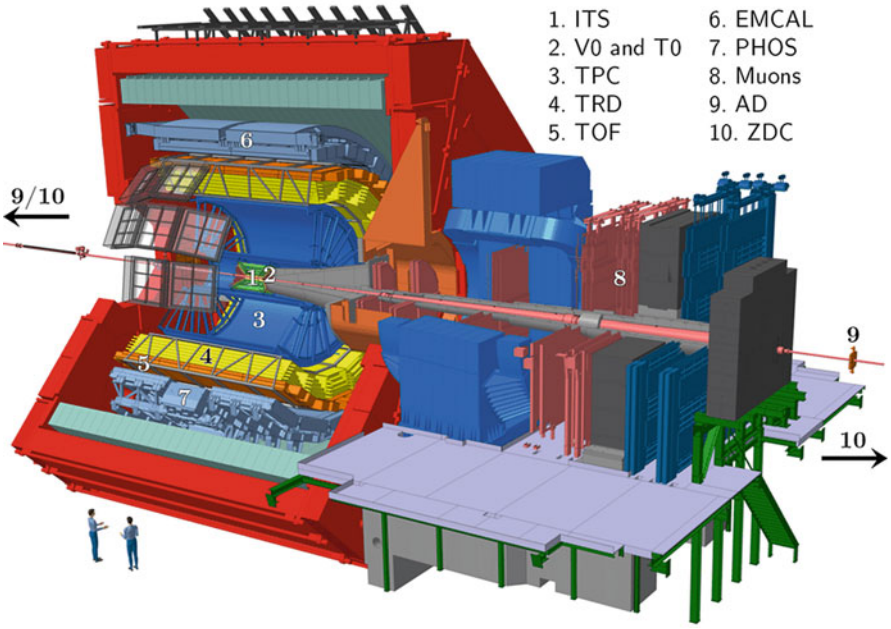

Fig. 1.5 Cut-away view of the ALICE detector. (From https://arxiv.org/abs/1812.08036. ©2015 CERN for the benefit of the ALICE Collaboration. Reproduced under License CC-BY-4.0)

The main tracking device of ALICE is a TPC [31, 32]. It provides up to 159 space points per track. The measurement of the energy deposit due to ionization provides a powerful tool for particle identification, especially for low-momentum particles, see [33]. For track reconstruction in the TPC and global track reconstruction, see Sect. 10.1. The TPC is surrounded by a transition radiation detector (TRD) used for triggering and electron identification.

The ALICE muon spectrometer covers only the forward region of the experiment and is dedicated to the study of quarkonia production, open heavy flavor production and vector meson properties via the muonic decay channel [34].

1.6.1.2 ATLAS

ATLAS [35] is one of the two general-purpose experiments at the LHC, the other one being CMS. Its vertex detector (see Fig. 1.6) originally consisted of three barrel pixel layers and three end-cap pixel disks on either side [36]. In 2014, a fourth pixel layer was inserted in the barrel between the existing pixel detectors and a new beam pipe with smaller radius [37].

The central tracker of ATLAS [36] consists of two parts: the silicon tracker (SCT) with four barrel layers and nine end-cap disks on either side, and the Transition Radiation Tracker (TRT) made of "straw tubes" which are proportional counters that contribute to particle identification via transition radiation; see Sect. 2.4.1. A charged particle hits at least 30 straw tubes on the way through the TRT; see Sect. 10.2.

The ATLAS muon spectrometer [35, Chap. 6] consists of a barrel part and two end-caps. The barrel spectrometer contains three concentric layers, each with eight large and eight small chambers of drift tubes. Each end-cap has four disks of drift tube chambers and cathode strip chambers. Resistive plate chambers on the barrel and thin gap chambers in the end-caps are used for trigger purposes.

1.6.1.3 CMS

CMS [38] is, besides ATLAS, the second general-purpose experiment at the LHC. Its vertex detector originally consisted of three barrel pixel layers and two end-cap pixel disks on either side [39]. In the winter of 2016/2017, this device was upgraded with a fourth barrel layer and a third end-cap disk on either side, giving at least four hits per track over the full solid angle covered by the detector [40].

The silicon strip tracker (SST) of CMS is the largest silicon tracker ever built. It is divided into four sections: the inner barrel (TIB), the outer barrel (TOB) and the two end-caps (TEC). Depending on its angle with respect to the beam axis, a charged particle crosses between eight and 14 sensors, out of which four to six are double-sided ones [41]. Track reconstruction in the SST is done mostly in conjunction with the Pixel Detector; see Sect. 10.3.

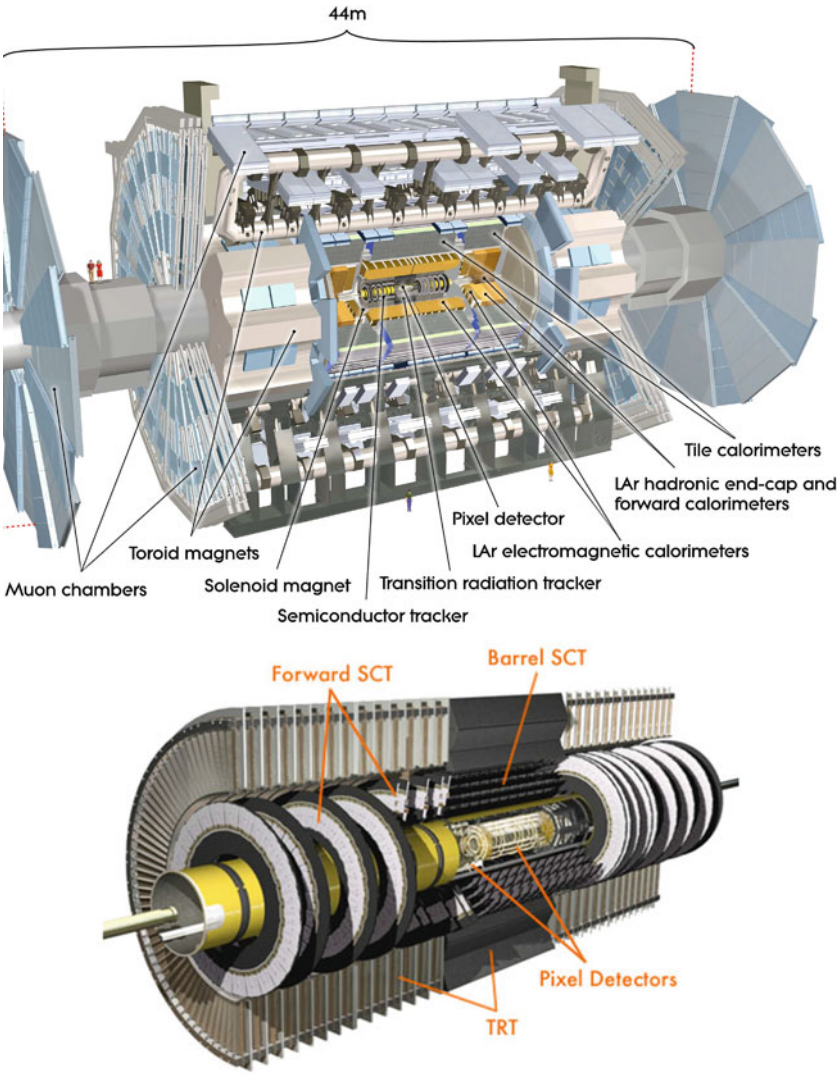

Fig. 1.6 Top: Cut-away view of the ATLAS detector. (From [35], reproduced under License CC-BY-3.0). Bottom: The central tracker. (From https://collaborationatlasfrance.web.cern.ch/content/tracker)

The CMS muon system [42] consists of four layers of muon stations inserted in the iron return yoke of the solenoid; see Fig. 1.7. The stations in the barrel region are equipped with drift tubes, and those in the end-caps are equipped with cathode strip chambers. In addition, resistive plate chambers are mounted in both the barrel and end-caps of CMS; they are used mainly for triggering.

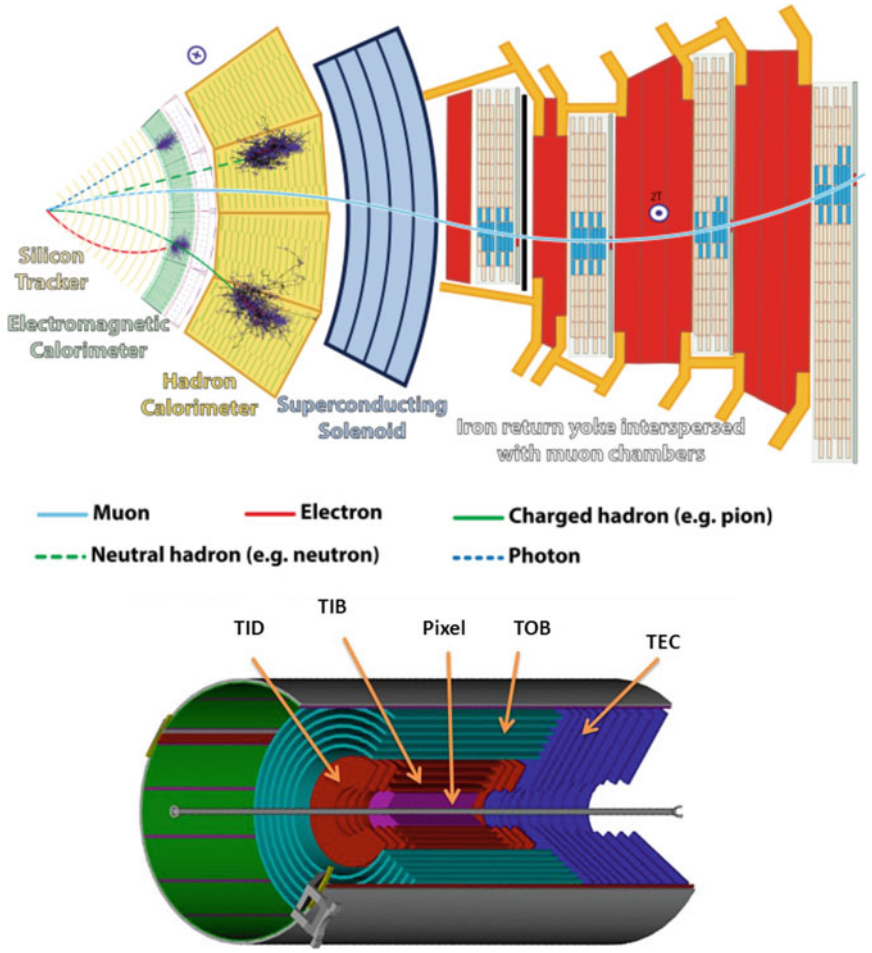

Fig. 1.7 Top: Schematic diagram of a sector of the barrel part of the CMS detector. (From [43], reproduced under License CC-BY-4.0). Bottom: Schematic view of the vertex detector and the silicon strip tracker. (Courtesy of W. Adam)

Figure 1.7 shows a schematic diagram of a sector of the barrel part of the CMS detector.

1.6.1.4 LHCb

LHCb [44] is the experiment at the LHC that is dedicated to precision measurements of CP violation and rare decays of B hadrons. Instead of surrounding the entire collision point with an enclosed detector as ATLAS and CMS, the LHCb experiment is designed to detect mainly particles in the forward direction.

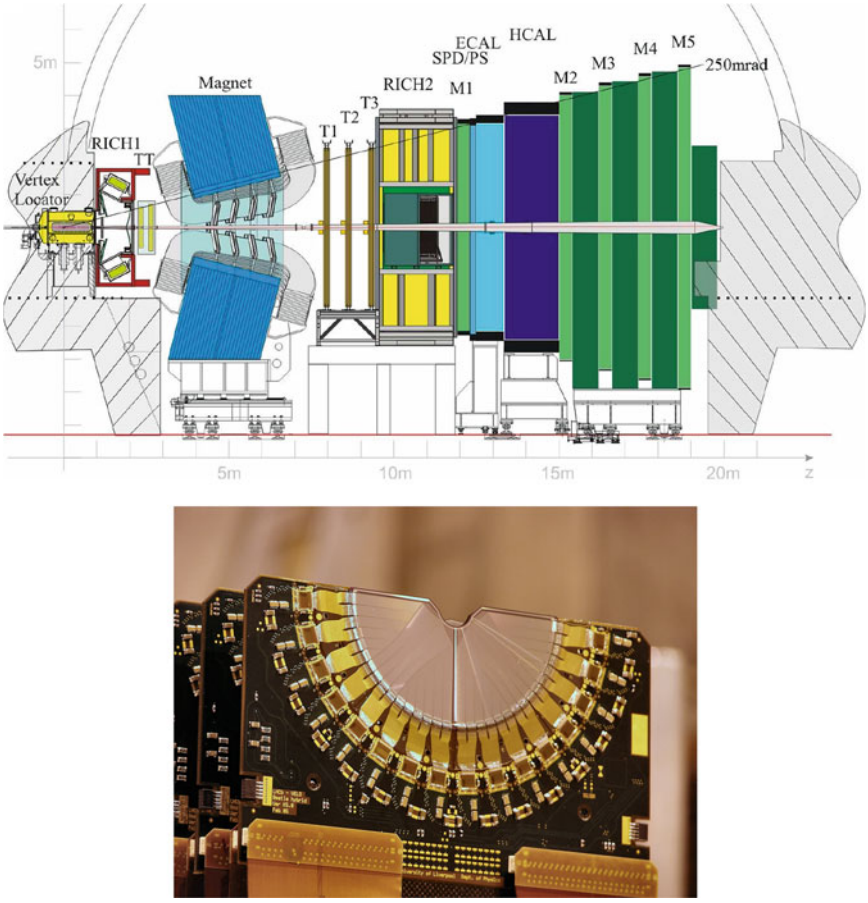

Fig. 1.8 Top: View of the LHCb detector. (From [44], reproduced under License CC-BY-3.0). Bottom: Sensor of the Vertex Locator (VELO)

The core of the LHCb [44] tracking system (see Fig. 1.8) is a silicon microstrip detector close to the interaction point, the Vertex Locator (VELO). It can be moved to a distance of only 7 mm from the proton beams and measures the position of the primary vertices and the impact parameters of the track with extremely high precision.

Up to end of 2018, the tracking downstream of the VELO was accomplished by the TT and the T stations. The Tracker Turicensis (TT) is a silicon microstrip detector placed upstream of the dipole magnet, which improves the momentum resolution of reconstructed tracks and reject pairs of tracks that in reality belong to the same particle. The magnet is placed behind the TT. It bends the flight path

of the particles in the x-z plane and therefore allows the determination of their momenta. The tracking system is completed by the T stations (T1-T2-T3), which, together with the information from the VELO and optionally the TT, determine the momentum and flight direction of the particles. The T stations are composed of silicon microstrip sensors close to the beam pipe and by straw tubes in the outer regions. For track reconstruction in LHCb, see Sect. 10.4.

After the upgrade of LHCb during 2019–2020 and starting with Run 3 of the LHC, the tracking downstream of the VELO is done by the SciFi, a homogeneous tracking system in scintillating fiber technology; see [20] and Sect. 1.4.

The LHCb muon system [44, Sect. 6.3], consisting of the muon stations M1 to M5, provides fast information for the muon trigger at Level 0 and muon identification for the high-level trigger and offline analysis. It comprises five stations interleaved with absorbers. The stations are mostly equipped with multi-wire proportional chambers, with the exception of the central part of the first chamber, where GEM detectors are used because of the high particle rate.

1.6.2 Belle II and CBM

1.6.2.1 Belle II

Belle II [45] is an experiment at the SuperKEKB collider at KEK in Japan. Its principal aim is the study of the properties of B mesons. The detector is shown in Fig. 1.9. The vertex detector consists of two parts, the pixel detector (PXD) with two layers of DEPFET pixels [46] and the Silicon Vertex Detector (SVD) with four layers of double-sided silicon strip detectors [47].

The central tracking device of Belle II is the CDC, a cylindrical drift chamber [48]. It has 56 layers of sense wires in nine superlayers, five with a total of 32 axial wire layers and four with 24 stereo wire layers. The stereo angle is between 2.6 and 4.2 degrees. For track reconstruction in Belle II, see Sect. 11.1.

The Belle II KLM system is designed to detect long-lived K-mesons and muons. It consists of alternating layers of iron plates, serving as flux return, and active detector elements. In the end-caps and the innermost two layers of the barrel, the active elements are scintillator strips; the rest of the barrel layers are equipped with resistive plate chambers, reused from Belle [49].

1.6.2.2 CBM

The Compressed Baryon Matter (CBM) experiment is a fixed target experiment [50] (see Fig. 1.10) at the future FAIR facility for antiproton and ion research. It is designed to investigate the properties of highly compressed baryon matter. Its

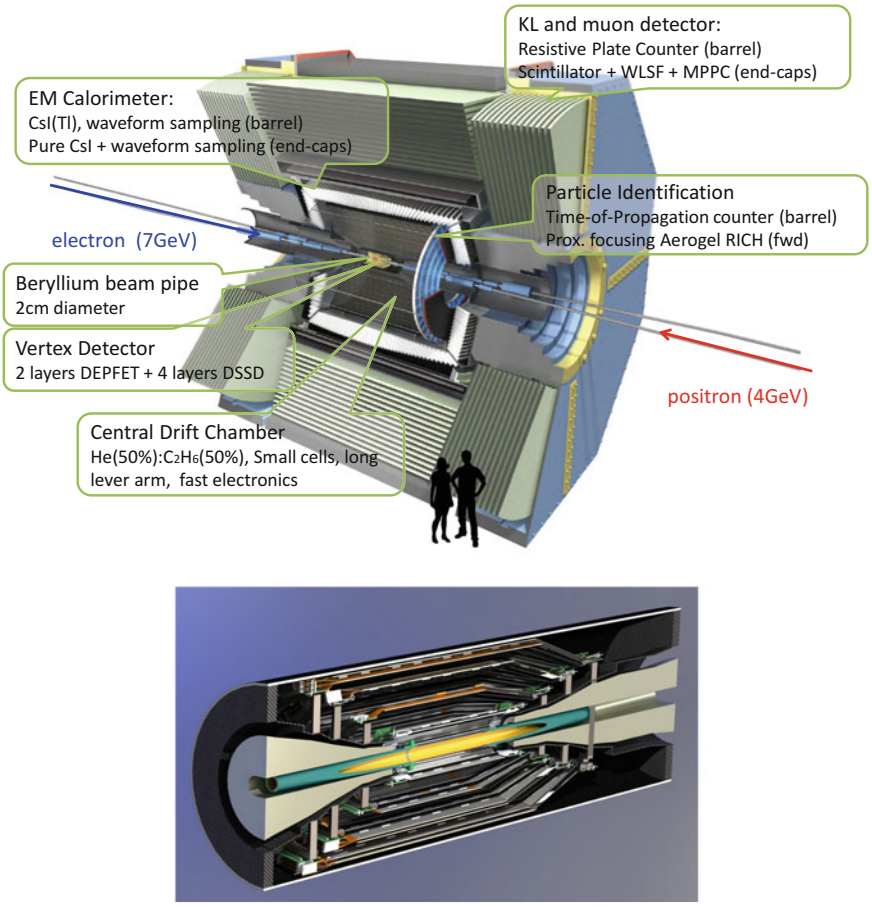

Fig. 1.9 Top: View of the Belle II detector. Bottom: The pixel detector and the silicon vertex detector

central tracking device is the Silicon Tracking System (STS [51]). It is designed for high multiplicity, up to 1000 charged particles per interaction, at high rates, up to 10 MHz, and consists of eight layers of double-sided silicon sensors between 30 and 100 cm downstream of the target, inside the magnetic field.

Each of the three Transition Radiation Detectors [52] is made of four MWPCs. Their main task is electron identification. Track reconstruction in the STS and the TRD is described in Sect. 11.2.

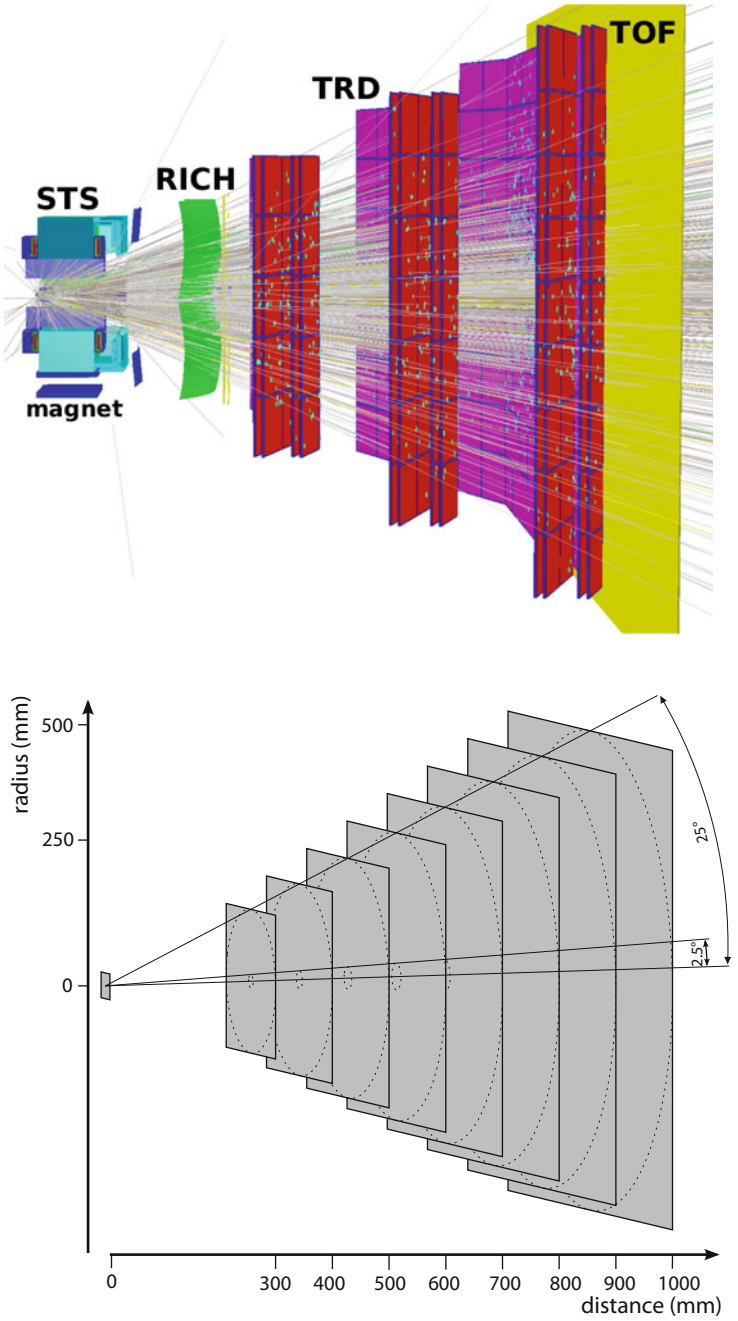

Fig. 1.10 Top: Schematic geometry of the CBM detector. (From [53]). Bottom: Schematic geometry of the silicon tracking system. (From [51])

References

1. C. Grupen, B. Shwartz, *Particle Detectors* (Cambridge University Press, 2008). https://books.google.at/books?id=XCPlJTu3GQkC
2. M. Krammer, W. Mitaroff, in *Handbook of Particle Detection and Imaging*, ed. by C. Grupen, I. Buvat (Springer, Berlin, 2012), p. 265
3. C. Grupen, I. Buvat (eds.), *Handbook of Particle Detection and Imaging* (Springer, Berlin/Heidelberg, 2012)
4. H. Crannell, G. Maurer, E. Carreira, Nucl. Instrum. Methods **142**(3), 455 (1977)
5. M. Pernicka, M. Regler, S. Sychkov, Nucl. Instrum. Methods **156**(1), 219 (1978)
6. W. Blum, W. Riegler, L. Rolandi, *Particle Detection with Drift Chambers*. Particle Acceleration and Detection (Springer, Berlin/Heidelberg, 2008). https://books.google.at/books?id=RgxUlAXv0RAC
7. A. Grossheim, J. Hu, A. Olin, Nucl. Instrum. Methods Phys. Res. A **623**(3), 954 (2010)
8. S. Bartalucci et al., Nucl. Instrum. Methods **192**(2), 223 (1982)
9. D.V. Thanh et al., in *Proceedings of the 2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)* (2017), p. 1
10. T. Dong et al., Nucl. Instrum. Methods Phys. Res. A **930**, 132 (2019)
11. G.V. Buren et al., Nucl. Instrum. Methods Phys. Res. A **566**(1), 22 (2006)
12. V. Bartsch et al., Nucl. Instrum. Methods Phys. Res. A **497**(2), 389 (2003)
13. K. Banzuzi et al., Nucl. Instrum. Methods Phys. Res. A **453**(3), 536 (2000)
14. M. Swartz et al., PoS **Vertex 2007**, 035 (2007). https://pos.sissa.it/057/035
15. P. Rehak et al., Nucl. Instrum. Methods Phys. Res. A **248**, 367 (1986)
16. E. Gatti, P. Rehak, Nucl. Instrum. Methods Phys. Res. A **541**, 47 (2005)
17. D. Nouais et al., Nucl. Instrum. Methods Phys. Res. A **501**(1), 119 (2003)
18. R. Ruchti, Ann. Rev. Nucl. Part. Sci. **46**, 281 (1996). https://www.annualreviews.org/doi/pdf/10.1146/annurev.nucl.46.1.281
19. C. Joram, G. Haefeli, B. Leverington, J. Instrum. **10**(08), C08005 (2015). https://iopscience.iop.org/article/10.1088/1748-0221/10/08/C08005
20. L. Gruber, on behalf of the LHCb SciFi tracker collaboration. Nucl. Instrum. Methods Phys. Res. A (2019). In press
21. S. Blusk, O. Buchmuller, A. Jacholkowski, T. Ruf, J. Schieck, S. Viret (eds.), *Proceedings of the First LHC Detector Alignment Workshop, CERN, Geneva, Switzerland* (2007). http://cdsweb.cern.ch/record/970621
22. 2nd LHC Detector Alignment Workshop (2007). https://indico.cern.ch/event/13681/timetable/?view=standard
23. 3rd LHC Detector Alignment Workshop (2009). https://indico.cern.ch/event/50502/timetable/?view=standard
24. P.A. Coe, D.F. Howell, R.B. Nickerson, Meas. Sci. Technol. **15**(11), 2175 (2004)
25. V. Blobel, C. Kleinwort, F. Meier, Comput. Phys. Commun. **182**(9), 1760 (2011)
26. V. Blobel, Millepede II. Talk given at the 2nd LHC Detector Alignment Workshop (2007). https://tinyurl.com/MillepedeII
27. V. Karimäki, T. Lampen, F.P. Schilling, The HIP algorithm for track based alignment and its application to the CMS pixel detector. Tech. Rep. CMS-NOTE-2006-018, CERN, Geneva (2006). http://cdsweb.cern.ch/record/926537
28. ALICE Collaboration, J. Instrum. **3**(08), S08002 (2008). https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08002
29. F. Meddi, for the ALICE Collaboration, Nucl. Instrum. Methods Phys. Res. A **465**(1), 40 (2001)
30. P. Riedler, ALICE Collaboration, Nucl. Phys. A **956**, 866 (2016)
31. C. Garabatos, for the ALICE Collaboration, Nucl. Instrum. Methods Phys. Res. A **535**(1), 197 (2004)
32. C. Lippmann, representing the ALICE TPC collaboration, Phys. Proc. **37**, 434 (2012)
33. W. Yu, for the ALICE TPC Collaboration, Nucl. Instrum. Methods Phys. Res. A **706**, 55 (2013)

34. D. Das, for the ALICE Collaboration, Nucl. Phys. A **862–863**, 223 (2011)
35. G. Aad et al., J. Instrum. **3**, S08003 (2008). https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08003
36. L. Rossi, Nucl. Instrum. Methods Phys. Res. A **580**(2), 1008 (2007)
37. F. Guescini, on behalf of the ATLAS Collaboration, Nucl. Instrum. Methods Phys. Res. A **796**, 60 (2015)
38. S. Chatrchyan et al., J. Instrum. **3**, S08004 (2008). https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08004
39. D. Bortoletto, representing the CMS Collaboration, Nucl. Instrum. Methods Phys. Res. A **579**(2), 669 (2007)
40. K. Klein, on behalf of the CMS Collaboration, Nucl. Instrum. Methods Phys. Res. A **845**, 101 (2017)
41. O. Pooth, Nucl. Instrum. Methods Phys. Res. A **569**(1), 21 (2006)
42. A. Sirunyan et al., J. Instrum. **13**(06), P06015 (2018). https://iopscience.iop.org/article/10.1088/1748-0221/13/06/P06015
43. CMS Detector Slice. CMS-PHO-GEN-2016-001 (2016). https://cds.cern.ch/record/2120661. Published under License CC-BY-4.0
44. A. Augusto Alves et al., J. Instrum. **3**, S08005 (2008). https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08005
45. J. Kahn, in *CERN-BINP Workshop for Young Scientists in* e^+e^- *Colliders*, ed. by V. Brancolini, L. Linssen (2017). http://cds.cern.ch/record/2301997
46. H.-G. Moser, on behalf of the DEPFET collaboration, Nucl. Instrum. Methods Phys. Res. A **831**, 85 (2016)
47. Y. Onuki, on behalf of the Belle II SVD Collaboration, Nucl. Instrum. Methods Phys. Res. A **765**, 99 (2014)
48. I. Adachi et al., on behalf of the Belle II Collaboration, Nucl. Instrum. Methods Phys. Res. A **907**, 46 (2018)
49. T. Aushev et al., Nucl. Instrum. Methods Phys. Res. A **789**, 134 (2015). https://arxiv.org/pdf/1406.3267v3.pdf
50. P. Senger, the CBM Collaboration, J. Phys.: Conf. Ser. **50**, 357 (2006). https://iopscience.iop.org/article/10.1088/1742-6596/50/1/048
51. J. Heuser et al. (eds.), *Technical Design Report for the CBM Silicon Tracking System (STS)* (GSI, Darmstadt, 2013). http://repository.gsi.de/record/54798
52. C. Blume, C. Bergmann, D. Emschermann (eds.), *The Transition Radiation Detector of the CBM Experiment at FAIR: Technical Design Report for the CBM Transition Radiation Detector (TRD)*. GSI-2018-01091 (Facility for Antiproton and Ion Research in Europe GmbH, Darmstadt, 2018). http://repository.gsi.de/record/217478
53. S. Lebedev, C. Hoehne, A. Lebedev, G. Ososkov, J. Phys. Conf. Ser. **396**, 2029 (2012)

Chapter 2
Event Reconstruction

Abstract The chapter gives an outline of the event reconstruction chain of a typical large experiment, from the trigger to the physics object reconstruction. The concept of the trigger is illustrated by two examples, CMS and LHCb, followed by a discussion of track reconstruction and its stages: hit generation, local reconstruction, and global reconstruction. The section on vertex reconstruction introduces a classification of vertices and sets the scene for the dedicated chapters on vertex finding and vertex fitting. The chapter concludes with some remarks on particle identification and reconstruction of physics objects such as electrons, muons, photons, jets, τ leptons, and missing energy.

2.1 Trigger and Data Acquisition

2.1.1 General Remarks

In the latest and most powerful collider, the Large Hadron Collider (LHC), bunches of protons collide with a nominal frequency of 40 MHz, i.e., every 25 ns. Due to gaps in the beams, at most 2556 bunches are actually present [1]. As the bunch revolution frequency equals 11,245 Hz, the average bunch crossing rate is about 28.7 MHz. The average number of individual proton collisions per bunch crossing, called the pile-up, depends on the luminosity. In the CMS experiment, it varied between about five and slightly more than 60 in 2018 [2], so that bunch crossings without collisions are extremely unlikely in CMS and likewise in ATLAS. The result of a bunch crossing with at least one collision is called an event. In CMS and ATLAS, the event rate is virtually the same as the bunch crossing rate. In the LHCb experiment, however, not all events are actually visible in the detector, so that the effective event rate is lower than the bunch crossing rate; see Sect. 2.1.3.

The most frequent value of pile-up observed by CMS in 2018 was around 30, corresponding to nearly 900 million individual collisions per second. Clearly, it is neither possible nor desirable for the experiment to record all of the event data produced during data taking. A selection mechanism is therefore required that tags

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_2

the physically interesting events and activates the recording mechanism, called data acquisition (DAQ). Such a selection mechanism is called a trigger.

Triggers have been deployed for many decades, ever since bubble chambers were superseded by experiments with electronic tracking detectors and calorimeters. Triggers are needed both in fixed target and in colliding beam experiments because of limitations in data rates, storage capacity and computing resources. In order to deal with the high event rates typical for electronic experiments, it is important to minimize the dead time of the trigger, i.e., the time after an event during which the system is not able to process another event. Triggers are therefore implemented in several stages or levels, with increasing computational complexity and decision time or latency. The principle and possible implementations are best demonstrated on examples. In the following the trigger/DAQ systems of the CMS and the LHCb experiments will be described in somewhat more detail.

2.1.2 The CMS Trigger System

CMS [3] is one of the two general purpose experiments at the LHC; see Sect. 1.6.1.3. Its trigger has two levels, the Level-1 Trigger (L1, [4]) and the High-Level Trigger (HLT, [5, 6]). During data taking virtually every bunch crossing results in at least one collision of protons, and the average primary event rate equals the average bunch crossing rate of 28.7 MHz. The trigger, however, must be able to process events separated by only 25 ns.

The task of the L1 trigger is to reduce the primary event rate to less than 100 kHz. This is achieved using high-speed customized hardware running up to 128 different algorithms. Its inputs come from the calorimeters, the muon detectors, and the beam monitors. Its latency is 3.2 μs, including data collection, decision making, and propagation back to the detector front-ends. A block diagram of the L1 trigger is shown in Fig. 2.1.

The input to the muon trigger (MT) comes from three different detector types. The local MTs find track segments, the regional MTs find tracks, and the global MT combines the information from all regional MTs, selects the best four muon candidates, and provides their momenta and directions.

The calorimeter trigger (CT) uses information from the both the electromagnetic and the hadronic calorimeter. The local CTs compute energy deposits, the regional CTs find candidates for electrons, photons, jets, isolated hadrons, and compute transverse energy sums. The transverse energy vector $\boldsymbol{E}_\mathrm{T}$ is defined as:

$$\boldsymbol{E}_\mathrm{T} = E \cdot \begin{pmatrix} \cos\phi \cos\lambda \\ \sin\phi \cos\lambda \end{pmatrix}, \tag{2.1}$$

where ϕ is the azimuth angle and $\lambda = \pi/2 - \theta$ is the dip angle of the particle or jet direction. The global CT sorts the candidates in all categories, computes total

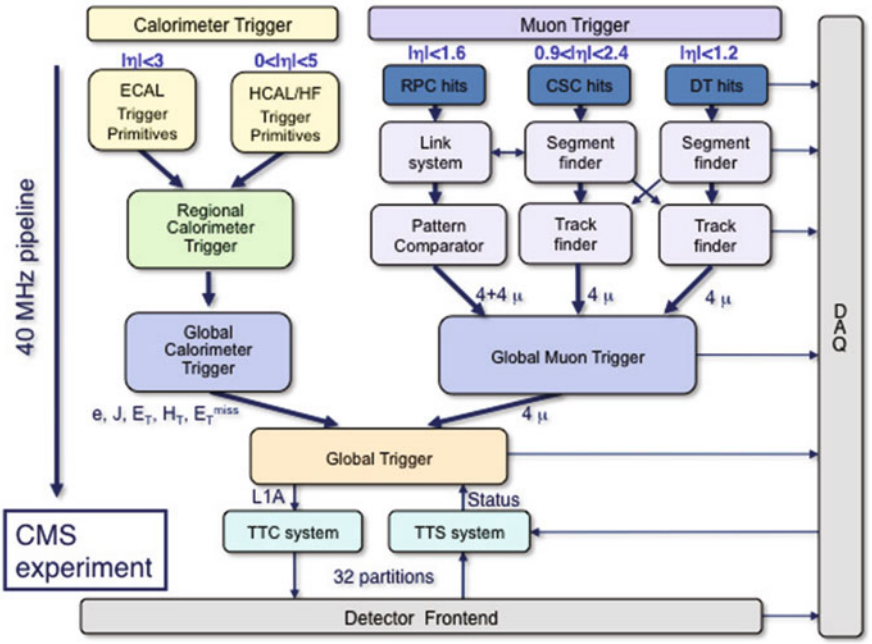

Fig. 2.1 Block diagram of the CMS L1 trigger. The details are explained in the text. (From [4], by permission of Elsevier)

and missing transverse energy sums, and computes jet multiplicities for different thresholds.

Finally, the global trigger makes the final decision which is passed on to the detector front-end electronics and the DAQ.

The HLT is designed to reduce the L1 output rate of 100 kHz to the final recording rate of $O(100)$ Hz. It is implemented purely in software which runs on a farm of commercial processors, using the full event data and performing the reconstruction and selection of physics objects such as electrons, photons, muons, τ leptons, hadronic jets, and missing energy; see Sect. 2.4.

2.1.3 The LHCb Trigger System

Although the intersection point of LHCb is tuned to lower luminosity than the one of CMS and ATLAS, the number of produced $b\bar{b}$ pairs is still of the order of 10^{11} per year. An efficient, robust and flexible trigger is required in order to cope with the harsh hadronic environment. It must be sensitive to many different final states.

While the average bunch crossing rate is 28.7 MHz, the frequency of events actually visible in the detector is about 13 MHz [7]. This must be reduced to about

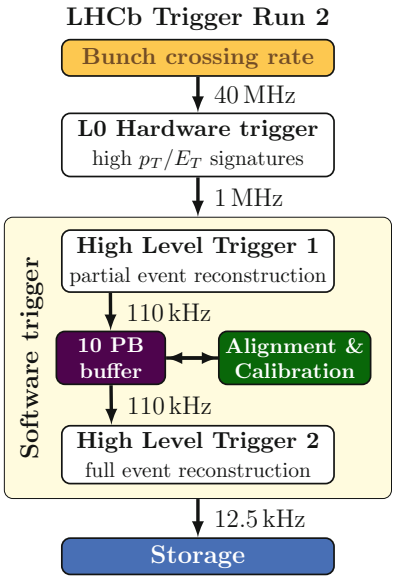

Fig. 2.2 Scheme of the LHCb trigger system. (From [8], reproduced under License CC-BY-4.0)

1 MHz, the frequency at which the detector can be read out. As in CMS, the trigger is organized in two levels, Level-0 (L0) and HLT (see Fig. 2.2).

The L0 trigger is implemented in field-programmable gate arrays (FPGAs) with a fixed latency of 4 µs [8]. It has two components: the calorimeter trigger, which looks for particles (electrons, photons, neutral pions, hadrons) with large transverse energies; and the muon trigger which uses the tracks reconstructed in the muon chambers and selects the two muons with the highest transverse momentum for each quadrant of the muon detector.

The HLT is implemented purely in software running on the nodes of the event filter farm [8]. It is subdivided in two stages, HLT1 and HLT2; see Sect. 10.4. HLT1 does a partial event reconstruction. Its output, at a rate of about 100 kHz, is directed to HLT2, which performs the full event reconstruction and writes the results to mass storage. For more details on track and vertex reconstruction, see Sect. 10.4.

2.2 Track Reconstruction

Track reconstruction is a central task in the analysis of the event data. Its aim is to provide estimates of the track parameters, i.e., the position, the direction, and the momentum of charged particles at one or several specific points or surfaces. For an excellent review of track reconstruction, see [9].

Only the tracks of stable or sufficiently long-lived charged particles—for instance electrons, protons, muons and charged pions—are visible in the tracking detectors. Short-lived particles, for instance B hadrons or J/ψ mesons, are reconstructed from

their decay products. The reconstruction of charged particles can be divided into the following steps:

A. Hit generation The recorded signals are converted to spatial coordinates, either 2D or 3D, using the detector-specific calibration constants. The coordinates will be called "measurements", "observations" or "hits" in the following. For some examples, see Sects. 1.2 and 1.3.

B. Local track reconstruction First, tracks are reconstructed in each tracking system. This can again be divided into two or three steps.

1. *Track segment reconstruction.* This step is relevant only for tracking systems that are in turn composed of several independent devices capable of giving at least the position and the direction of the particle and possibly the momentum. A typical example is the barrel muon system of the CMS experiment which consists of four muon stations, each with up to 12 layers of drift tubes. Although a muon station is not considered a fully-fledged tracking system in the context of track reconstruction, it provides enough hits to estimate the parameters of a straight line track segment [10].
2. *Track finding.* In this step, hits or track segments are clustered to track candidates, i.e., collections of hits, such that ideally all hits or segments in a collection are produced by the same particle. Track finding can be done iteratively, especially in the very-high multiplicity events recorded by the LHC experiments. In this approach, "easy" tracks with high momentum and small material effects are found first, while more difficult tracks are found in the subsequent passes (see Sect. 10.3). Algorithms for track finding are discussed in detail in Chap. 5. Specific solutions by experiments are discussed in Chaps. 10 and 11.
3. *Track fitting.* For each track candidate, the track model is fitted to the hits in order to get the best estimates of the track parameters. This is the topic of Chap. 6. The track fit also gives an indication of the quality of the fit, usually in the form of a chi-square statistic χ^2. An abnormally large value of the chi-square statistic indicates either a random combination of hits, i.e., a "fake" or "ghost" track, or the presence of outliers in the track candidate. Outliers can either be removed from the track or down-weighted by employing a robust estimator; see Sects. 6.2 and 6.4.2.

C. Global track reconstruction After the local track reconstruction, the tracks found in the individual tracking systems have to be combined to global track candidates. To this end, the track candidates accepted by the track fit in the main tracking system are extrapolated to the other tracking systems and checked for compatibility with the tracks reconstructed there. As the estimated track parameters in different tracking systems are stochastically independent, the test for compatibility is usually based on the chi-square statistic of their weighted mean, but more sophisticated machine learning methods can be applied as well [11]. The successful combination of local tracks to a global candidate is normally followed by a track fit of the latter.

Some tracking systems do not have enough redundancy to allow stand-alone local track reconstruction, for instance a pixel vertex detector with only two layers. In this

case, the tracks found in the global track reconstruction are extrapolated and used to find compatible hits, which are then attached to the track; see Sect. 5.1.7.

D. Assessment of track quality Not every track candidate generated by the track finding is a valid track. Testing the track hypothesis and assessing the track quality after the track fit is therefore mandatory. This is the topic of Sect. 6.4.

Given a careful calibration of the tracking devices (Sects. 1.2 and 1.3) the chi-square statistic of the track fit is used to reject fake tracks (Sect. 6.4.1) and to trigger the search for outliers (Sect. 6.4.2) or kinks (Sect. 6.4.3). Another important ingredient to the assessment of track quality is the track length or the effective number of hits attached to the track after removal or down-weighting of outliers; see Sect. 6.4.1. Checking the compatibility with the collision or production region can also be used to reject fake tracks.

If the track reconstruction proceeds iteratively, it is tempting to remove the hits used in an iteration from the hit pool to simplify the task of the subsequent iterations. It is, however, advisable to remove only those hits that are unambiguously attached to tracks of the highest quality.

2.3 Vertex Reconstruction

A point where particles are produced in a collision or a decay is called a vertex. The point of collision of two beam particles in a collider or of a beam particle with a target particle in a fixed-target experiment is called the primary vertex. In high-luminosity colliders, such as the LHC, many collisions occur in a single bunch crossing; consequently, there are many primary vertices. It is, however, statistically almost certain that at most one of the collisions generates a pattern recognized by the trigger as being of potential physical interest. The vertex of this collision is called the signal vertex.

Many of the particles produced at a primary vertex, including the signal vertex, are unstable and decay at a secondary vertex. The aim of of vertex reconstruction is to find sets of particles that have been produced at the same vertex, to estimate the vertex position, test whether the assignment of the particles to the vertex is correct, and improve the estimates of the track parameters by imposing the vertex constraint. Vertex finding and vertex fitting are the topics of Chaps. 7 and 8, respectively. For the various types of secondary vertices and how to find them, see Chap. 9. Examples of experimental strategies are given in Chaps. 10 and 11.

2.4 Physics Objects Reconstruction

Both the trigger and the physics analysis require not just tracks, but objects that represent physical entities such as electrons, photons, muons, τ leptons, jets, missing energy, etc. Object identification can be obtained by two complementary

approaches: dedicated detectors for particle identification (PID), and combining information from different sub-detectors.

2.4.1 Particle ID by Dedicated Detectors

Charged particles can be identified by dedicated detectors in various ways [12].

Measurement of the velocity Given the momentum as determined by the tracking system, the mass can be estimated. Velocity can be measured directly by time-of-flight detectors [13], or indirectly by measuring the emission angle of Cherenkov radiation in Cherenkov detectors [14] and time-of-propagation detectors [15].

Energy loss by ionization In a large range of velocity, the expected energy loss by ionization is proportional to $(m/p)^2$, where m is the unknown mass and p is the momentum of the particle; see Sect. 4.5.2.1. In practice, the most probable energy loss is estimated from a number of measurements. In a silicon tracker, energy loss is measured in each sensor [16]; in a drift or time projection chamber, the energy loss is measured for each wire hit or for each cluster in the endplates, respectively [17].

Transition radiation Transition radiation (TR) is electromagnetic radiation in the X-ray band. It is emitted when an ultra-relativistic particle crosses the boundary between two media with different dielectric constants. The radiator is combined with a gaseous sensing device that measures the TR signal and the position of the particle [13, 18].

2.4.2 Particle and Object ID by Tracking and Calorimetry

PID in dedicated detectors is complemented by combining information from the tracking systems and the calorimeters.

Electrons Electrons and positrons are identified as such by the fact that they have a reconstructed track and a cluster in the electromagnetic calorimeter that matches the track in energy and position. For the special treatment of electrons in the track fitting, see Sects. 6.2.3, 10.2 and 10.3.

Photons Clusters in the electromagnetic calorimeter that are not matched to a track or a cluster in the hadronic calorimeters are candidates for photons.

Muons Global tracks with hits in both the central tracking system and the muon tracking system are candidates for muons.

Jets Jets are narrow bundles of charged and neutral particles produced by the hadronization of a quark or a gluon. Jet reconstruction algorithms are based on clustering the charged tracks, but should also provide a good correspondence

between the energy deposits in the calorimeters and the reconstructed tracks. This is the aim of the particle flow method, which originated in the ALEPH experiment at the LEP collider [19], and is now employed by LHC experiments as well [20, 21]. In this approach, the energy of a charged hadron is estimated by a weighted average of the track momentum and the associated calorimeter energy. The energy of the photons is measured by the electromagnetic calorimeter, and the energy of neutral hadrons is measured by the hadronic calorimeter.

Tau leptons Tau leptons have to be reconstructed from their decay products (Sect. 9.2). In two thirds of the cases, τ leptons decay into hadrons, typically into one or three charged mesons (predominantly pions), often accompanied by neutral pions decaying into photons and an invisible neutrino. Therefore, the particle flow approach can be applied to τ leptons as well [22, 23].

Missing energy Missing transverse energy is a signature for invisible particles such as neutrinos, dark matter, and neutral supersymmetric particles. In a typical collider experiment, it is a global quantity computed from the transverse momentum/energy components of all jets, electrons, photons, muons, and τ leptons in the event. In the LHC experiments, it has to be corrected for contributions from the pile-up collisions in the same bunch crossing; see for instance [24].

References

1. LHC report: full house for the LHC (2017). https://home.cern/news/news/accelerators/lhc-report-full-house-lhc
2. CMS Collaboration. Public CMS Luminosity Information (2018). https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults
3. S. Chatrchyan et al., J. Instrum. **3**, S08004 (2008). https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08004
4. P. Klabbers, for the CMS Collaboration, Phys. Proc. **37**, 1908 (2012)
5. N. Neumeister, for the CMS Collaboration, Nucl. Instrum. Methods Phys. Res. A **462**(1), 254 (2001)
6. D. Giordano, on behalf of the CMS Collaboration, Nucl. Phys. B Proc. Suppl. **150**, 299 (2006)
7. T. Head on behalf of the LHCb trigger project, J. Instrum. **9**(09), C09015 (2014). https://iopscience.iop.org/article/10.1088/1748-0221/9/09/C09015
8. R. Aaij et al., Performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC. Tech. Rep. CERN-LHCb-DP-2019-001, CERN, Geneva (2018). https://cds.cern.ch/record/2652801
9. R. Mankel, Rept. Prog. Phys. **67**, 553 (2004). https://arxiv.org/pdf/physics/0402039.pdf
10. The CMS collaboration, J. Instrum. **8**(11), P11002 (2013). https://iopscience.iop.org/article/10.1088/1748-0221/8/11/P11002
11. R. Frühwirth, Nucl. Instrum. Methods Phys. Res. A **356**, 493 (1995)
12. J. Engelfried, in *Handbook of Particle Detection and Imaging*, ed. by C. Grupen, I. Buvat (Springer, Berlin, Heidelberg, 2012), p. 125
13. O. Ullaland, R. Forty, in *Landolt-Börnstein – Group I Elementary Particles, Nuclei and Atoms 21B1 (Detectors for Particles and Radiation. Part 1: Principles and Methods)*, ed. by C. Fabjan, H. Schopper (Springer, Berlin, Heidelberg, 2011). https://materials.springer.com/bp/docs/978-3-642-03606-4

14. B. Ratcliff, J. Schwiening, in *Handbook of Particle Detection and Imaging*, ed. by C. Grupen, I. Buvat (Springer, Berlin, Heidelberg, 2012), p. 453
15. J. Fast, on behalf of the Belle II Barrel Particle Identification Group, Nucl. Instrum. Methods Phys. Res. A **876**, 145 (2017)
16. L. Quertenmont, Nucl. Phys. B Proc. Suppl. **215**(1), 95 (2011)
17. W. Yu, for the ALICE TPC Collaboration, Nucl. Instrum. Methods Phys. Res. A **706**, 55 (2013)
18. B. Mindur, on behalf of the ATLAS Collaboration, Nucl. Instrum. Methods Phys. Res. A **845**, 257 (2017)
19. D. Buskulic et al., Nucl. Instrum. Methods Phys. Res. A **360**(3), 481 (1995)
20. M. Aaboud et al., Eur. Phys. J. C **77**(7), 466 (2017)
21. A.M. Sirunyan et al., J. Instrum. **12**(10), P10003 (2017). https://iopscience.iop.org/article/10.1088/1748-0221/12/10/P10003
22. C.F. Galea, on behalf of the ATLAS Collaboration, Nucl. Part. Phys. Proc. **287–288**, 111 (2017)
23. CMS Collaboration, J. Instrum. **7**(01), P01001 (2012). https://iopscience.iop.org/article/10.1088/1748-0221/7/01/P01001
24. P. Berta, on behalf of the ATLAS Collaboration, Nucl. Part. Phys. Proc. **273–275**, 1121 (2016)

-

Chapter 3
Statistics and Numerical Methods

Abstract The chapter gives an outline of some statistical and numerical methods that will be applied in later chapters. The first section deals with the minimization of functions. Several gradient-based methods and a popular non-gradient method are discussed. The following section discusses statistical models and the estimation of model parameters. The basics of linear and nonlinear regression models and state space models are presented, including least-squares estimation and the (extended) Kalman filter. The final section gives a brief overview of clustering and different types of clustering algorithms.

3.1 Function Minimization

The minimization (or maximization) of a multivariate function $F(\boldsymbol{x})$ is a frequent task in solving non-linear systems of equations, clustering, maximum-likelihood estimation, function and model fitting, supervised learning, and similar problems. A basic classification of minimization methods distinguishes between methods that require the computation of the gradient or even the Hessian matrix of the function and gradient-free methods. All methods discussed in the following subsections are iterative and require a suitable starting point $\boldsymbol{x}_0$. Implementations of the methods discussed in the following, along with many others, can, for instance, be found in the MATLAB® Optimization Toolbox™ [1] and in the Python package scipy.optimize [2].

3.1.1 Newton–Raphson Method

If $F(\boldsymbol{x})$ is at least twice continuously differentiable in its domain, it can be approximated by its second order Taylor expansion $\hat{F}$ at the starting point $\boldsymbol{x}_0$:

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_3

$$F(\boldsymbol{x}_0 + \boldsymbol{h}) \approx \hat{F}(\boldsymbol{x}) = F(\boldsymbol{x}_0) + \boldsymbol{g}(\boldsymbol{x}_0)\,\boldsymbol{h} + \frac{1}{2}\,\boldsymbol{h}^{\mathsf{T}}\boldsymbol{H}(\boldsymbol{x}_0)\,\boldsymbol{h}, \tag{3.1}$$

where $\boldsymbol{g}(\boldsymbol{x}) = \nabla F(\boldsymbol{x})$ is the gradient and $\boldsymbol{H}(\boldsymbol{x}) = \nabla^2 F(\boldsymbol{x})$ is the Hessian matrix of $F(\boldsymbol{x})$. The step $\boldsymbol{h}$ is determined such that $\hat{F}$ has a stationary point at $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{h}$, i.e., that the gradient of $\hat{F}$ is zero at $\boldsymbol{x}_1$:

$$\nabla \hat{F}(\boldsymbol{x}_1) = \boldsymbol{g}(\boldsymbol{x}_0) + \boldsymbol{h}^{\mathsf{T}}\boldsymbol{H}(\boldsymbol{x}_0) = \boldsymbol{0} \Longrightarrow \boldsymbol{h} = -[\boldsymbol{H}(\boldsymbol{x}_0)]^{-1}\boldsymbol{g}(\boldsymbol{x}_0)^{\mathsf{T}}. \tag{3.2}$$

Note that if $\boldsymbol{x}$ is a column vector, $\boldsymbol{g}(\boldsymbol{x}_0)$ is a row vector. In order to ensure that the Wolfe conditions [3] are satisfied, Eq. (3.2) is often relaxed to:

$$\boldsymbol{h} = -\eta[\boldsymbol{H}(\boldsymbol{x}_0)]^{-1}\boldsymbol{g}(\boldsymbol{x}_0)^{\mathsf{T}}, \tag{3.3}$$

with a learning rate $\eta \in (0, 1)$. Inverting the Hessian matrix can be computationally expensive; in this case, $\boldsymbol{h}$ can be computed by finding an approximate solution to the linear system $\boldsymbol{H}(\boldsymbol{x}_0) \cdot \boldsymbol{h} = -\boldsymbol{g}(\boldsymbol{x}_0)^{\mathsf{T}}$. This procedure is iterated to produce a sequence of values according to:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta[\boldsymbol{H}(\boldsymbol{x}_k)]^{-1}\boldsymbol{g}(\boldsymbol{x}_k)^{\mathsf{T}}. \tag{3.4}$$

If the starting point $\boldsymbol{x}_0$ is sufficiently close to a local minimum, the sequence converges quadratically to the local minimum. In practice, the iteration is stopped as soon as the norm $|\boldsymbol{g}(\boldsymbol{x}_k)|$ of the gradient falls below some predefined bound ε. If F is a convex function, the local minimum is also the global minimum.

3.1.2 Descent Methods

As the computation of the Hessian matrix is computationally costly, various methods that do not require it have been devised, for instance, descent methods. A descent method is an iterative algorithm that searches for an approximate minimum of F by decreasing the value of F in every iteration. The iteration has the form $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k\,\boldsymbol{d}_k$, where $\boldsymbol{d}_k$ is the search direction and η_k is the step-size parameter. As with the Newton–Raphson method, when a (local) minimum is reached, it cannot be left anymore.

3.1.2.1 Line Search

A search direction $\boldsymbol{d}$ is called a descent direction at the point $\boldsymbol{x} \in \mathbb{R}^n$ if $\boldsymbol{g}(\boldsymbol{x}) \cdot \boldsymbol{d} < 0$. If η is sufficiently small, then

$$F(\boldsymbol{x} + \eta\,\boldsymbol{d}) < F(\boldsymbol{x}). \tag{3.5}$$

Once a search direction $\boldsymbol{d}_k$ has been chosen at the point $\boldsymbol{x}_k$, line search implies that the line $\boldsymbol{x} = \boldsymbol{x}_k + \lambda \boldsymbol{d}_k$ is followed to its closest local minimum. For various line search methods such as fixed and variable step size, interpolation, golden section or Fibonacci's method, see [4, 5].

3.1.2.2 Steepest Descent

Steepest descent follows the opposite direction of the gradient. If it is combined with line search, each search direction is orthogonal to the previous one, leading to a zig-zag search path. This can be very inefficient in the vicinity of a minimum where the Hessian matrix has a large condition number (ratio of the largest to the smallest eigenvalue); see Fig. 3.1.

3.1.2.3 Quasi-Newton Methods

In the Newton-Raphson method, the search direction is $\boldsymbol{d} = -[\boldsymbol{H}(\boldsymbol{x}_k)]^{-1}\boldsymbol{g}(\boldsymbol{x}_k)^{\mathsf{T}}$. If $\boldsymbol{H}(\boldsymbol{x}_k)$ is positive definite, then $\boldsymbol{d}$ is a descent direction, and so is $-\boldsymbol{A}\boldsymbol{g}(\boldsymbol{x}_k)^{\mathsf{T}}$ for any positive definite matrix $\boldsymbol{A}$. In a quasi-Newton method, $\boldsymbol{A}$ is constructed as an approximation to the inverse Hessian matrix, using only gradient information. The initial search direction is the negative gradient, $\boldsymbol{d}_0 = -\boldsymbol{g}(\boldsymbol{x}_0)^{\mathsf{T}}$, and the initial matrix $\boldsymbol{A}_0$ is the identity matrix. Each iteration performs a line search along the current search direction:

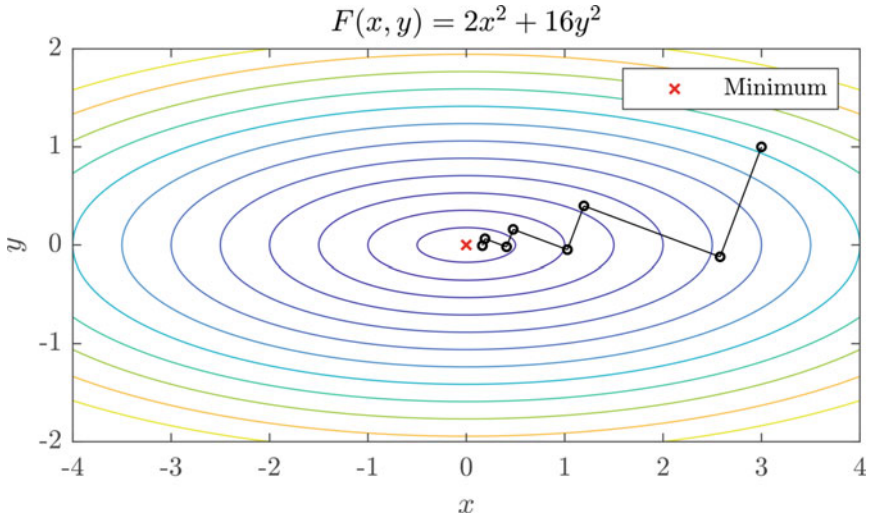

Fig. 3.1 Contour lines of the function $F(x, y) = 2x^2 + 16y^2$, and steepest descent with line search starting at the point $\boldsymbol{x}_0 = (3; 1)$

$$\lambda_k = \arg_\lambda \min_\lambda (\boldsymbol{x}_k + \lambda \boldsymbol{d}_k), \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k. \tag{3.6}$$

The new search direction is then computed according to:

$$\boldsymbol{d}_{k+1} = -\boldsymbol{A}_{k+1} \boldsymbol{g}(\boldsymbol{x}_{k+1})^\mathsf{T}, \tag{3.7}$$

where $\boldsymbol{A}_{k+1}$ is the updated approximation to the inverse Hessian matrix. There are two different algorithms for computing the update [6, p. 422], both using the change of the gradient along the step, denoted by $\boldsymbol{y}_k = [\boldsymbol{g}(\boldsymbol{x}_{k+1}) - \boldsymbol{g}(\boldsymbol{x}_k)]^\mathsf{T}$.

Davidon–Fletcher–Powell algorithm

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \lambda_k \frac{\boldsymbol{d}_k \boldsymbol{d}_k^\mathsf{T}}{\boldsymbol{d}_k^\mathsf{T} \boldsymbol{y}_k} - \frac{\boldsymbol{A}_k \boldsymbol{y}_k \boldsymbol{y}_k^\mathsf{T} \boldsymbol{A}_k}{\boldsymbol{y}_k^\mathsf{T} \boldsymbol{A}_k \boldsymbol{y}_k}. \tag{3.8}$$

Broyden–Fletcher–Goldfarb–Shanno algorithm

$$\boldsymbol{A}_{k+1} = \lambda_k \frac{\boldsymbol{d}_k \boldsymbol{d}_k^\mathsf{T}}{\boldsymbol{d}_k^\mathsf{T} \boldsymbol{y}_k} + \left(\boldsymbol{I} - \frac{\boldsymbol{d}_k \boldsymbol{y}_k^\mathsf{T}}{\boldsymbol{d}_k^\mathsf{T} \boldsymbol{y}_k} \right) \boldsymbol{A}_k \left(\boldsymbol{I} - \frac{\boldsymbol{y}_k \boldsymbol{d}_k^\mathsf{T}}{\boldsymbol{y}_k^\mathsf{T} \boldsymbol{d}_k} \right). \tag{3.9}$$

3.1.2.4 Conjugate Gradients

If the function $F(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^n$ is quadratic of the form $F(\boldsymbol{x}) = \frac{1}{2} \boldsymbol{x}^\mathsf{T} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{b}^\mathsf{T} \boldsymbol{x} + c$ with positive definite $\boldsymbol{A}$, the global minimum can be found in exactly n steps, if line search with a set of conjugate search directions is used. Such a set $\{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n\}$ is characterized by the following conditions:

$$\boldsymbol{d}_i^\mathsf{T} \boldsymbol{A} \boldsymbol{d}_j = 0, \quad i \neq j, \quad \boldsymbol{d}_i^\mathsf{T} \boldsymbol{A} \boldsymbol{d}_i \neq 0, \quad i = 1, \ldots, n. \tag{3.10}$$

The set is linearly independent and a basis of $\mathbb{R}^n$. An example for $n = 2$ is shown in Fig. 3.2.

In the general case, the conjugate gradient method proceeds by successive approximations, generating a new search direction in every iteration. Given an approximation $\boldsymbol{x}_k$, the new search direction is $\boldsymbol{d}_k = -\boldsymbol{g}(\boldsymbol{x}_k)^\mathsf{T} + \beta_k \boldsymbol{d}_{k-1}$, where $\boldsymbol{d}_0$ is arbitrary. A line search along direction $\boldsymbol{d}_k$ gives the next approximation:

$$\lambda_k = \arg_\lambda \min(\boldsymbol{x}_k + \lambda \boldsymbol{d}_k), \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k. \tag{3.11}$$

Different variants of the algorithm exist corresponding to different prescriptions for computing β_k. Two of them are given here [6, p. 406–416].

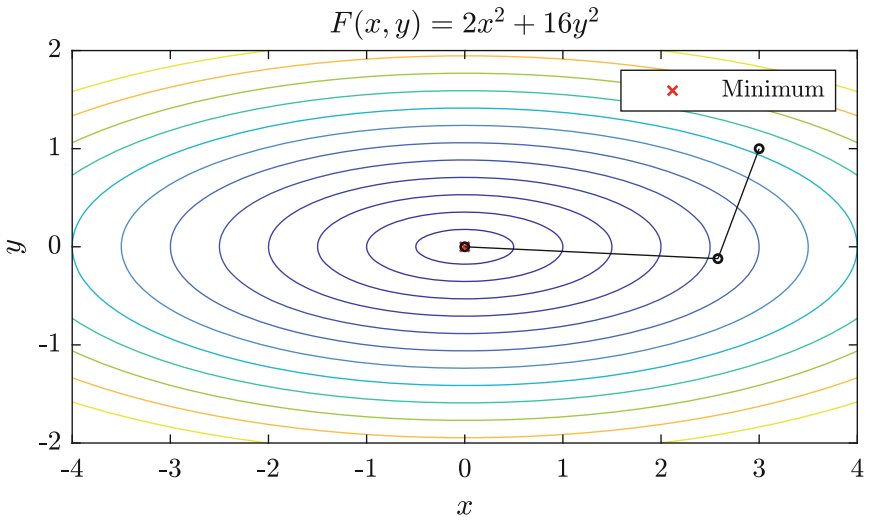

Fig. 3.2 Contour lines of the function $F(x, y) = 2x^2 + 16y^2$, and descent with line search and conjugate gradients starting at the point $\boldsymbol{x}_0 = (3; 1)$. The minimum is reached in two steps

Fletcher–Reeves algorithm

$$\beta_k = \frac{\boldsymbol{g}(\boldsymbol{x}_k)\,\boldsymbol{g}(\boldsymbol{x}_k)^\mathsf{T}}{\boldsymbol{g}(\boldsymbol{x}_{k-1})\,\boldsymbol{g}(\boldsymbol{x}_{k-1})^\mathsf{T}}. \tag{3.12}$$

Polak-Ribière algorithm

$$\beta_k = \frac{\boldsymbol{g}(\boldsymbol{x}_k)\,[\boldsymbol{g}(\boldsymbol{x}_k) - \boldsymbol{g}(\boldsymbol{x}_{k-1})]^\mathsf{T}}{\boldsymbol{g}(\boldsymbol{x}_{k-1})\,\boldsymbol{g}(\boldsymbol{x}_{k-1})^\mathsf{T}}. \tag{3.13}$$

It is customary to set β_k to zero if k is a multiple of n, in order to avoid accumulation of rounding errors. As such, convergence to the minimum in n steps is no longer guaranteed for non-quadratic $F(\boldsymbol{x})$.

3.1.3 Gradient-Free Methods

A popular gradient-free method is the downhill-simplex or Nelder–Mead algorithm [7]. It can be applied to functions whose derivatives are unknown, do not exist everywhere, or are too costly or difficult to compute. In n dimensions, the method stores $n+1$ test points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n+1}$ at every iteration, ordered by increasing function values, and the centroid $\boldsymbol{x}_0$ of all points but the last one. The simplex

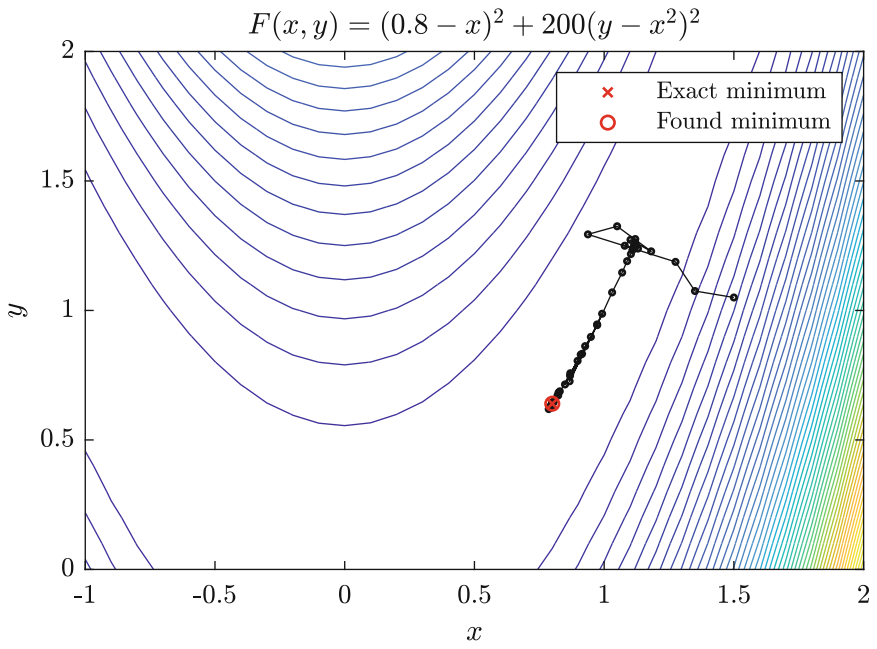

Fig. 3.3 Contour lines of the Rosenbrock function $F(x, y) = (0.8 - x)^2 + 200\,(y - x^2)^2$ and minimization with the downhill-simplex method starting at the point $\boldsymbol{x}_0 = (1.5; 1)$. With the tolerance 10^{-8} on the function value the minimum is reached after 90 steps

generated by the test points is then modified according to the function values in the test points. The allowed modifications are: reflection, expansion, contraction and shrinking. The iteration is terminated when the function value of the best point does not change significantly anymore. The size of the initial simplex is important; choosing it too small can lead to a very localized search. On the other hand, it is possible to escape from a local minimum by restarting the search with a sufficiently large simplex.

An example with the Rosenbrock function $F(x, y) = (0.8-x)^2+200\,(y-x^2)^2$ is shown in Fig. 3.3. The function has a very shallow minimum at $x = 0.64$, $y = 0.8$. With the tolerance 10^{-8} on the function value the minimum is reached after 90 steps.

Other gradient-free methods are simulated annealing, tabu search, particle swarm optimization, genetic algorithms, etc.

3.2 Statistical Models and Estimation

In the context of this book a statistical model is defined as a functional dependence of observed quantities (observations or measurements) on unknown quantities of interest (parameters or state vectors). The parameters cannot be observed directly,

and the observations are subject to stochastic uncertainties. The aim is to estimate the parameters from the observations according to some criterion of optimality.

3.2.1 Linear Regression Models

A linear regression model has the following general form [8]:

$$\boldsymbol{m} = \boldsymbol{F}\,\boldsymbol{p} + \boldsymbol{c} + \boldsymbol{\varepsilon}, \quad \mathsf{E}\,[\boldsymbol{\varepsilon}] = \boldsymbol{0}, \quad \mathsf{Var}\,[\boldsymbol{\varepsilon}] = \boldsymbol{V} = \boldsymbol{G}^{-1}, \tag{3.14}$$

where $\boldsymbol{m}$ is the $(n \times 1)$-vector of observations, $\boldsymbol{F}$ is the known $(n \times m)$ model matrix with $m \leq n$ and assumed to be of full rank, $\boldsymbol{p}$ is the $(m \times 1)$ vector of model parameters, $\boldsymbol{c}$ is a known constant offset, and $\boldsymbol{\varepsilon}$ is the $(n \times 1)$ vector of observation errors with zero expectation and $(n \times n)$ covariance matrix $\boldsymbol{V}$, assumed to be known.

LS estimation of $\boldsymbol{p}$ requires the minimization of the following objective function:

$$\mathcal{S}(\boldsymbol{p}) = (\boldsymbol{m} - \boldsymbol{F}\,\boldsymbol{p} - \boldsymbol{c})^{\mathsf{T}}\,\boldsymbol{G}\;(\boldsymbol{m} - \boldsymbol{F}\,\boldsymbol{p} - \boldsymbol{c})\,. \tag{3.15}$$

The least-squares (LS) estimator $\tilde{\boldsymbol{p}}$ and its covariance matrix $\boldsymbol{C}$ are given by:

$$\tilde{\boldsymbol{p}} = (\boldsymbol{F}^{\mathsf{T}}\boldsymbol{G}\,\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathsf{T}}\boldsymbol{G}(\boldsymbol{m} - \boldsymbol{c}), \quad \boldsymbol{C} = (\boldsymbol{F}^{\mathsf{T}}\boldsymbol{G}\,\boldsymbol{F})^{-1}. \tag{3.16}$$

The estimator $\tilde{\boldsymbol{p}}$ is unbiased and the estimator with the smallest covariance matrix among all estimators that are linear functions of the observations. If the distribution of $\boldsymbol{\varepsilon}$ is a multivariate normal distribution, the estimator is efficient, i.e., has the smallest possible covariance matrix among all unbiased estimators.

The residuals $\boldsymbol{r}$ of the regression are defined by:

$$\boldsymbol{r} = \boldsymbol{m} - \boldsymbol{c} - \boldsymbol{F}\,\tilde{\boldsymbol{p}}, \quad \boldsymbol{R} = \mathsf{Var}\,[\boldsymbol{r}] = \boldsymbol{V} - \boldsymbol{F}\,(\boldsymbol{F}^{\mathsf{T}}\boldsymbol{G}\,\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathsf{T}}. \tag{3.17}$$

The standardized residuals $\boldsymbol{s}$, also called the "pulls" in high-energy physics, are given by:

$$s_i = \frac{r_i}{\sqrt{\boldsymbol{R}_{ii}}}, \quad i = 1, \ldots, n. \tag{3.18}$$

If the model is correctly specified, the pulls have mean 0 and standard deviation 1. The chi-square statistic of the regression is defined as:

$$\chi^2 = \boldsymbol{r}^{\mathsf{T}}\boldsymbol{G}\,\boldsymbol{r}, \quad \text{with } \mathsf{E}\left[\chi^2\right] = n - m. \tag{3.19}$$

If the observation errors are normally distributed, χ^2 is χ^2-distributed with $d = n - m$ degrees of freedom; its expectation is d and its variance is $2d$. Its p-value p is defined by the following probability transform:

$$p = 1 - G_d(\chi^2) = \int_{\chi^2}^{\infty} g_d(x)\,\mathrm{d}x, \tag{3.20}$$

where $G_k(x)$ is the cumulative distribution function of the χ^2-distribution with k degrees of freedom and $g_k(x)$ is its probability density function (PDF). Large values of χ^2 correspond to small p-values. If the model is correctly specified, p is uniformly distributed in the unit interval. A very small p-value indicates a misspecification of the model or of the covariance matrix $\boldsymbol{V}$, or both.

3.2.2 Nonlinear Regression Models

The linear regression model in Eq. (3.14) can be generalized to a nonlinear model:

$$\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{p}) + \boldsymbol{\varepsilon}, \quad \mathsf{E}[\boldsymbol{\varepsilon}] = \boldsymbol{0}, \quad \mathsf{Var}[\boldsymbol{\varepsilon}] = \boldsymbol{V} = \boldsymbol{G}^{-1}, \tag{3.21}$$

where $\boldsymbol{f}$ is a $(n \times 1)$-vector of smooth functions of m variables. LS estimation of $\boldsymbol{p}$ requires the minimization of the following objective function:

$$\mathcal{S}(\boldsymbol{p}) = \left[\boldsymbol{m} - \boldsymbol{f}(\boldsymbol{p})\right]^{\mathsf{T}} \boldsymbol{G} \left[\boldsymbol{m} - \boldsymbol{f}(\boldsymbol{p})\right]. \tag{3.22}$$

The function $\mathcal{S}(\boldsymbol{p})$ can be minimized with any of the methods discussed in Sect. 3.1. The one used most frequently is probably the Gauss-Newton method, based on the first-order Taylor expansion of $\boldsymbol{f}$ and resulting in the following iteration:

$$\tilde{\boldsymbol{p}}_{k+1} = \tilde{\boldsymbol{p}}_k + (\boldsymbol{F}_k^{\mathsf{T}} \boldsymbol{G} \boldsymbol{F}_k)^{-1} \boldsymbol{F}_k^{\mathsf{T}} \boldsymbol{G} \left[\boldsymbol{m} - \boldsymbol{f}(\tilde{\boldsymbol{p}}_k)\right], \quad \boldsymbol{F}_k = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\right|_{\tilde{\boldsymbol{p}}_k}. \tag{3.23}$$

At each step, the covariance matrix $\boldsymbol{C}_{k+1}$ of $\tilde{\boldsymbol{p}}_{k+1}$ is approximately given by:

$$\boldsymbol{C}_{k+1} = (\boldsymbol{F}_k^{\mathsf{T}} \boldsymbol{G} \boldsymbol{F}_k)^{-1}. \tag{3.24}$$

In general, the covariance matrix of the final estimate $\tilde{\boldsymbol{p}}$ can be approximated by the inverse of the Hessian of $\mathcal{S}(\boldsymbol{p})$ at $\tilde{\boldsymbol{p}}$. The final chi-square statistic χ^2 is given by:

$$\chi^2 = \left[\boldsymbol{m} - \boldsymbol{f}(\tilde{\boldsymbol{p}})\right]^{\mathsf{T}} \boldsymbol{G} \left[\boldsymbol{m} - \boldsymbol{f}(\tilde{\boldsymbol{p}})\right]. \tag{3.25}$$

In the case of Gaussian observation errors, the chi-square statistic is approximately χ^2-distributed, and its p-value is approximately uniformly distributed. The iteration is stopped when the chi-square statistic does not change significantly any more.

3.2.3 State Space Models

A dynamic or state space model describes the state of an object in space or time, such as a rocket or a charged particle [9]. The state usually changes continuously, but is assumed to be of interest only at discrete instances in the present context. These instances are labeled with indices from 0, the initial state, to n, the final state. The state at instance k is specified by the state vector $\boldsymbol{q}_k$. The spatial or temporal evolution of the state is described by the system equation, which is Eq. (3.26) in the linear case and Eq. (3.41) in the general case.

3.2.3.1 Linear State Space Models and the Kalman Filter

In the simplest case, the state at instant k is an affine function of the state at instant $k-1$ plus a random disturbance called the system or process noise, with known expectation and covariance matrix:

$$\boldsymbol{q}_k = \boldsymbol{F}_{k|k-1}\boldsymbol{q}_{k-1} + \boldsymbol{d}_k + \boldsymbol{\gamma}_k, \quad \mathsf{E}\left[\boldsymbol{\gamma}_k\right] = \boldsymbol{g}_k, \quad \mathsf{Var}\left[\boldsymbol{\gamma}_k\right] = \boldsymbol{Q}_k. \tag{3.26}$$

The process noise $\boldsymbol{\gamma}_k$ may affect only a subset of the state vector, in which case its covariance matrix $\boldsymbol{Q}_k$ is not of full rank.

In most cases, only some or even none of the components of the state vector can be observed directly. Instead, the observations are functions of the state plus an observation error. In the simplest case, this function is again affine:

$$\boldsymbol{m}_k = \boldsymbol{H}_k\boldsymbol{q}_k + \boldsymbol{c}_k + \boldsymbol{\varepsilon}_k, \quad \mathsf{E}\left[\boldsymbol{\varepsilon}_k\right] = \boldsymbol{0}, \quad \mathsf{Var}\left[\boldsymbol{\varepsilon}_k\right] = \boldsymbol{V}_k = \boldsymbol{G}_k{}^{-1}. \tag{3.27}$$

Process noise and observation errors are assumed to be independent.

Given observations $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n$, an initial state $\boldsymbol{q}_0$ and an initial state covariance matrix $\boldsymbol{C}_0$, all states can be estimated recursively by the Kalman filter [10]. Assume there is an estimated state vector $\tilde{\boldsymbol{q}}_{k-1}$ with its covariance matrix $\boldsymbol{C}_{k-1}$ at instant $k-1$. The estimated state vector at instant k is obtained by a prediction step followed by an update step. After the last update step, the full information contained in all observations can be propagated back to all previous states by the smoother. If both process noise and observation errors are normally distributed, the Kalman filter is the optimal filter; if not, it is still the best linear filter.

Prediction step The state vector and its covariance matrix are propagated to the next instance using the system equation Eq. (3.26) and linear error propagation:

$$\tilde{\boldsymbol{q}}_{k|k-1} = \boldsymbol{F}_{k|k-1}\tilde{\boldsymbol{q}}_{k-1} + \boldsymbol{d}_k + \boldsymbol{g}_k, \quad \boldsymbol{C}_{k|k-1} = \boldsymbol{F}_{k|k-1}\boldsymbol{C}_{k-1}\boldsymbol{F}_{k|k-1}^{\mathsf{T}} + \boldsymbol{Q}_k. \tag{3.28}$$

Update step The updated state vector is the weighted mean of the predicted state vector and the observation $\boldsymbol{m}_k$. There are two equivalent ways to compute the update. The first one uses the gain matrix $\boldsymbol{K}_k$:

$$\tilde{\boldsymbol{q}}_k = \tilde{\boldsymbol{q}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{m}_k - \boldsymbol{c}_k - \boldsymbol{H}_k\tilde{\boldsymbol{q}}_{k|k-1}), \quad \boldsymbol{C}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{C}_{k|k-1}, \tag{3.29}$$

$$\boldsymbol{K}_k = \boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^\mathsf{T}(\boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^\mathsf{T})^{-1}. \tag{3.30}$$

The second one is a multivariate weighted mean:

$$\tilde{\boldsymbol{q}}_k = \boldsymbol{C}_k\left[\boldsymbol{C}_{k|k-1}^{-1}\tilde{\boldsymbol{q}}_{k|k-1} + \boldsymbol{H}_k^\mathsf{T}\boldsymbol{G}_k(\boldsymbol{m}_k - \boldsymbol{c}_k)\right], \tag{3.31}$$

$$\boldsymbol{C}_k = (\boldsymbol{C}_{k|k-1}^{-1} + \boldsymbol{H}_k^\mathsf{T}\boldsymbol{G}_k\boldsymbol{H}_k)^{-1}. \tag{3.32}$$

The update step has an associated chi-square statistic χ_k^2, which can be computed from the predicted residual $\boldsymbol{r}_{k|k-1}$ or from the updated residual $\boldsymbol{r}_k$:

$$\boldsymbol{r}_{k|k-1} = \boldsymbol{m}_k - \boldsymbol{c}_k - \boldsymbol{H}_k\tilde{\boldsymbol{q}}_{k|k-1}, \ \boldsymbol{R}_{k|k-1} = \mathsf{Var}\left[\boldsymbol{r}_{k|k-1}\right] = \boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^\mathsf{T}, \tag{3.33}$$

$$\boldsymbol{r}_k = \boldsymbol{m}_k - \boldsymbol{c}_k - \boldsymbol{H}_k\tilde{\boldsymbol{q}}_k, \ \boldsymbol{R}_k = \mathsf{Var}\left[\boldsymbol{r}_k\right] = \boldsymbol{V}_k - \boldsymbol{H}_k\boldsymbol{C}_k\boldsymbol{H}_k^\mathsf{T}, \tag{3.34}$$

$$\chi_k^2 = \boldsymbol{r}_{k|k-1}^\mathsf{T}\boldsymbol{R}_{k|k-1}^{-1}\boldsymbol{r}_{k|k-1} = \boldsymbol{r}_k^\mathsf{T}\boldsymbol{R}_k^{-1}\boldsymbol{r}_k. \tag{3.35}$$

If the model is correctly specified and both process noise and observation errors are normally distributed, χ_k^2 is χ^2-distributed with a number of degrees of freedom equal to the dimension of $\boldsymbol{m}_k$. The total chi-square χ_{tot}^2 of the filter is obtained by summing χ_k^2 over all k.

Smoothing The smoother propagates the full information contained in the last estimate $\tilde{\boldsymbol{q}}_n$ back to all previous states. There are again two equivalent formulations. The first one uses the gain matrix $\boldsymbol{A}_k$ of the smoother:

$$\tilde{\boldsymbol{q}}_{k|n} = \tilde{\boldsymbol{q}}_k + \boldsymbol{A}_k(\tilde{\boldsymbol{q}}_{k+1|n} - \tilde{\boldsymbol{q}}_{k+1|k}), \quad \boldsymbol{A}_k = \boldsymbol{C}_k\boldsymbol{F}_{k+1}^\mathsf{T}\boldsymbol{C}_{k+1|k}^{-1}, \tag{3.36}$$

$$\boldsymbol{C}_{k|n} = \boldsymbol{C}_k - \boldsymbol{A}_k(\boldsymbol{C}_{k+1|k} - \boldsymbol{C}_{k+1|n})\boldsymbol{A}_k^\mathsf{T}. \tag{3.37}$$

This formulation is numerically unstable, as the difference of the two positive definite matrices in Eq. (3.37) can fail to be positive definite as well because of rounding errors. The second, numerically stable formulation realizes the smoother by running two filters, one forward and one backward, on the same sets of observations. The smoothed state is a weighted mean of the states of the two filters:

$$\tilde{\boldsymbol{q}}_{k|n} = \boldsymbol{C}_{k|n}\left[{\boldsymbol{C}_k}^{-1}\tilde{\boldsymbol{q}}_k + \left(\boldsymbol{C}^{\mathrm{b}}_{k|k+1}\right)^{-1}\tilde{\boldsymbol{q}}^{\mathrm{b}}_{k|k+1}\right],\quad \boldsymbol{C}^{-1}_{k|n} = \boldsymbol{C}^{-1}_k + \left(\boldsymbol{C}^{\mathrm{b}}_{k|k+1}\right)^{-1}, \tag{3.38}$$

where $\tilde{\boldsymbol{q}}^{\mathrm{b}}_{k|k+1}$ is the predicted state from the backward filter and $\boldsymbol{C}^{\mathrm{b}}_{k|k+1}$ its covariance matrix. Alternatively, the predicted state from the forward filter and the updated state from the backward filter can be combined. The smoother step has an associated chi-square statistic $\chi^2_{k|n}$, which can be computed from the smoothed residuals:

$$\boldsymbol{r}_{k|n} = \boldsymbol{m}_k - \boldsymbol{c}_k - \boldsymbol{H}_k\tilde{\boldsymbol{q}}_{k|n},\ \boldsymbol{R}_{k|n} = \mathsf{Var}\left[\boldsymbol{r}_{k|n}\right] = \boldsymbol{V}_k - \boldsymbol{H}_k\boldsymbol{C}_{k|n}\boldsymbol{H}^{\mathsf{T}}_k, \tag{3.39}$$

$$\chi^2_{k|n} = \boldsymbol{r}^{\mathsf{T}}_{k|n}\boldsymbol{R}^{-1}_{k|n}\boldsymbol{r}_{k|n}. \tag{3.40}$$

If the model is correctly specified and both process noise and observation errors are normally distributed, $\chi^2_{k|n}$ is χ^2-distributed with a number of degrees of freedom equal to the dimension of $\boldsymbol{m}_k$. As the smoothed state vectors are estimated from the same information, they are correlated. The prescription for computing their joint covariance matrix is given in [11].

The Kalman filter can also be implemented as an information filter and as a square-root filter [10].

3.2.3.2 Nonlinear State Space Models and the Extended Kalman Filter

In the applications of the Kalman filter to track fitting, see Chap. 6, the system equation is usually nonlinear; see Sect. 4.3 and Fig. 4.4. It has the following form:

$$\boldsymbol{q}_k = \boldsymbol{f}_{k|k-1}(\boldsymbol{q}_{k-1}) + \boldsymbol{\gamma}_k,\quad \mathsf{E}\left[\boldsymbol{\gamma}_k\right] = \boldsymbol{g}_k,\quad \mathsf{Var}\left[\boldsymbol{\gamma}_k\right] = \boldsymbol{Q}_k. \tag{3.41}$$

In the prediction step, the exact linear error propagation is replaced by an approximate linearized error propagation; see also Sect. 4.4 and Fig. 4.5.

$$\begin{aligned}\tilde{\boldsymbol{q}}_{k|k-1} &= \boldsymbol{f}_{k|k-1}(\tilde{\boldsymbol{q}}_{k-1}) + \boldsymbol{g}_k,\quad \boldsymbol{F}_{k|k-1} = \left.\frac{\partial \boldsymbol{f}_{k|k-1}}{\partial \boldsymbol{q}_{k-1}}\right|_{\tilde{\boldsymbol{q}}_{k-1}},\\ \boldsymbol{C}_{k|k-1} &= \boldsymbol{F}_{k|k-1}\boldsymbol{C}_{k-1}\boldsymbol{F}^{\mathsf{T}}_{k|k-1} + \boldsymbol{Q}_k.\end{aligned} \tag{3.42}$$

More rarely, also the measurement equation can be nonlinear:

$$\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{q}_k) + \boldsymbol{\varepsilon}_k,\quad \mathsf{E}\left[\boldsymbol{\varepsilon}_k\right] = \boldsymbol{0},\quad \mathsf{Var}\left[\boldsymbol{\varepsilon}_k\right] = \boldsymbol{V}_k = {\boldsymbol{G}_k}^{-1}. \tag{3.43}$$

The first order Taylor expansion of $\boldsymbol{h}_k$ at $\tilde{\boldsymbol{q}}_{k|k-1}$ gives:

$$\boldsymbol{h}_k(\boldsymbol{q}_k) \approx \boldsymbol{H}_k\boldsymbol{q}_k + \boldsymbol{c}_k,\quad \boldsymbol{c}_k = \boldsymbol{h}_k(\tilde{\boldsymbol{q}}_{k|k-1}) - \boldsymbol{H}_k\tilde{\boldsymbol{q}}_{k|k-1}. \tag{3.44}$$

Using this $\boldsymbol{H}_k$ and $\boldsymbol{c}_k$ in Eqs. (3.29) and (3.30) or Eqs. (3.31) and (3.32) gives the update equations for the nonlinear measurement equation. If required, the update can be iterated by re-expanding $\boldsymbol{h}_k$ at $\tilde{\boldsymbol{q}}_k$ and recomputing $\boldsymbol{H}_k$, $\boldsymbol{c}_k$ and $\boldsymbol{K}_k$. The smoother can be implemented via its gain matrix, Eqs. (3.36) and (3.37), or by the backward filter, Eq. (3.38).

3.3 Clustering

Clustering is the classification of objects into groups, such that similar objects end up in the same group. The similarity of objects can be summarized in a distance/similarity matrix containing the pair-wise distances/similarities of the objects to be clustered. A cluster is then a group with small distances/large similarities inside the group and large distances/small similarities to objects outside the group. After clustering, the resulting cluster structure should be validated by measuring the internal consistency of each cluster. For further information and examples, the reader is directed to the literature [12–14].

3.3.1 Hierarchical Clustering

Hierarchical clustering groups the objects with a sequence of nested partitions [12, Chapter 3]. If the sequence starts from single-object clusters and merges them to larger clusters, the clustering is called agglomerative; if the sequence starts from a single cluster containing all objects and splits it into successively smaller clusters, the clustering is called divisive. At any stage of the clustering, all clusters are pairwise disjoint. The number of clusters is not necessarily specified in advance, but can determined "on the fly". If in divisive clustering a cluster is considered to be valid, further splitting is not required, but also not forbidden. If in agglomerative clustering the merging of two clusters results in an invalid cluster, the merger is undone.

3.3.2 Partitional Clustering

Partitional clustering directly divides the objects into a predefined number K of clusters, usually by optimizing an objective function that describes the global quality of the cluster structure [12, Chapter 4]. Partitional clustering can be repeated for several values of K in order to find the optimal cluster structure. In the fuzzy variant of partitional clustering, an object can belong to more than one cluster with a certain degree of membership [15].

3.3.3 Model-Based Clustering

In model-based clustering, the objects are assumed to be drawn from a mixture distribution with two or more components. Each component is described by a PDF and has an associated probability or "weight" in the mixture [16]. The parameters of the mixture are usually estimated by the Expectation-Maximization (EM) algorithm [17–19]; see Sect. 7.2.2. A by-product of the EM algorithm is the posterior probability π_{ik} of object i belonging to cluster k, for all objects and all clusters. The result is again a fuzzy clustering.

References

1. MathWorks, MATLAB Optimization Toolbox. https://www.mathworks.com/products/optimization.html
2. SciPy v1.4.1 Reference Guide, https://tinyurl.com/SciPy-optimize
3. P. Wolfe, SIAM Rev. **11**(2), 226 (1969)
4. P. Pedregal, *Introduction to Optimization* (Springer, New York, 2006)
5. L. Foulds, *Optimization Techniques* (Springer, New York, 1981)
6. M. Bazaraa, H. Sherali, C. Shetty, *Nonlinear Programming: Theory and Algorithms* (Wiley, Hoboken, 2005)
7. J.A. Nelder, R. Mead, Comput. J. **7**, 308 (1965)
8. N.H.H. Bingham, J.M. Fry, *Regression: Linear Models in Statistics* (Springer, London, 2010)
9. R. Prado, M. West, *Time Series* (CRC Press, Boca Raton, 2010)
10. B.D.O. Anderson, J.B. Moore, *Optimal Filtering* (Dover Publications, Mineola, 2005)
11. W. Hulsbergen, Nucl. Instrum. Methods Phys. Res. A **600**(2), 471 (2009)
12. R. Xu, D. Wunsch, *Clustering* (Wiley–IEEE Press, Hoboken, 2009)
13. L. Kaufman, P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley, Hoboken, 2005)
14. E.A. Patrick, *Fundamentals of Pattern Recognition* (Prentice–Hall, Upper Saddle River, 1972)
15. J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* (Springer, Boston, 1981)
16. V. Melnykov, R. Maitra, Stat. Surv. **4**, 80 (2010). https://projecteuclid.org/euclid.ssu/1272547280
17. A.P. Dempster, N.M. Laird, D.B. Rubin, J. R. Stat. Soc. Ser. B **39**(1), 1 (1977)
18. J.A. Bilmes, A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report TR-97-021, International Computer Science Institute (1998). http://www.icsi.berkeley.edu/ftp/global/pub/techreports/1997/tr-97-021.pdf
19. S. Borman, The Expectation Maximization Algorithm—A short tutorial. Technical Report, University of Utah (2004). https://tinyurl.com/BormanEMTutorial

Part II
Track Reconstruction

Chapter 4
Track Models

Abstract The chapter shows how the equations of motion for charged particles in a homogeneous or inhomogeneous magnetic field are solved. Various types of parametrizations are presented, and formulas for track propagation and error propagation are derived. As the effects of the detector material on the trajectory have to be taken into account, the statistical properties of multiple Coulomb scattering, energy loss by ionization, and energy loss by bremsstrahlung are discussed; then it is shown how the effects can be treated in the track reconstruction. As multiple scattering in thin layers and energy loss by bremsstrahlung have distinctive non-Gaussian features, an approximation by normal mixtures is presented.

4.1 The Equations of Motion

Consider a charged particle with mass m and charge $Q = qe$, where e is the elementary charge and q is an integer, usually $q = \pm 1$. Its trajectory or position $\boldsymbol{r}(t) = (x(t), y(t), z(t))^{\mathsf{T}}$ in a magnetic field $\boldsymbol{B}(\boldsymbol{r})$, as a function of time, is determined by the equations of motion given by the Lorentz force $\boldsymbol{F} \propto q\,\boldsymbol{v} \times \boldsymbol{B}$, where $\boldsymbol{v} = \mathrm{d}\boldsymbol{r}/\mathrm{d}t$ is the velocity of the particle. In vacuum, Newton's second law reads [1]:

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = k q\, \boldsymbol{v}(t) \times \boldsymbol{B}(\boldsymbol{r}(t)), \tag{4.1}$$

where $\boldsymbol{p} = \gamma m \boldsymbol{v}$ is the momentum vector of the particle, $\gamma = (1 - \boldsymbol{v}^2/c^2)^{-1/2}$ is the Lorentz factor, and k is a unit-dependent proportionality factor. If the units are GeV/c for $\boldsymbol{p}$, meter for $\boldsymbol{r}$, and Tesla for $\boldsymbol{B}$, then $k = 0.29979\,\mathrm{GeV}/c\,\mathrm{T}^{-1}\,\mathrm{m}^{-1}$. The trajectory is uniquely defined by the initial conditions, i.e., the six degrees of freedom specified for instance by the initial position and the initial momentum. If these are tied to a reference surface, five degrees of freedom are necessary and sufficient. Geometrical quantities other than position and velocity can also be used

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_4

to specify the initial conditions. The collection $\boldsymbol{q} = (q_1, \ldots, q_m)$ of these quantities is called the initial track parameter vector or the initial state vector.

Equation (4.1) can be written in terms of the path length $s(t)$ along the trajectory instead of t, giving [1]:

$$\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}s^2} = k\psi \cdot \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}s} \times \boldsymbol{B}(\boldsymbol{r}(s)) = \boldsymbol{\Gamma}\left(s, \boldsymbol{r}(s), \dot{\boldsymbol{r}}(s)\right), \tag{4.2}$$

where $\dot{\boldsymbol{r}}(s) \equiv \mathrm{d}\boldsymbol{r}/\mathrm{d}s$, $\psi = q/p$ and $p = |\boldsymbol{p}|$. In simple cases, this equation has closed-form solutions. In a homogeneous magnetic field, the solution is a helix; it reduces to a straight line in the limit of $\boldsymbol{B} \equiv \boldsymbol{0}$. In the general case of an inhomogeneous magnetic field, one has to resort to numerical methods such as Runge–Kutta integration of the equations of motion (Sect. 4.3.2.1), parametrization by polynomials or splines [1], or other approximations [2]; see Sect. 4.3.2.2.

Equation (4.2) can be expressed in terms of other independent variables. For example, if the equations of motion are integrated in a cylindrical detector geometry, the radius R is a natural integration variable. In a planar detector geometry, the position coordinate z could be the variable of choice [2, 3].

4.2 Track Parametrization

Different detector geometries often lead to different choices of the track parameters. However, the parametrization of the trajectory should comply with some basic requirements: the parameters should be continuous with respect to small changes of the trajectory; the choice of track parameters should facilitate the local expansion of the track model into a linear function; and the stochastic uncertainties of the estimated parameters should follow a Gaussian distribution as closely as possible. In order to fulfill, for instance, the continuity requirement, curvature should be used rather than radius of curvature, and inverse (transverse) momentum rather than (transverse) momentum.

In a barrel-type detector system typical for the central part of collider experiments, a natural reference surface of the track parameters is a cylinder with radius R, centered around the global z-axis, which usually coincides with the beam line. The track parameters are, in this case, defined at the point of intersection P between the track and the reference cylinder. In such a system, one possible choice of track parametrization is the following:

$$q_1 = q/p_\mathrm{T},\ q_2 = \phi,\ q_3 = \tan\lambda,\ q_4 = R\Phi,\ q_5 = z, \tag{4.3}$$

where $p_\mathrm{T} = p\cos\lambda$ is the transverse momentum, ϕ is the azimuth angle of the tangent of the track at P, λ is the dip angle (complement of the polar angle) of the tangent at P, and $R\Phi$ and z are the cylindrical coordinates of P in the global coordinate system, see Fig. 4.1.

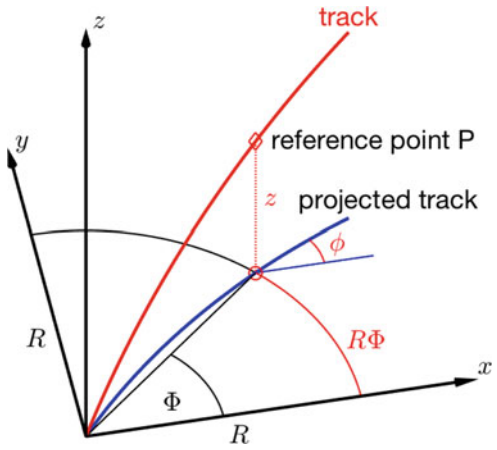

Fig. 4.1 Track parametrization according to Eq. (4.3). The parameter $\tan\lambda$ is the slope of the tangent at the reference point with respect to the (x, y)-plane

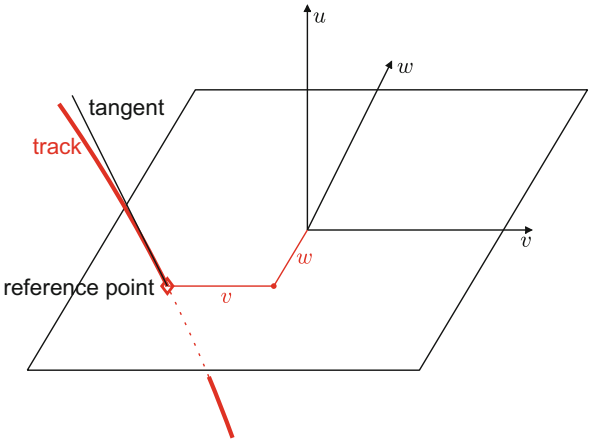

Fig. 4.2 Track parametrization according to Eq. (4.4). The parameters $\mathrm{d}v/\mathrm{d}u$ and $\mathrm{d}w/\mathrm{d}u$ are the direction tangents of the track at the reference point with respect to the u-axis

In a detector system based on planar detector elements, the natural reference surface is a plane. Such a surface is uniquely determined by a normal vector of the plane and the position of a reference point inside the plane. A local coordinate system is defined such that the u-axis is parallel to the normal vector and the v- and w-axes are inside the plane. A natural choice of track parameters is now

$$q_1 = \psi,\ q_2 = \mathrm{d}v/\mathrm{d}u,\ q_3 = \mathrm{d}w/\mathrm{d}u,\ q_4 = v,\ q_5 = w, \tag{4.4}$$

where $\psi = q/p$, $\mathrm{d}v/\mathrm{d}u$ is the tangent of the angle between the projection of the track tangent into the (u, v)-plane and the u-axis, $\mathrm{d}w/\mathrm{d}u$ is the tangent of the angle between the projection of the track tangent into the (u, w)-plane and the u-axis, and v and w are the local coordinates of the intersection point of the track with the plane; see Fig. 4.2. The quantities $\mathrm{d}v/\mathrm{d}u$ and $\mathrm{d}w/\mathrm{d}u$ are also called direction tangents, as the tangent vector to the track is proportional to the vector $(\mathrm{d}v/\mathrm{d}u, \mathrm{d}w/\mathrm{d}u, 1)^{\mathsf{T}}$.

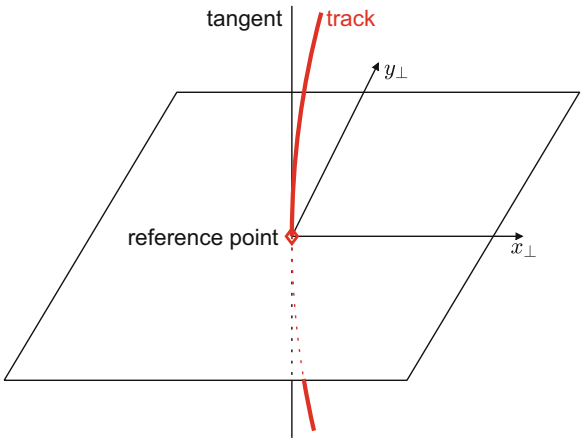

Fig. 4.3 Track parametrization according to Eq. (4.5). The parameters $x_\perp$ and $y_\perp$ are the distances from the reference point in the plane perpendicular to the track. The direction of the tangent is measured in global polar coordinates

Another planar reference system is the curvilinear frame, which is useful for transporting uncertainties of track parameters; see Sect. 4.4. It is a hybrid local/global reference frame. The curvilinear plane is always orthogonal to the direction of the track with the parametrization:

$$q_1 = \psi,\ q_2 = \phi,\ q_3 = \lambda,\ q_4 = x_\perp,\ q_5 = y_\perp, \tag{4.5}$$

where $x_\perp$ and $y_\perp$ are orthogonal position coordinates inside the plane, see Fig. 4.3. The $x_\perp$-axis is parallel to the global (x, y)-plane. The azimuth and dip angles ϕ and λ are defined at the point of intersection of the track with the curvilinear plane, but their values are measured in the global Cartesian coordinate system. The tangent vector is thus proportional to $(\cos\lambda\cos\phi, \cos\lambda\sin\phi, \sin\lambda)^\mathsf{T}$.

In addition to the above-mentioned, surface-based frames, a global, Cartesian coordinate frame is also frequently used. In this frame, the track parameters are the position vector $\boldsymbol{r}$ and the momentum vector $\boldsymbol{p}$ at that point. Since these are not tied to a surface, six parameters are needed in order to uniquely specify the state of the track. The rank of their covariance matrix is at most five.

4.3 Track Propagation

The track model, given by the solution of the equations of motion, describes the functional dependence of the state vector $\boldsymbol{q}_j$ at a surface j on the state vector $\boldsymbol{q}_i$ at a different surface i:

$$\boldsymbol{q}_j = \boldsymbol{f}_{j|i}\left(\boldsymbol{q}_i\right). \tag{4.6}$$

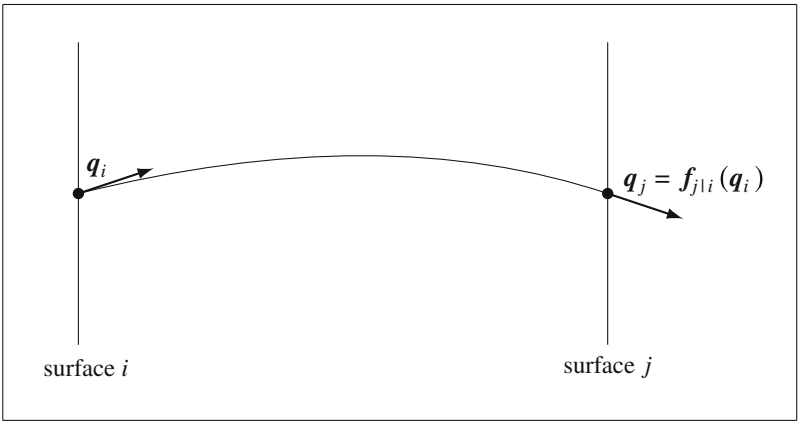

Fig. 4.4 Track propagator from surface i to surface j

The function $f_{j|i}$ is called the track propagator from surface i to surface j, see Fig. 4.4. When closed-form solutions of the equations of motion exist, e.g., in the two situations of vanishing magnetic field and homogeneous magnetic field, the track propagator can be written as an explicit function of the path length. In the straight-line solution of the equations of motion in a vanishing magnetic field, it is easy to also derive an analytical formula for the path length between two surfaces. For the helical solution in a homogeneous magnetic field, however, such an analytical formula exists only for propagation to cylinders with symmetry axis parallel to the field direction or to planes orthogonal to the field direction. Otherwise, a Newton iteration or a parabolic approximation has to be used to find the path length.

4.3.1 Homogeneous Magnetic Fields

The helical track propagator takes the solution to Eq. (4.2) as a starting point. The solution [4] can be written as:

$$\boldsymbol{r}(s) = \boldsymbol{r}_0 + \frac{\delta}{K}(\theta - \sin\theta)\cdot\boldsymbol{h} + \frac{\sin\theta}{K}\cdot\boldsymbol{t}_0 + \frac{\alpha}{K}(1-\cos\theta)\cdot\boldsymbol{n}_0, \tag{4.7}$$

where $\boldsymbol{r}(s)$ is the position vector of the point on the helix at path length s from the reference point $\boldsymbol{r}_0$ (at $s = 0$), $\boldsymbol{h} = \boldsymbol{B}/|\boldsymbol{B}|$ is the normalized magnetic field vector, $\boldsymbol{t} = \boldsymbol{p}/p$ is the unit tangent vector to the track, $\boldsymbol{n} = (\boldsymbol{h}\times\boldsymbol{t})/\alpha$ with $\alpha = |\boldsymbol{h}\times\boldsymbol{t}|$, $\delta = \boldsymbol{h}\cdot\boldsymbol{t}$, $K = -k\psi|\boldsymbol{B}|$, and $\theta = Ks$. In the following, the subscript "0" indicates quantities defined at the initial point $s = 0$. Any point along the trajectory can be specified by a corresponding value of s. The equation of the unit tangent vector $\boldsymbol{t}$ is found by differentiating Eq. (4.7) with respect to s,

$$\boldsymbol{t}(s) = \frac{\mathrm{d}\boldsymbol{r}(s)}{\mathrm{d}s} = \delta\,(1 - \cos\theta)\cdot\boldsymbol{h} + \cos\theta\cdot\boldsymbol{t}_0 + \alpha\sin\theta\cdot\boldsymbol{n}_0. \tag{4.8}$$

For a given value of s, any desired set of track parameters can be calculated from Eq. (4.7) (positions) and Eq. (4.8) (directions). In the helical track model, the momentum p is constant and therefore has the same value for all s.

4.3.2 Inhomogeneous Magnetic Fields

In an inhomogeneous magnetic field, the equations of motion have no exact closed-form solutions, and one has to resort to numerical, approximate solutions.

4.3.2.1 Runge–Kutta Methods

Runge–Kutta methods are iterative algorithms for the approximate numerical solutions of ordinary differential equations, given initial values. Among Runge–Kutta methods, the Runge–Kutta–Nyström algorithm is specifically designed for second-order equations such as Eq. (4.2). In the fourth-order version a step of length h, starting at $s = s_n$, is computed by [1]:

$$\begin{aligned}\boldsymbol{r}_{n+1} &= \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n + h^2(\boldsymbol{k}_1 + \boldsymbol{k}_2 + \boldsymbol{k}_3)/6,\\ \dot{\boldsymbol{r}}_{n+1} &= \dot{\boldsymbol{r}}_n + h(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)/6,\end{aligned} \tag{4.9}$$

with the intermediate stages $\boldsymbol{k}$ defined by

$$\begin{aligned}\boldsymbol{k}_1 &= \boldsymbol{\Gamma}(s_n, \boldsymbol{r}_n, \dot{\boldsymbol{r}}_n),\\ \boldsymbol{k}_2 &= \boldsymbol{\Gamma}(s_n + h/2, \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n/2 + h^2\boldsymbol{k}_1/8, \dot{\boldsymbol{r}}_n + h\boldsymbol{k}_1/2),\\ \boldsymbol{k}_3 &= \boldsymbol{\Gamma}(s_n + h/2, \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n/2 + h^2\boldsymbol{k}_1/8, \dot{\boldsymbol{r}}_n + h\boldsymbol{k}_2/2),\\ \boldsymbol{k}_4 &= \boldsymbol{\Gamma}(s_n + h, \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n + h^2\boldsymbol{k}_3/2, \dot{\boldsymbol{r}}_n + h\boldsymbol{k}_3),\end{aligned} \tag{4.10}$$

where $\boldsymbol{r}_n$ is the position of the particle at $s = s_n$, $\dot{\boldsymbol{r}}_n$ is the unit tangent vector, and $\boldsymbol{\Gamma}$ is defined in Eq. (4.2). The magnetic field needs to be looked up three times per step, at the positions $\boldsymbol{r}_n$, $\boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n/2 + h^2\boldsymbol{k}_1/8$, and $\boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n + h^2\boldsymbol{k}_3/2$. If the field at the final position $\boldsymbol{r}_{n+1}$, which is the starting position of the next step, is approximated by the field used for $\boldsymbol{k}_4$, only two lookups are required per step.

If integration variables other than the path length s are used, for instance the radius R or the position coordinate z, the integration equations Eq. (4.9) are very similar. The difference is that the step length h must be expressed in terms of R or z, and that the function $\boldsymbol{\Gamma}$ has a different form when expressed in other variables [3].

If the field is (almost) homogeneous, as for example in a solenoid, the step size h can be chosen to be constant; otherwise a variable step size, taking into account the local inhomogeneity of the field, is more efficient. Determining the variable step size along the propagation is done by a step-size selection algorithm. The essence of such an algorithm is to assess the local error ϵ of the Runge–Kutta step and compare it to a user-defined error tolerance τ. The so-called embedded Runge–Kutta pairs, originally invented by Fehlberg [5], provide a measure of the local error in an elegant way by producing solutions of different orders during the same step with little extra computational cost. The higher-order solution is denoted $\boldsymbol{r}_{n+1}$, whereas the lower-order solution is denoted $\hat{\boldsymbol{r}}_{n+1}$. The difference

$$\epsilon = |\boldsymbol{r}_{n+1} - \hat{\boldsymbol{r}}_{n+1}| \tag{4.11}$$

between these solutions constitutes a measure of the error of the step. A popular algorithm for the step size h_{n+1} of step $n+1$ is [6]

$$h_{n+1} = h_n \left(\frac{\tau}{\epsilon}\right)^{1/(q+1)}, \tag{4.12}$$

where h_n is the size of step n and q is the order of the lower-order solution. This algorithm will effectively shorten the step size if the local error is larger than the tolerance and lengthen it if the local error is smaller than the tolerance, forcing the local error to oscillate around the tolerance.

The original version of the Runge–Kutta–Nyström algorithm is not of the embedded type and therefore contains no direct recipe for estimating the local error. It was realized by the authors of [7], however, that a fourth-order derivative of $\boldsymbol{r}$ can be formed by a combination of the various stages calculated along the step. This derivative is implicitly the difference between a fourth-order solution and a third-order solution and can therefore be used as a measure of the local error ϵ:

$$\epsilon = h^2 \cdot |\boldsymbol{k}_1 - \boldsymbol{k}_2 - \boldsymbol{k}_3 + \boldsymbol{k}_4|. \tag{4.13}$$

This error measure can be used for step-size selection according to Eq. (4.12) and constitutes an adaptive version of the Runge–Kutta–Nyström algorithm when used alongside the integration steps in Eq. (4.9).

4.3.2.2 Approximate Analytical Formula

An approximate analytical formula for track extrapolation in an inhomogeneous field is described in [2]. The magnetic field $\boldsymbol{B}(z)$ is assumed to depend only on the z-coordinate. The particle is assumed to move along the z-axis, and the track parameters are x, y, t_x, t_y, ψ, where t_x, t_y are the direction tangents. In this parametrization, the equations of motion read:

$$
\begin{aligned}
x' &= t_x, \\
y' &= t_y, \\
t_x' &= h \cdot [t_x t_y B_1 - (1 + t_x^2) B_1 + t_y B_3] = \boldsymbol{a}(z) \cdot \boldsymbol{B}(z), \\
t_y' &= h \cdot [(1 + t_y^2) B_1 - t_x t_y B_2 - t_x B_3] = \boldsymbol{b}(z) \cdot \boldsymbol{B}(z), \\
\psi' &= 0,
\end{aligned}
\tag{4.14}
$$

where $h = k\psi \left(1 + t_x^2 + t_y^2\right)^{1/2}$ and the prime denotes differentiation with respect to z. The aim is to find formulas for the extrapolation of (t_x, t_y) from z_0 to z_e; the extrapolation of x and y can then be performed by integration of the direction tangents.

Let $T(z) = T(t_x(z), t_y(z))$ be a function of t_x and t_z. Then

$$
T' = \frac{\partial T}{\partial t_x} t_x' + \frac{\partial T}{\partial t_y} t_y' = \left(\frac{\partial T}{\partial t_x} \boldsymbol{a} + \frac{\partial T}{\partial t_y} \boldsymbol{b} \right) \cdot \boldsymbol{B} = \sum_{i_1=1}^{3} T_{i_1} B_{i_1}. \tag{4.15}
$$

The derivatives of the functions T_{i_1} can be represented in a similar way:

$$
T_{i_1}' = \frac{\partial T_{i_1}'}{\partial t_x} t_x' + \frac{\partial T_{i_1}'}{\partial t_y} t_y' = \left(\frac{\partial T_{i_1}'}{\partial t_x} \boldsymbol{a} + \frac{\partial T_{i_1}'}{\partial t_y} \boldsymbol{b} \right) \cdot \boldsymbol{B} = \sum_{i_2=1}^{3} T_{i_1 i_2} B_{i_2}. \tag{4.16}
$$

When this process is continued, the following functions can be defined recursively:

$$
T_{i_1 \ldots i_{k-1}}' = \sum_{i_k=1}^{3} T_{i_1 \ldots i_k} B_{i_k}, \tag{4.17}
$$

$$
T_{i_1 \ldots i_k} = \frac{\partial T_{i_1 \ldots i_{k-1}}}{\partial t_x} a_{i_k} + \frac{\partial T_{i_1 \ldots i_{k-1}}}{\partial t_y} b_{i_k}. \tag{4.18}
$$

The exact relation

$$
T(z_\mathrm{e}) = T(z_0) + \int_{z_0}^{z_e} T'(z) \mathrm{d}z \tag{4.19}
$$

can then be expanded into:

$$\begin{aligned} T(z_\mathrm{e}) &= T(z_0) + \sum_{i_1=1}^{3} \int_{z_0}^{z_\mathrm{e}} T_{i_1}(z_1) B_{i_1}(z_1) \mathrm{d}z_1 = \\ &= T(z_0) + \sum_{i_1=1}^{3} \int_{z_0}^{z_\mathrm{e}} \left(T_{i_1}(z_0) + \int_{z_0}^{z_1} T'_{i_1}(z_2) \mathrm{d}z_2 \right) B_{i_1}(z_1) \mathrm{d}z_1 = \\ &= T(z_0) + \sum_{i_1=1}^{3} T_{i_1}(z_0) \int_{z_0}^{z_\mathrm{e}} B_{i_1}(z_1) \mathrm{d}z_1 + \\ &\quad + \sum_{i_1=1}^{3} \int_{z_0}^{z_\mathrm{e}} B_{i_1} \int_{z_0}^{z_1} \sum_{i_2=1}^{3} T_{i_1 i_2}(z_2) \mathrm{d}z_2 \mathrm{d}z_1 \\ &= \ldots \end{aligned} \tag{4.20}$$

If the expansion is terminated after n steps, the compact form of Eq. (4.20) reads:

$$\begin{aligned} T(z_\mathrm{e}) \approx T(z_0) + \sum_{k=1}^{n} \sum_{i_1 \ldots i_k} T_{i_1 \ldots i_k}(z_0) \times \left(\int_{z_0}^{z_\mathrm{e}} B_{i_1}(z_1) \ldots \int_{z_0}^{z_{k-1}} B_{i_k}(z_k) \mathrm{d}z_k \ldots \mathrm{d}z_1 \right) + \\ + O\left(\frac{k B \psi (z_\mathrm{e} - z_0)^{n+1}}{(n+1)!} \right) \end{aligned} \tag{4.21}$$

The integral over the field components is taken along an approximate trajectory. Setting $T = t_x$ and $T = t_y$ gives the desired extrapolation formulas. For a comparison with a Runge–Kutta solver, see [2].

4.4 Error Propagation

The task of transporting the covariance matrix of the track parameters is essential for any track reconstruction algorithm, either from one set of track parameters to another in the same reference surface, or during track propagation from one surface to another.

This so-called error propagation is straightforward when the transformed track parameter vector is a strictly linear function of the initial track parameter vector. The track propagator in Eq. (4.6), however, is in general a non-linear function, and exact error propagation is not feasible. The common solution is approximate linearized error propagation; see Sect. 3.2.3.2 and Fig. 4.5. The derivatives of the track propagator $\boldsymbol{f}_{j|i}$ are collected in the Jacobian matrix $\boldsymbol{F}_{j|i}$:

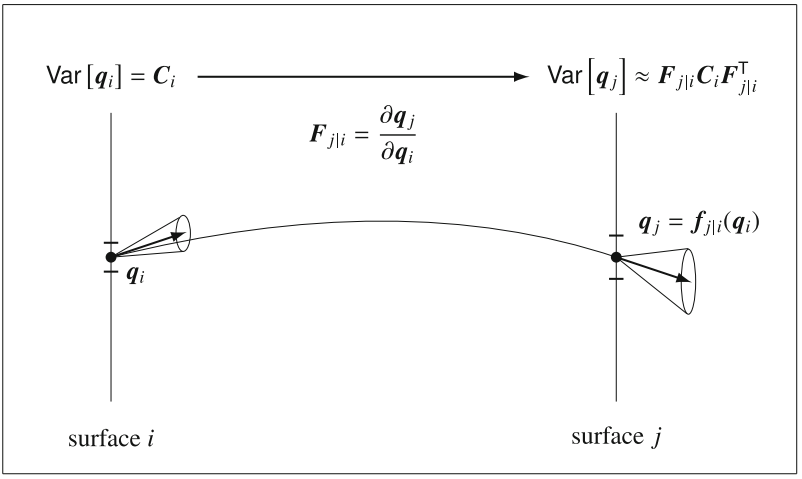

Fig. 4.5 Error propagation from surface i to surface j. The error bars around the track positions and the cones around the direction vectors symbolize the growing uncertainty of the track parameters

$$\boldsymbol{F}_{j|i} = \frac{\partial \boldsymbol{q}_j}{\partial \boldsymbol{q}_i}. \tag{4.22}$$

The covariance matrix $\boldsymbol{C}_i$ of the track parameters at surface i is transported to surface j according to:

$$\boldsymbol{C}_j \approx \boldsymbol{F}_{j|i}\boldsymbol{C}_i\boldsymbol{F}_{j|i}^{\mathsf{T}}. \tag{4.23}$$

A general method for computing the Jacobians is numerical differentiation [8, Section 5.7], by propagating a reference track and five other, nearby tracks from surface i to surface j. Consider a reference track with parameter vector $\boldsymbol{q}_i$ at surface i. A small variation of component l ($l = 1, \ldots, 5$) in $\boldsymbol{q}_i$ is introduced by adding to $\boldsymbol{q}_i$ the vector $\boldsymbol{\Delta}_l = (\delta_{1l}h_l, \ldots, \delta_{5l}h_l)^{\mathsf{T}}$. The corresponding change in parameter k ($k = 1, \ldots, 5$) at surface j is

$$\Delta_{kl} = f_k(\boldsymbol{q}_i + \boldsymbol{\Delta}_l) - f_k(\boldsymbol{q}_i), \tag{4.24}$$

where f_k is component k of the track propagator $\boldsymbol{f}_{j|i}$. The elements $(\boldsymbol{F}_{j|i})_{kl}$ of the numerical Jacobian matrix $\boldsymbol{F}_{j|i}$ are then obtained by evaluating

$$(\boldsymbol{F}_{j|i})_{kl} = \frac{\Delta_{kl}}{h_l}. \tag{4.25}$$

This procedure works for all track propagators, irrespective of whether or not they are in closed form.

4.4.1 Homogeneous Magnetic Fields

The exposition in this section closely follows the treatment in [4]. In a homogeneous magnetic field, it is possible to obtain analytical formulas for the Jacobians defined in Eq. (4.22). The problem of calculating transport Jacobians from one plane of arbitrary orientation to another is naturally decomposed into three separate parts because the error propagation from one spatial location to another is performed most easily in a coordinate frame which moves along the track, i.e., the curvilinear frame introduced in Sect. 4.2. Therefore, the natural decomposition is first a transformation from a local coordinate system to the curvilinear frame at the initial surface, then a transport within the curvilinear frame to the destination surface, and finally a transformation from the curvilinear frame to a local frame at the destination surface. The total Jacobian is the matrix product of the three intermediate Jacobians.

The starting point of calculating the transport Jacobians are differentials relating variations of position, direction, and momentum at the initial point ($s = 0$) to variations of the same quantities at any other point along the helix. These differentials are given by

$$\mathrm{d}\boldsymbol{r} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{r}_0} \cdot \mathrm{d}\boldsymbol{r}_0 + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{t}_0} \cdot \mathrm{d}\boldsymbol{t}_0 + \frac{\partial \boldsymbol{r}}{\partial \psi_0} \cdot \mathrm{d}\psi_0 + \frac{\partial \boldsymbol{r}}{\partial s} \cdot \mathrm{d}s, \tag{4.26}$$

$$\mathrm{d}\boldsymbol{t} = \frac{\partial \boldsymbol{t}}{\partial \boldsymbol{t}_0} \cdot \mathrm{d}\boldsymbol{t}_0 + \frac{\partial \boldsymbol{t}}{\partial \psi_0} \cdot \mathrm{d}\psi_0 + \frac{\partial \boldsymbol{t}}{\partial s} \cdot \mathrm{d}s, \tag{4.27}$$

where $\mathrm{d}\psi_0$ is the variation of the signed inverse momentum at the initial point and $\mathrm{d}s$ is the change in path length of the helix due to the variations at the initial point. An illustration of this effect is shown in Fig. 4.6.

The partial derivatives are obtained by direct differentiation of Eqs. (4.7) and (4.8). The results are:

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{r}_0} \cdot \mathrm{d}\boldsymbol{r}_0 = \mathrm{d}\boldsymbol{r}_0, \tag{4.28}$$

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{t}_0} \cdot \mathrm{d}\boldsymbol{t}_0 = \frac{\theta - \sin\theta}{K} \cdot (\boldsymbol{h} \cdot \mathrm{d}\boldsymbol{t}_0) \cdot \boldsymbol{h} + \frac{\sin\theta}{K} \cdot \mathrm{d}\boldsymbol{t}_0 + \frac{1 - \cos\theta}{K} \cdot (\boldsymbol{h} \times \mathrm{d}\boldsymbol{t}_0)\,, \tag{4.29}$$

$$\frac{\partial \boldsymbol{r}}{\partial \psi_0} \cdot \mathrm{d}\psi_0 = \frac{1}{\psi} [s \cdot \boldsymbol{t} + \boldsymbol{r}_0 - \boldsymbol{r}] \cdot \mathrm{d}\psi_0, \tag{4.30}$$

$$\frac{\partial \boldsymbol{r}}{\partial s} \cdot \mathrm{d}s = \boldsymbol{t} \cdot \mathrm{d}s, \tag{4.31}$$

$$\frac{\partial \boldsymbol{t}}{\partial \boldsymbol{t}_0} \cdot \mathrm{d}\boldsymbol{t}_0 = \cos\theta \cdot \mathrm{d}\boldsymbol{t}_0 + (1 - \cos\theta) \cdot (\boldsymbol{h} \cdot \mathrm{d}\boldsymbol{t}_0) \cdot \boldsymbol{h} + \sin\theta \cdot (\boldsymbol{h} \times \mathrm{d}\boldsymbol{t}_0)\,, \tag{4.32}$$

$$\frac{\partial \boldsymbol{t}}{\partial \psi_0} \cdot \mathrm{d}\psi_0 = \frac{\alpha K s}{\psi} \cdot \boldsymbol{n} \cdot \mathrm{d}\psi_0, \tag{4.33}$$

$$\frac{\partial \boldsymbol{t}}{\partial s} \cdot \mathrm{d}s = \alpha K \cdot \boldsymbol{n} \cdot \mathrm{d}s. \tag{4.34}$$

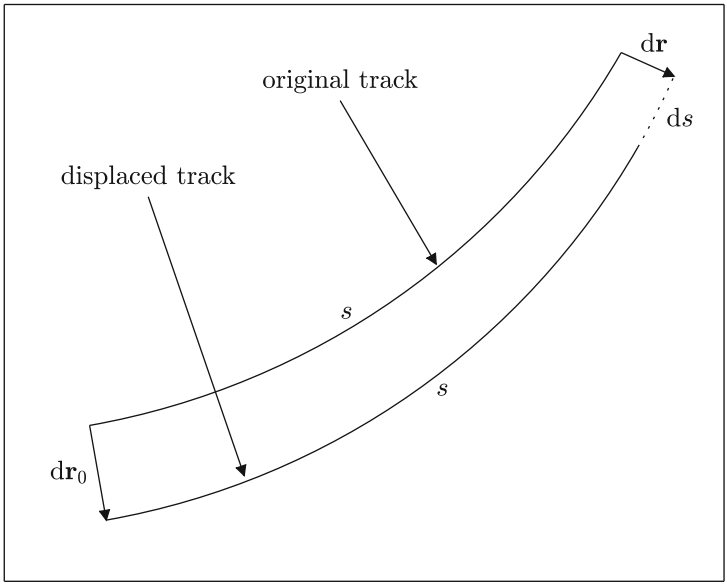

Fig. 4.6 A track and the displaced track due to a variation $\mathrm{d}\boldsymbol{r}_0$ are shown. In the error propagation, the change $\mathrm{d}s$ of the path length has to be taken into account. In this specific case, $\mathrm{d}\boldsymbol{r}$ is understood to be perpendicular to the track (see also Eq. (4.41))

4.4.1.1 Transformation from One Curvilinear Frame to Another

The curvilinear frame is uniquely defined at each point along the track by three orthogonal unit vectors $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{t}$, defining a coordinate system $(x_\perp, y_\perp, z_\perp)$. The vector $\boldsymbol{t}$ has been defined above as the unit vector parallel to the track, and pointing in the particle direction. The two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are defined by

$$\boldsymbol{u} = \frac{\boldsymbol{z} \times \boldsymbol{t}}{|\boldsymbol{z} \times \boldsymbol{t}|}, \quad \boldsymbol{v} = \boldsymbol{t} \times \boldsymbol{u}, \tag{4.35}$$

where $\boldsymbol{z}$ is the unit vector pointing in the direction of the global z-axis. This means that the $z_\perp$-axis is pointing along the particle direction, the $x_\perp$-axis is parallel to the global (x, y)-plane, while the $y_\perp$-axis is given by the requirement that the three axes should form a Cartesian, right-handed coordinate system. The relations between the momentum components $\left(p_x, p_y, p_z\right)$ in the global Cartesian frame and the angles are:

$$\begin{aligned} p_x &= p \cos\lambda \cos\phi, \\ p_y &= p \cos\lambda \sin\phi, \\ p_z &= p \sin\lambda. \end{aligned} \tag{4.36}$$

The Jacobian of the transformation from a curvilinear frame $(\psi, \phi, \lambda, x_\perp, y_\perp)$ at $s_0 = 0$ to the same set of parameters at path length s is then derived by forming the differentials $\mathrm{d}\boldsymbol{r}$ and $\mathrm{d}\boldsymbol{t}$, introducing the specific constraints given by the curvilinear frames,

$$\mathrm{d}\boldsymbol{r}_0 = \boldsymbol{u}_0 \cdot \mathrm{d}x_{\perp 0} + \boldsymbol{v}_0 \cdot \mathrm{d}y_{\perp 0}, \tag{4.37}$$

$$\mathrm{d}\boldsymbol{t}_0 = \frac{\partial \boldsymbol{t}_0}{\partial \phi_0} \cdot \mathrm{d}\phi_0 + \frac{\partial \boldsymbol{t}_0}{\partial \lambda_0} \cdot \mathrm{d}\lambda_0 = \cos\lambda_0 \cdot \boldsymbol{u}_0 \cdot \mathrm{d}\phi_0 + \boldsymbol{v}_0 \cdot \mathrm{d}\lambda_0, \tag{4.38}$$

$$\mathrm{d}\boldsymbol{r} = \boldsymbol{u} \cdot \mathrm{d}x_\perp + \boldsymbol{v} \cdot \mathrm{d}y_\perp, \tag{4.39}$$

$$\mathrm{d}\boldsymbol{t} = \cos\lambda \cdot \boldsymbol{u} \cdot \mathrm{d}\phi + \boldsymbol{v} \cdot \mathrm{d}\lambda. \tag{4.40}$$

Moreover, since $\mathrm{d}\boldsymbol{r}$ is now defined to be a variation in a plane perpendicular to the track, the functional dependence of $\mathrm{d}s$ on the variations of position, direction, and momentum at the initial point can be evaluated by multiplying Eq. (4.26) with $\boldsymbol{t}$ and using the constraint $\mathrm{d}\boldsymbol{r} \cdot \boldsymbol{t} = 0$. One obtains

$$\mathrm{d}s = -\boldsymbol{t} \cdot \mathrm{d}\boldsymbol{r}_0 - \boldsymbol{t} \cdot \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{t}_0} \cdot \mathrm{d}\boldsymbol{t}_0 \right) - \left(\boldsymbol{t} \cdot \frac{\partial \boldsymbol{r}}{\partial \psi_0} \right) \cdot \mathrm{d}\psi_0. \tag{4.41}$$

Inserting Eq. (4.41) and Eqs. (4.28) – (4.34) into Eqs. (4.26) and (4.27), making use of Eqs. (4.37) and (4.40), yields a set of equations relating variations of the parameters at the initial surface to the variations at the destination surface. These can then be manipulated in a straightforward manner to yield the differentials of the parameters at the destination surface. Since, for instance, the differential $\mathrm{d}x_\perp$ is defined as

$$\mathrm{d}x_\perp = \frac{\partial x_\perp}{\partial \psi_0} \cdot \mathrm{d}\psi_0 + \frac{\partial x_\perp}{\partial \phi_0} \cdot \mathrm{d}\phi_0 + \frac{\partial x_\perp}{\partial \lambda_0} \cdot \mathrm{d}\lambda_0 + \frac{\partial x_\perp}{\partial x_{\perp 0}} \cdot \mathrm{d}x_{\perp 0} + \frac{\partial x_\perp}{\partial y_{\perp 0}} \cdot \mathrm{d}y_{\perp 0}, \tag{4.42}$$

the different terms in the desired Jacobian can be identified as the multiplying factors of the variations at the initial surface. A complete list of these can be found in Appendix A. Since a perfectly helical track model is assumed, the momentum at the destination surface is equal to the momentum at the initial surface.

4.4.1.2 Transformations Between Curvilinear and Local Frames at a Fixed Point on the Particle Trajectory

Consider the transformation from a local plane—defined by a unit vector $\boldsymbol{i}$ normal to the plane and two, orthogonal unit vectors $\boldsymbol{j}$ and $\boldsymbol{k}$ inside the plane—to the curvilinear frame. These three unit vectors define the coordinate system (u, v, w) introduced in Sect. 4.2. The aim is to derive the Jacobian of the transformation between

$\left(\psi, v', w', v, w\right)$, where $v' = \mathrm{d}v/\mathrm{d}u$ and $w' = \mathrm{d}w/\mathrm{d}u$, and $(\psi, \phi, \lambda, x_\perp, y_\perp)$ at a given point s on the particle trajectory. The relevant differentials are now

$$\mathrm{d}\boldsymbol{r} = \boldsymbol{u} \cdot \mathrm{d}x_\perp + \boldsymbol{v} \cdot \mathrm{d}y_\perp = \boldsymbol{j} \cdot \mathrm{d}v + \boldsymbol{k} \cdot \mathrm{d}w + \boldsymbol{t} \cdot \mathrm{d}s, \tag{4.43}$$

$$\mathrm{d}\boldsymbol{t} = \cos\lambda \cdot \boldsymbol{u} \cdot \mathrm{d}\phi + \boldsymbol{v} \cdot \mathrm{d}\lambda = \frac{\partial \boldsymbol{t}}{\partial v'} \cdot \mathrm{d}v' + \frac{\partial \boldsymbol{t}}{\partial w'} \cdot \mathrm{d}w' + \frac{\partial \boldsymbol{t}}{\partial s} \cdot \mathrm{d}s. \tag{4.44}$$

Again $\mathrm{d}\boldsymbol{r}$ is orthogonal to $\boldsymbol{t}$, therefore $\mathrm{d}s$ can be calculated from Eq. (4.43), in a similar manner as before. The result is

$$\mathrm{d}s = -(\boldsymbol{t} \cdot \boldsymbol{j}) \cdot \mathrm{d}v - (\boldsymbol{t} \cdot \boldsymbol{k}) \cdot \mathrm{d}w. \tag{4.45}$$

By inserting Eq. (4.45) into Eqs. (4.43) and (4.44) and following the same procedure as in Sect. 4.4.1.1, explicit expressions of the differentials can again be constructed. For this also $\boldsymbol{t}$ and its derivatives, expressed in the local parameters, are needed. The formulas are:

$$\boldsymbol{t} = \frac{1}{\sqrt{1 + v'^2 + w'^2}} \left[\boldsymbol{i} + v'\boldsymbol{j} + w'\boldsymbol{k}\right], \tag{4.46}$$

$$\frac{\partial \boldsymbol{t}}{\partial v'} = \frac{1}{\sqrt{1 + v'^2 + w'^2}} \left[\boldsymbol{j} - \frac{v'}{\sqrt{1 + v'^2 + w'^2}}\boldsymbol{t}\right], \tag{4.47}$$

$$\frac{\partial \boldsymbol{t}}{\partial w'} = \frac{1}{\sqrt{1 + v'^2 + w'^2}} \left[\boldsymbol{k} - \frac{w'}{\sqrt{1 + v'^2 + w'^2}}\boldsymbol{t}\right]. \tag{4.48}$$

The Jacobian of the transformation from the curvilinear to the local frame can be derived by inverting the Jacobian of the transformation from the local to the curvilinear frame. The expressions of both Jacobians can be found in Appendix A.

4.4.1.3 Transformations Between Global Cartesian and Local Frames

The global Cartesian frame $(p_x,\ p_y,\ p_z,\ x,\ y,\ z)$ is useful for vertex reconstruction purposes as well as for track reconstruction in zero and inhomogeneous magnetic fields. Zero magnetic field is typical in, for instance, test-beam applications.

First the transformations between a local frame and the global Cartesian frame is considered. The basic strategy is to go via an intermediate Cartesian frame aligned with the local frame under consideration. This Cartesian frame is related to the global Cartesian frame via a pure rotation. The intermediate Cartesian frame C' will be denoted by primed quantities $(p'_x,\ p'_y,\ p'_z,\ x',\ y',\ z')$, where the x'- and y'-axes are parallel to the v- and w-axes of the local frame, respectively. The Jacobian $\boldsymbol{R}$ of the transformation from the global, Cartesian frame to the intermediate frame C' consists of two similar, three-by-three blocks, each containing the relevant

rotation matrix. The other entries of this matrix are zero. The Jacobian $\boldsymbol{J}_{C'\to L}$ of the transformation from the primed Cartesian frame C' to the local frame L is constructed from the following formulas:

$$\frac{q}{p} = \frac{q}{\sqrt{{p'_x}^2 + {p'_y}^2 + {p'_z}^2}}, \tag{4.49}$$

$$v' = \frac{p'_x}{p'_z}, \tag{4.50}$$

$$w' = \frac{p'_y}{p'_z}. \tag{4.51}$$

The total Jacobian is given by the product $\boldsymbol{J}_{C'\to L} \cdot \boldsymbol{R}$.

For the inverse transformation, the Jacobian $\boldsymbol{J}_{L\to C'}$ can be derived from the following relations:

$$p'_x = \frac{q}{\psi} \cdot \frac{s_z \cdot v'}{\sqrt{1 + v'^2 + w'^2}}, \tag{4.52}$$

$$p'_y = \frac{q}{\psi} \cdot \frac{s_z \cdot w'}{\sqrt{1 + v'^2 + w'^2}}, \tag{4.53}$$

$$p'_z = \frac{q}{\psi} \cdot \frac{s_z}{\sqrt{1 + v'^2 + w'^2}}, \tag{4.54}$$

where $s_z = \text{sign}(p_z)$ is the sign of the z-component of the momentum vector in the local frame. It is needed in order to uniquely specify the state of the track in the local frame. The total Jacobian is in this case given by the product $\boldsymbol{R}^T \cdot \boldsymbol{J}_{L\to C'}$, using the same matrix $\boldsymbol{R}$ as above. The Jacobians $\boldsymbol{J}_{C'\to L}$ and $\boldsymbol{J}_{L\to C'}$ are shown in Appendix A.

4.4.2 Inhomogeneous Magnetic Fields

As mentioned above, the common way of obtaining the Jacobian matrices needed for the linearized error propagation is to expand the track propagator functions to first order in a Taylor series. As there are no analytical expressions for the track propagator in the case of inhomogeneous magnetic fields, this approach is unfortunately impossible. Another common technique is the error propagation by numerical derivatives described earlier. This method is slow but robust and accurate. A third way of obtaining the Jacobian matrices is to differentiate the recursion formulae of the numerical integration method directly. This is the essence of the so-called Bugge-Myrheim method [3].

In the parameter propagation, the propagated global track parameters:

$$\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \dot{\boldsymbol{r}} = \boldsymbol{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}, \tag{4.55}$$

are obtained by integrating the equations of motion, using some recursion formulas. With the Runge–Kutta–Nyström method, one step (numbered by n) becomes:

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n + \frac{h^2}{6}(\boldsymbol{k}_1 + \boldsymbol{k}_2 + \boldsymbol{k}_3) = \boldsymbol{F}_n(\boldsymbol{r}_n, \dot{\boldsymbol{r}}_n), \tag{4.56}$$

$$\dot{\boldsymbol{r}}_{n+1} = \dot{\boldsymbol{r}}_n + \frac{h}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4) = \boldsymbol{G}_n(\boldsymbol{r}_n, \dot{\boldsymbol{r}}_n). \tag{4.57}$$

To obtain the Jacobian matrix of the propagated global track parameters with respect to the initial track parameters $\boldsymbol{q}_i$, the recursion formulae (Eq. (4.57)) have to be differentiated with respect to $\boldsymbol{q}_i$, giving:

$$\boldsymbol{J}_{n+1} = \begin{pmatrix} \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{q}_i} \\ \frac{\partial \dot{\boldsymbol{r}}_{n+1}}{\partial \boldsymbol{q}_i} \end{pmatrix} = \begin{pmatrix} \frac{\partial \boldsymbol{F}_n}{\partial \boldsymbol{q}_i} \\ \frac{\partial \boldsymbol{G}_n}{\partial \boldsymbol{q}_i} \end{pmatrix} = \begin{pmatrix} \frac{\partial \boldsymbol{F}_n}{\partial \boldsymbol{r}_n} & \frac{\partial \boldsymbol{F}_n}{\partial \dot{\boldsymbol{r}}_n} \\ \frac{\partial \boldsymbol{G}_n}{\partial \boldsymbol{r}_n} & \frac{\partial \boldsymbol{G}_n}{\partial \dot{\boldsymbol{r}}_n} \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial \boldsymbol{r}_n}{\partial \boldsymbol{q}_i} \\ \frac{\partial \dot{\boldsymbol{r}}_n}{\partial \boldsymbol{q}_i} \end{pmatrix} = \boldsymbol{D}_n \cdot \boldsymbol{J}_n, \tag{4.58}$$

where the derivatives $\partial \boldsymbol{r}_n / \partial \boldsymbol{q}_i$ and $\partial \dot{\boldsymbol{r}}_n / \partial \boldsymbol{q}_i$ of the $6 \times j$ Jacobian $\boldsymbol{J}$ are given by the following $3 \times j$ matrices:

$$\frac{\partial \boldsymbol{r}_n}{\partial \boldsymbol{q}_i} = \begin{pmatrix} \frac{\partial x_n}{\partial q_{i,1}} & \cdots & \frac{\partial x_n}{\partial q_{i,j}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_n}{\partial q_{i,1}} & \cdots & \frac{\partial z_n}{\partial q_{i,j}} \end{pmatrix}, \quad \frac{\partial \dot{\boldsymbol{r}}_n}{\partial \boldsymbol{q}_i} = \begin{pmatrix} \frac{\partial t_{x,n}}{\partial q_{i,1}} & \cdots & \frac{\partial t_{x,n}}{\partial q_{i,j}} \\ \vdots & \ddots & \vdots \\ \frac{\partial t_{z,n}}{\partial q_{i,1}} & \cdots & \frac{\partial t_{z,n}}{\partial q_{i,j}} \end{pmatrix}. \tag{4.59}$$

$\boldsymbol{D}_n$ is a 6×6 matrix containing the recursion formulae $\boldsymbol{F}_n$ and $\boldsymbol{G}_n$ differentiated with respect to the global track parameters:

$$\boldsymbol{D}_n = \frac{\partial(\boldsymbol{F}_n, \boldsymbol{G}_n)}{\partial(\boldsymbol{r}_n, \dot{\boldsymbol{r}}_n)} = \begin{pmatrix} \frac{\partial \boldsymbol{F}_n}{\partial \boldsymbol{r}_n} & \frac{\partial \boldsymbol{F}_n}{\partial \dot{\boldsymbol{r}}_n} \\ \frac{\partial \boldsymbol{G}_n}{\partial \boldsymbol{r}_n} & \frac{\partial \boldsymbol{G}_n}{\partial \dot{\boldsymbol{r}}_n} \end{pmatrix}, \tag{4.60}$$

giving the 3×3 matrices

$$\frac{\partial \boldsymbol{F}_n}{\partial \boldsymbol{r}_n} = \begin{pmatrix} \frac{\partial F_{n,1}}{\partial x_n} & \cdots & \frac{\partial F_{n,1}}{\partial z_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_{n,3}}{\partial x_n} & \cdots & \frac{\partial F_{n,3}}{\partial z_n} \end{pmatrix}, \quad \frac{\partial \boldsymbol{F}_n}{\partial \dot{\boldsymbol{r}}_n} = \begin{pmatrix} \frac{\partial F_{n,1}}{\partial t_{x,n}} & \cdots & \frac{\partial F_{n,1}}{\partial t_{z,n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_{n,3}}{\partial t_{x,n}} & \cdots & \frac{\partial F_{n,3}}{\partial t_{z,n}} \end{pmatrix}, \tag{4.61}$$

and

$$\frac{\partial \boldsymbol{G}_n}{\partial \boldsymbol{r}_n} = \begin{pmatrix} \frac{\partial G_{n,1}}{\partial x_n} & \cdots & \frac{\partial G_{n,1}}{\partial z_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_{n,3}}{\partial x_n} & \cdots & \frac{\partial G_{n,3}}{\partial z_n} \end{pmatrix}, \quad \frac{\partial \boldsymbol{G}_n}{\partial \dot{\boldsymbol{r}}_n} = \begin{pmatrix} \frac{\partial G_{n,1}}{\partial t_{x,n}} & \cdots & \frac{\partial G_{n,1}}{\partial t_{z,n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_{n,3}}{\partial t_{x,n}} & \cdots & \frac{\partial G_{n,3}}{\partial t_{z,n}} \end{pmatrix}. \tag{4.62}$$

By writing the recursion formulae of the derivatives as a product of $\boldsymbol{D}_n$ and $\boldsymbol{J}_n$ (Eq. (4.58)), the recursion formulae $\boldsymbol{F}_n$ and $\boldsymbol{G}_n$ can be differentiated with respect to the global track parameters $\boldsymbol{r}_n$ and $\dot{\boldsymbol{r}}_n$ instead of the initial track parameters $\boldsymbol{q}_i$. This greatly simplifies the differentiation, giving:

$$\begin{aligned} \frac{\partial \boldsymbol{F}_n}{\partial \boldsymbol{r}_n} &= 1 + \frac{h^2}{6}\left(\frac{\partial \boldsymbol{k}_1}{\partial \boldsymbol{r}_n} + \frac{\partial \boldsymbol{k}_2}{\partial \boldsymbol{r}_n} + \frac{\partial \boldsymbol{k}_3}{\partial \boldsymbol{r}_n}\right), \\ \frac{\partial \boldsymbol{F}_n}{\partial \dot{\boldsymbol{r}}_n} &= h + \frac{h^2}{6}\left(\frac{\partial \boldsymbol{k}_1}{\partial \dot{\boldsymbol{r}}_n} + \frac{\partial \boldsymbol{k}_2}{\partial \dot{\boldsymbol{r}}_n} + \frac{\partial \boldsymbol{k}_3}{\partial \dot{\boldsymbol{r}}_n}\right), \\ \frac{\partial \boldsymbol{G}_n}{\partial \boldsymbol{r}_n} &= \frac{h}{6}\left(\frac{\partial \boldsymbol{k}_1}{\partial \boldsymbol{r}_n} + 2\frac{\partial \boldsymbol{k}_2}{\partial \boldsymbol{r}_n} + 2\frac{\partial \boldsymbol{k}_3}{\partial \boldsymbol{r}_n} + \frac{\partial \boldsymbol{k}_4}{\partial \boldsymbol{r}_n}\right), \\ \frac{\partial \boldsymbol{G}_n}{\partial \dot{\boldsymbol{r}}_n} &= 1 + \frac{h}{6}\left(\frac{\partial \boldsymbol{k}_1}{\partial \dot{\boldsymbol{r}}_n} + 2\frac{\partial \boldsymbol{k}_2}{\partial \dot{\boldsymbol{r}}_n} + 2\frac{\partial \boldsymbol{k}_3}{\partial \dot{\boldsymbol{r}}_n} + \frac{\partial \boldsymbol{k}_4}{\partial \dot{\boldsymbol{r}}_n}\right). \end{aligned} \tag{4.63}$$

In order to calculate these derivatives explicitly, the individual stages of the Runge–Kutta–Nyström method have to be differentiated with respect to the global track parameters:

$$\boldsymbol{A}_l = \frac{\partial \boldsymbol{k}_l}{\partial \dot{\boldsymbol{r}}_n}, \qquad \boldsymbol{C}_l = \frac{\partial \boldsymbol{k}_l}{\partial \boldsymbol{r}_n}, \tag{4.64}$$

where l denotes the individual stages, and $\boldsymbol{k}_l$ is given by the equations of motion of the global track parameters in Eq. (4.2):

$$\begin{aligned}\frac{\mathrm{d}^2x}{\mathrm{d}s^2} &= x'' = \xi\,(t_y B_z - t_z B_y),\\ \frac{\mathrm{d}^2y}{\mathrm{d}s^2} &= y'' = \xi\,(t_z B_x - t_x B_z),\\ \frac{\mathrm{d}^2z}{\mathrm{d}s^2} &= z'' = \xi\,(t_x B_y - t_y B_x),\end{aligned} \tag{4.65}$$

where $\xi \equiv k\psi$. Writing the 3×3 matrices $\boldsymbol{A}_l$ and $\boldsymbol{C}_l$ in a general form yields:

$$\boldsymbol{A} = \begin{pmatrix} \frac{\partial x''}{\partial t_x} & \cdots & \frac{\partial x''}{\partial t_z} \\ \vdots & \ddots & \vdots \\ \frac{\partial z''}{\partial t_x} & \cdots & \frac{\partial z''}{\partial t_z} \end{pmatrix} = \begin{pmatrix} 0 & \xi B_z & -\xi B_y \\ -\xi B_z & 0 & \xi B_x \\ \xi B_y & -\xi B_x & 0 \end{pmatrix} \tag{4.66}$$

and

$$\boldsymbol{C} = \begin{pmatrix} \frac{\partial x''}{\partial x} & \cdots & \frac{\partial x''}{\partial z} \\ \vdots & \ddots & \vdots \\ \frac{\partial z''}{\partial x} & \cdots & \frac{\partial z''}{\partial z} \end{pmatrix} = \xi \begin{pmatrix} (t_y B_{z;x} - t_z B_{y;x}) & (t_y B_{z;y} - t_z B_{y;y}) & (t_y B_{z;z} - t_z B_{y;z}) \\ (t_z B_{x;x} - t_x B_{z;x}) & (t_z B_{x;y} - t_x B_{z;y}) & (t_z B_{x;z} - t_x B_{z;z}) \\ (t_x B_{y;x} - t_y B_{x;x}) & (t_x B_{y;y} - t_y B_{x;y}) & (t_x B_{y;z} - t_y B_{x;z}) \end{pmatrix}, \tag{4.67}$$

with

$$B_{u;v} = \frac{\partial B_u}{\partial v}, \quad u, v \in \{x, y, z\}. \tag{4.68}$$

With the help of these matrices, the elements of the matrix $\boldsymbol{D}_n$ in Eq. (4.60) are computed. $\boldsymbol{D}_n$ is then multiplied by $\boldsymbol{J}_n$ to produce the transported Jacobian $\boldsymbol{J}_{n+1}$; see Eq. (4.58). This procedure is repeated for every recursion step, transforming $\boldsymbol{J}$ along the way. If, for instance, a planar detector element is reached at the end of the propagation steps and a measurement is to be included in a track reconstruction algorithm, the Jacobian can be transported from the global parameters to a local set of parameters as described in Sect. 4.4.1.3.

When applied to real problems, the field gradients in $\boldsymbol{C}$ are usually quite costly to calculate. The calculations can be significantly sped up by setting elements of $\boldsymbol{C}$ with negligible influence on the Jacobians to zero. The sensitivity of the Jacobians to the field gradients can be checked by a simulation study.

4.5 Material Effects

4.5.1 Multiple Scattering

4.5.1.1 The Distribution of the Scattering Angle

Elastic Coulomb scattering of particles heavier than electrons, including the muon, is dominated by the atomic nucleus. The PDF of the single scattering angle θ has the following form [9]:

$$f(\theta) = \begin{cases} \dfrac{k\theta}{(\theta^2 + a^2)^2} & \text{if } \theta \le b, \\ 0 & \text{otherwise,} \end{cases} \tag{4.69}$$

with the normalization constant

$$k = 2a^2 \left(1 + a^2/b^2\right). \tag{4.70}$$

The PDF in Eq. (4.69) is derived from the Rutherford scattering formula by setting $\sin\theta \approx \theta$ and introducing a minimal and a maximal scattering angle, thereby avoiding the singularity at $\theta = 0$. The minimal and maximal angles a, b depend on the nuclear charge Z and the atomic mass number A of the nucleus as well as on the momentum p of the scattered particle [9]:

$$a = \frac{2.66 \cdot 10^{-6} \cdot Z^{1/3}}{p}, \quad b = \frac{0.14}{A^{1/3} \cdot p}. \tag{4.71}$$

Here and in the following, p is assumed to be given in units of GeV/c. The ratio

$$\rho = b/a \approx \left(\frac{204}{Z^{1/3}}\right)^2 \tag{4.72}$$

depends only on the nuclear charge Z, with the exception of very heavy nuclei, where the approximation $A \approx 2Z$ breaks down. The size of ρ is typically of the order 10^4. The normalization constant k can therefore be approximated by $k \approx 2a^2$. It follows that the expectations of θ and θ^2 are approximately given by:

$$\mathsf{E}[\theta] = \frac{4.18 \cdot 10^{-6} \cdot Z^{1/3}}{p}, \quad \mathsf{E}[\theta^2] = \frac{2.84 \cdot 10^{-11} \cdot Z^{2/3} \cdot \ln(159\, Z^{-1/3})}{p^2}. \tag{4.73}$$

In track reconstruction, the scattering angle θ is of little use, as it is more convenient to work with two projected scattering angles instead, for instance, $\theta_x = \theta \cos\phi$ and $\theta_y = \theta \sin\phi$, if the particle runs parallel to the z-axis. Under

the assumption that the azimuthal angle ϕ is independent of θ and uniform in the interval $[0, 2\pi]$, their joint PDF is given by [10]:

$$g(\theta_x, \theta_y) = \begin{cases} \frac{1}{\pi} \cdot \frac{a^2}{(\theta_x^2 + \theta_y^2 + a^2)^2}, & \text{if } 0 \le \theta_x^2 + \theta_y^2 \le b^2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.74}$$

The support of the joint PDF is a circle around the origin with radius b. The projected angles θ_x and θ_y are uncorrelated but not independent. The marginal PDF of the projected angle in the interval $[-b, b]$ has the following form:

$$f(\theta_\mathrm{p}) = \frac{a^2}{\pi} \left[\frac{\sqrt{b^2 - \theta_\mathrm{p}^2}}{(a^2 + b^2)(\theta_\mathrm{p}^2 + a^2)} + \frac{\arccos\left(\sqrt{\theta_\mathrm{p}^2 + a^2} \,/ \sqrt{a^2 + b^2}\right)}{(\theta_\mathrm{p}^2 + a^2)^{3/2}} \right], \tag{4.75}$$

with θ_p either θ_x or θ_y. The marginal PDF has zero mean, a single mode at $\theta_\mathrm{p} = 0$ with $f(0) \approx 1/(2a)$, and the following variance:

$$\mathsf{var}\,[\theta_\mathrm{p}] = \mathsf{E}[\theta^2]/2 = \frac{1.42 \cdot 10^{-11} \cdot Z^{2/3} \cdot \ln(159\, Z^{-1/3})}{p^2}. \tag{4.76}$$

The range of the distribution is very large. For silicon ($Z = 14$), it is about ± 2500 standard deviations.

If the projected scattering angle is still small after N scattering processes, it is approximately equal to the sum of the individual projected scattering angles. Its distribution can be obtained either by the convolution of N single scattering distributions [10] or by the Molière theory [11–13]. Figure 4.7 shows the two PDFs for a muon with $p = 1$ GeV/c and four silicon scatterers. Their thickness d is given in fractions of a radiation length [14]. The corresponding numbers of scattering processes in the four scatterers have been chosen as $N = 2^{10}, 2^{13}, 2^{16}$ and 2^{19}, respectively, spanning the range from about 0.2% to about 90% of a radiation length. There is excellent agreement for all scatterers but the thickest one; in this case, the tails are more persistent in the Molière PDF than in the PDF obtained by convolution. The discrepancy is at angles larger than the maximum scattering angle b, which is equal to about 46 mrad in silicon for $p = 1$ GeV/c.

For the purpose of track reconstruction, the "exact" distribution of the projected scattering angle has to be approximated by a single Gaussian or a mixture of Gaussians; see Sect. 6.2.3. The most commonly used approximation by a single Gaussian is the Highland formula, proposed in [15] and modified in [16]. The most up-to-date version can be found in [14], which gives the standard deviation of the projected scattering angle as:

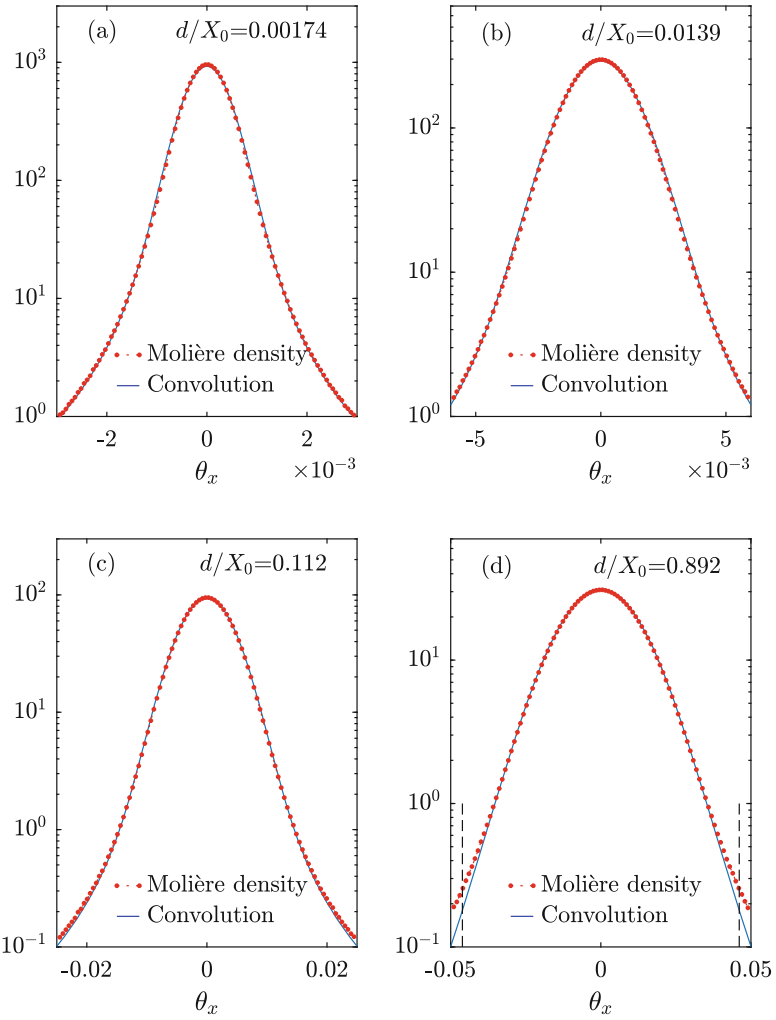

Fig. 4.7 Probability density functions of the projected multiple scattering angle for silicon targets obtained by convolution (solid lines) and frequency distributions obtained by simulation from the Molière densities (dots). The vertical dashed lines in figure (d) show the upper limit $b \approx 46$ mrad in silicon

$$\sqrt{\mathsf{var}\,[\theta_{\mathrm{p}}]} \approx \frac{0.0136\,z}{\beta p}\sqrt{\frac{d}{X_0}}\left[1 + 0.038\ln\left(\frac{d\,z^2}{\beta^2 X_0}\right)\right], \tag{4.77}$$

where p, β and z are the momentum, velocity, and charge number of the incident particle, and d/X_0 is the thickness of the scatterer in units of the radiation length.

If the scatterer consists of a composite material with k components, its radiation length X_0 is given by:

$$\frac{1}{X_0} = \sum_{i=1}^{k} \frac{f_i}{X_i}, \tag{4.78}$$

where X_i is the radiation length of component i in g/cm^2 and f_i is the mass fraction of component i [17]. In the following, it is more convenient to express d and X_0 in centimeters:

$$\frac{X_0}{\text{cm}} = \frac{\text{g/cm}^3}{\varrho} \frac{X_0}{\text{g/cm}^2}, \tag{4.79}$$

where ϱ is the density of the material in units of g/cm^3.

In a thin scatterer, a Gaussian distribution is but a poor approximation of the actual distribution of the multiple scattering angle, because of the latter's large range. A better, though still far from perfect, approximation can be obtained using a normal mixture with two components, one modeling the "core", the other modeling the "tails" [10, 18]. The standardized two-component mixture PDF of θ_p with variance equal to 1 has the following form:

$$f(\theta_\text{p}) = (1-\varepsilon) \cdot \varphi(\theta_\text{p}; 0, \sigma_1^2) + \varepsilon \cdot \varphi(\theta_\text{p}; 0, \sigma_2^2), \tag{4.80}$$

where $\sigma_1^2 < \sigma_2^2$, $\varepsilon < 1/2$ and $\sigma_2^2 = [1-(1-\varepsilon)\,\sigma_1^2]/\varepsilon$.

The core variance σ_1^2 is parametrized in terms of the reduced thickness $d_0' = d/(\beta^2 X_0)$, where X_0 is the radiation length of the scatterer:

$$\sigma_1^2 = 8.471 \cdot 10^{-1} + 3.347 \cdot 10^{-2} \cdot \ln d_0' - 1.843 \cdot 10^{-3} \cdot (\ln d_0')^2 \tag{4.81}$$

The tail weight ε is parametrized in terms of a modified reduced thickness $d_0'' = Z^{2/3} d/(\beta^2 X_0)$:

$$\varepsilon = \begin{cases} 4.841 \cdot 10^{-2} + 6.348 \cdot 10^{-3} \cdot \ln d_0'' + 6.096 \cdot 10^{-4} \cdot (\ln d_0'')^2, & \text{if } \ln d_0'' < 0.5, \\ -1.908 \cdot 10^{-2} + 1.106 \cdot 10^{-1} \cdot \ln d_0'' - 5.729 \cdot 10^{-3} \cdot (\ln d_0'')^2, & \text{if } \ln d_0'' \geq 0.5, \end{cases} \tag{4.82}$$

Finally, the standardized PDF in Eq. (4.80) is scaled with the total standard deviation of the projected scattering angle without the logarithmic correction; see Eq. (4.77). This ensures that the variance of the scattering angle is strictly additive if the scatterer is divided into thinner slices. For a comparison of the mixture model with simulations by GEANT4 [19], see [18].

4.5.1.2 Multiple Scattering in Track Propagation

If a charged particle is propagated through material, the covariance matrix of the track parameters is augmented by the additional uncertainty on direction and possibly position caused by multiple scattering. The algorithmic treatment is different for "thin" and "thick" scatterers. By definition, in a thin scatterer the offset—the change in position of the passing particle—is negligible in relation to the spatial resolution of the surrounding detectors, so that only the direction is affected. For instance, in a silicon sensor with a typical thickness of 0.3 mm the standard deviation of the offset is less than 0.2 µm for momenta above 1 GeV; see Eq. (4.107) below. As the spatial resolution is in the order of 10 µm, the offset can be safely neglected, and the sensor is considered as a thin scatterer.

If, on the other hand, the scatterer is a 5 cm thick iron absorber in a muon spectrometer, the standard deviation of the offset is about 0.66 mm for a muon with 1 GeV, and the offset can be as large as 2 mm. This is no longer negligible in a muon spectrometer equipped with drift chambers having a typical spatial resolution of 150 µm. The absorber should therefore be treated as a thick scatterer, at least for low-momentum muons.

Thin Scatterers

Consider a thin scatterer with nominal thickness d, radiation length X_0, and unit normal vector $\boldsymbol{n}$ at the point where the track crosses the scatterer. The track is specified by a vector $\boldsymbol{q}$ of track parameters and the associated covariance matrix $\boldsymbol{C}$. In a thin scatterer, only the sub-matrix of $\boldsymbol{C}$ corresponding to the track direction is augmented. The details depend on the choice of the track parametrization; see Sect. 4.2.

A. The track is parametrized as in Eq. (4.3):

$$q_1 = \frac{q}{p_\mathrm{T}},\ q_2 = \phi,\ q_3 = \tan\lambda,\ q_4 = R\Phi,\ q_5 = z. \tag{4.83}$$

1. Compute the unit direction vector $\boldsymbol{a}$ and the momentum p of the track:

$$\boldsymbol{a} = (\cos\phi\cos\lambda, \sin\phi\cos\lambda, \sin\lambda)^\mathsf{T},\quad p = 1/(|\,q_1|\cos\lambda). \tag{4.84}$$

2. Compute the effective thickness t in units of X_0:

$$t = \frac{d\,|\boldsymbol{a}|}{|\boldsymbol{a}\cdot\boldsymbol{n}|\,X_0}. \tag{4.85}$$

3. Compute the variance of the projected multiple scattering angle, assuming $\beta = 1$ and $q = 1$ if unknown:

$$\mathsf{var}\,[\theta_{\mathrm{p}}] = \frac{1.85 \cdot 10^{-4} \cdot t}{p^2}\,(1 + 0.038 \ln t)^2\,. \tag{4.86}$$

4a. If $\sin\phi = 0$ compute the covariance matrix $\boldsymbol{D}$ of $(a_2, a_3)^{\mathsf{T}}$:

$$\boldsymbol{D} = \boldsymbol{J} \cdot \boldsymbol{C}\,(2{:}3,2{:}3) \cdot \boldsymbol{J}^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J} = \begin{pmatrix} \cos\phi\cos\lambda & 0 \\ 0 & \cos^3\lambda \end{pmatrix}. \tag{4.87}$$

Augment $\boldsymbol{D}$ by the contribution of multiple scattering [1]:

$$\boldsymbol{D}' = \boldsymbol{D} + \mathsf{var}\,[\theta_{\mathrm{p}}] \cdot \begin{pmatrix} 1 - {a_2}^2 & -a_2 a_3 \\ -a_2 a_3 & 1 - a_3^2 \end{pmatrix}. \tag{4.88}$$

Modify $\boldsymbol{C}$:

$$\boldsymbol{C}\,(2{:}3,2{:}3) = \boldsymbol{J}^{-1} \cdot \boldsymbol{D}' \cdot (\boldsymbol{J}^{-1})^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J}^{-1} = \begin{pmatrix} 1/\cos\phi\cos\lambda & 0 \\ 0 & 1/\cos^3\lambda \end{pmatrix}. \tag{4.89}$$

4b. If $\sin\phi \neq 0$ compute the covariance matrix $\boldsymbol{D}$ of $(a_1, a_3)^{\mathsf{T}}$:

$$\boldsymbol{D} = \boldsymbol{J} \cdot \boldsymbol{C}\,(2{:}3,2{:}3) \cdot \boldsymbol{J}^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J} = \begin{pmatrix} -\sin\phi\cos\lambda & -\cos\phi\sin\lambda\cos^2\lambda \\ 0 & \cos^3\lambda \end{pmatrix}. \tag{4.90}$$

Augment $\boldsymbol{D}$ by the contribution of multiple scattering [1]:

$$\boldsymbol{D}' = \boldsymbol{D} + \mathsf{var}\,[\theta_{\mathrm{p}}] \cdot \begin{pmatrix} 1 - {a_1}^2 & -a_1 a_3 \\ -a_1 a_3 & 1 - a_3^2 \end{pmatrix}. \tag{4.91}$$

Modify $\boldsymbol{C}$:

$$\boldsymbol{C}\,(2{:}3,2{:}3) = \boldsymbol{J}^{-1} \cdot \boldsymbol{D}' \cdot (\boldsymbol{J}^{-1})^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J}^{-1} = \begin{pmatrix} -1/\sin\phi\cos\lambda & -\cos\phi\sin\lambda/\sin\phi\cos^2\lambda \\ 0 & 1/\cos^3\lambda \end{pmatrix}. \tag{4.92}$$

B. The track is parametrized as in Eq. (4.4):

$$q_1 = \psi,\ q_2 = \mathrm{d}v/\mathrm{d}u,\ q_3 = \mathrm{d}w/\mathrm{d}u,\ q_4 = v,\ q_5 = w. \tag{4.93}$$

1. Compute the direction vector $\boldsymbol{a}$ and the momentum p of the track:

$$\boldsymbol{a} = (q_2, q_3, 1)^{\mathsf{T}}, \quad p = 1/|\,q_1|. \tag{4.94}$$

2. Compute the effective thickness t and the variance of the projected multiple scattering angle as in Eqs. (4.85) and (4.86).
3. Extract the covariance matrix $\boldsymbol{D}$ of $(q_2, q_3)^{\mathsf{T}}$:

$$\boldsymbol{D} = \boldsymbol{C}\,(2{:}3,2{:}3). \tag{4.95}$$

4. Augment $\boldsymbol{D}$ by the contribution of multiple scattering [1] and modify $\boldsymbol{C}$:

$$\boldsymbol{C}\,(2{:}3,2{:}3) = \boldsymbol{D} + \mathsf{var}\,[\theta_{\mathrm{p}}] \cdot (1 + q_2^2 + q_3^2) \cdot \begin{pmatrix} 1+q_2^2 & q_2 q_3 \\ q_2 q_3 & 1+q_3^2 \end{pmatrix}. \tag{4.96}$$

C. The track is parametrized as in Eq. (4.5):

$$q_1 = \frac{q}{p},\ q_2 = \phi,\ q_3 = \lambda,\ q_4 = x_\perp,\ q_5 = y_\perp. \tag{4.97}$$

1. Compute the unit direction vector $\boldsymbol{a}$ and the momentum p of the track:

$$\boldsymbol{a} = (\cos\phi\cos\lambda, \sin\phi\cos\lambda, \sin\lambda)^{\mathsf{T}}, \quad p = 1/|\,q_1|. \tag{4.98}$$

2. Compute the effective thickness t and the variance of the projected multiple scattering angle as in Eqs. (4.85) and (4.86).

3a. If $\sin\phi = 0$, compute the covariance matrix $\boldsymbol{D}$ of $(a_2, a_3)^{\mathsf{T}}$:

$$\boldsymbol{D} = \boldsymbol{J} \cdot \boldsymbol{C}\,(2{:}3,2{:}3) \cdot \boldsymbol{J}^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J} = \begin{pmatrix} \cos\phi\cos\lambda & 0 \\ 0 & \cos\lambda \end{pmatrix}. \tag{4.99}$$

Augment $\boldsymbol{D}$ by the contribution of multiple scattering as in Eq. (4.88) and modify $\boldsymbol{C}$:

$$\boldsymbol{C}\,(2{:}3,2{:}3) = \boldsymbol{J}^{-1} \cdot \boldsymbol{D}' \cdot (\boldsymbol{J}^{-1})^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J}^{-1} = \begin{pmatrix} 1/\cos\phi\cos\lambda & 0 \\ 0 & 1/\cos\lambda \end{pmatrix}. \tag{4.100}$$

3b. If $\sin\phi \neq 0$, compute the covariance matrix $\boldsymbol{D}$ of $(a_1, a_3)^{\mathsf{T}}$:

$$\boldsymbol{D} = \boldsymbol{J} \cdot \boldsymbol{C}\,(2{:}3,2{:}3) \cdot \boldsymbol{J}^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J} = \begin{pmatrix} -\sin\phi\cos\lambda & -\cos\phi\sin\lambda \\ 0 & \cos\lambda \end{pmatrix}. \tag{4.101}$$

Augment $\boldsymbol{D}$ by the contribution of multiple scattering as in Eq. (4.91) and modify $\boldsymbol{C}$:

$$\boldsymbol{C}\,(2{:}3,2{:}3) = \boldsymbol{J}^{-1} \cdot \boldsymbol{D}' \cdot (\boldsymbol{J}^{-1})^{\mathsf{T}}, \quad \text{with}$$
$$\boldsymbol{J}^{-1} = \begin{pmatrix} -1/\sin\phi\cos\lambda & -\cos\phi\sin\lambda/\sin\phi\cos^2\lambda \\ 0 & 1/\cos\lambda \end{pmatrix}. \tag{4.102}$$

Thick Scatterers

A thick scatterer can be treated in two ways: sliced into a number of thin scatterers or considered a single scatterer. The first approach gives more precise results, in particular when the scatterer is magnetized and the incoming particle is expected to be strongly deflected and suffer significant energy loss. A typical example for this situation is a low-momentum muon crossing the instrumented iron return yoke of the CMS experiment [20].

In the second approach, the uncertainties of both the direction and the position at the exit point have to be increased. In addition, energy loss may have to be considered, and an estimate of the actual track length L in the scatterer must be available. If no estimate of L is returned by the track propagation algorithm, L can be approximated by the distance between the entry and the exit point times a safety factor that depends on the momentum and can be tuned by simulation. Alternatively, L can be determined from the most probable trajectory in the scatterer [21].

Assume that the track parameters and their covariance matrix at the exit of the scatterer are given by $\boldsymbol{q}$ and $\boldsymbol{C}$ in the curvilinear parametrization; see Eq. (4.97).

1. Extract the entry and exit momenta p_1 and p_2. Under the assumption that the momentum p drops linearly between the entry and the exit, the average of $1/p^2$ is $1/(p_1 p_2)$.
2. Compute the variance per unit length of the projected multiple scattering angle:

$$t = \frac{1}{X_0}, \quad \sigma_0^2 = \frac{1.85 \cdot 10^{-4} \cdot t}{p_1 p_2}\,(1 + 0.038 \ln t)^2\,. \tag{4.103}$$

3. Extract the covariance matrix $\boldsymbol{D}$ of $(q_2, q_3, q_4, q_5)^{\mathsf{T}}$:

$$\boldsymbol{D} = \boldsymbol{C}\,(2{:}5,2{:}5). \tag{4.104}$$

4. Compute the unit direction vector $\boldsymbol{a}$ in the global coordinate system:

$$\boldsymbol{a} = (\cos\phi\cos\lambda, \sin\phi\cos\lambda, \sin\lambda)\,. \tag{4.105}$$

5. Transform $\boldsymbol{a}$ into the curvilinear system by rotating $\boldsymbol{a}$ into $\boldsymbol{a}' = (0, 0, 1)^{\mathsf{T}}$:

$$\boldsymbol{a}' = \boldsymbol{U}\boldsymbol{a}, \quad \text{with} \quad \boldsymbol{U} = \begin{pmatrix} \dfrac{{a_1}^2 a_3 + {a_2}^2}{1 - {a_3}^2} & -\dfrac{a_1 a_2}{1 + a_3} & -a_1 \\ -\dfrac{a_1 a_2}{1 + a_3} & \dfrac{{a_1}^2 + {a_2}^2 a_3}{1 - {a_3}^2} & -a_2 \\ a_1 & a_2 & a_3 \end{pmatrix}. \tag{4.106}$$

6. Compute the joint covariance matrix $\boldsymbol{E}$ of $\boldsymbol{q}' = (a_1', a_2', x_\perp, y_\perp)^{\mathsf{T}}$ [1]:

$$\boldsymbol{E} = \sigma_0^2 \cdot \begin{pmatrix} L & 0 & L^2/2 & 0 \\ 0 & L & 0 & L^2/2 \\ L^2/2 & 0 & L^3/3 & 0 \\ 0 & L^2/2 & 0 & L^3/3 \end{pmatrix}. \tag{4.107}$$

7. Rotate the direction part of $\boldsymbol{E}$ back into the global system:

$$\boldsymbol{E}' = \boldsymbol{T} \cdot \boldsymbol{E} \cdot \boldsymbol{T}^{\mathsf{T}}, \quad \text{with} \quad \boldsymbol{T} = \begin{pmatrix} \dfrac{{a_1}^2 a_3 + {a_2}^2}{1 - {a_3}^2} & -\dfrac{a_1 a_2}{1 + a_3} & 0 & 0 \\ -\dfrac{a_1 a_2}{1 + a_3} & \dfrac{{a_1}^2 + {a_2}^2 a_3}{1 - {a_3}^2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.108}$$

8. Transform the direction cosines a_1, a_2 back to ϕ, λ:

$$\boldsymbol{E}'' = \boldsymbol{J} \cdot \boldsymbol{E}' \cdot \boldsymbol{J}^{\mathsf{T}}, \quad \text{with}$$

$$\boldsymbol{J} = \begin{pmatrix} -\sin\phi/\cos\lambda & \cos\phi/\cos\lambda & 0 & 0 \\ -\cos\phi/\sin\lambda & -\sin\phi/\sin\lambda & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{if} \quad \lambda \neq 0; \quad \text{and} \tag{4.109}$$

$$\boldsymbol{J} = \begin{pmatrix} -\sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{if} \quad \lambda = 0. \tag{4.110}$$

9. Augment $\boldsymbol{D}$ by the contribution of multiple scattering and modify $\boldsymbol{C}$:

$$\boldsymbol{C}\,(2{:}5,2{:}5) = \boldsymbol{D} + \boldsymbol{E}''. \tag{4.111}$$

4.5.2 Energy Loss by Ionization

4.5.2.1 Mean Energy Loss

A heavy, charged particle passing through material suffers loss of energy due to ionization of the material. The mean energy loss of the particle is given by the Bethe–Bloch formula [14]:

$$\frac{\mathrm{d}E}{\mathrm{d}s} = -Kz^2\frac{Z\rho}{A\beta^2}\left(\frac{1}{2}\ln\frac{2m_ec^2\beta^2\gamma^2W_{\max}}{I^2} - \beta^2 - \frac{\delta}{2}\right), \tag{4.112}$$

where K is a constant, z is the charge number of the incoming particle, Z and A are the atomic number and atomic mass of the material, ρ is the density of the material, m_e is the electron mass, $W_{\max}$ is the maximum energy transfer in a single collision, I is the mean excitation energy, and δ is the density effect correction. For an incoming particle of mass M, $W_{\max}$ is given by:

$$W_{\max} = \frac{2m_ec^2\beta^2\gamma^2}{1 + 2\gamma m_e/M + (m_e/M)^2}. \tag{4.113}$$

For light particles such as electrons and positrons, the Bethe–Bloch formula needs some modifications. An alternative expression for light particles is [1]:

$$\frac{\mathrm{d}E}{\mathrm{d}s} = -K\frac{Z\rho}{A}\left(\ln\frac{2m_ec^2}{I} + 1.5\ln\gamma - 0.975\right). \tag{4.114}$$

4.5.2.2 Ionization Energy Loss in Track Propagation

In tracking detectors, material layers traversed during propagation are often considered as discrete. Knowing the thickness of the traversed layer, the Bethe-Bloch formula is used to modify the momentum part of the track parameter vector by the expected change before propagating to the next layer. Fluctuations in the ionization energy loss are often considered so small that they are neglected.

For detectors such as electromagnetic or hadronic calorimeters, however, ionization energy loss takes place continuously during propagation. In this case, it is possible to augment the vector of global track parameters in Eq. (4.55) with a parameter containing the track momentum [3, 22], e. g., $\xi = k\psi$ as defined in Sect. 4.4.2:

$$\boldsymbol{u} = (x, y, z, \Xi)^\mathsf{T}, \quad \dot{\boldsymbol{u}} = \left(t_x, t_y, t_z, \xi\right)^\mathsf{T}, \tag{4.115}$$

where Ξ is an auxiliary parameter corresponding to the integrated change in ξ. Track parameter and Jacobian matrix propagations can thereby be carried out in

way very similar to the one described in Sect. 4.4.2, effectively including effects of energy loss continuously while traversing the detector.

4.5.3 Energy Loss by Bremsstrahlung

4.5.3.1 Mean and Distribution of the Energy Loss

High-energy electrons lose energy in a material mainly by bremsstrahlung [14]. The dependence on the material can be summarized in a characteristic length, called the radiation length X_0. It is defined as the average distance over which a high-energy electron loses $1 - 1/e \approx 63\%$ of its energy. Although it is not strictly identical to the radiation length that is characteristic for multiple scattering (see Sect. 4.5.1), the same length is used in both contexts for the sake of convenience.

The rate of the mean energy loss by bremsstrahlung is nearly proportional to the energy:

$$\mathsf{E}\left[-\frac{\mathrm{d}E}{\mathrm{d}s}\right] \approx \frac{E}{X_0}, \tag{4.116}$$

and on average, the energy decreases approximately exponentially as a function of the reduced path length $t = s/X_0$:

$$\mathsf{E}\left[E(t)\right] = E_0 \exp(-t), \tag{4.117}$$

where E_0 is the initial energy. The energy loss is subject to large fluctuations, as a substantial part of the electron energy can be carried away by a single photon. A simplified PDF of E as a function of t has the following form, called the Bethe–Heitler model [23]:

$$h(E) = \frac{\left[\ln(E_0/E)\right]^{t/\ln 2 - 1}}{E_0\, \Gamma\left(t/\ln 2\right)}. \tag{4.118}$$

The PDF can be rewritten in terms of the remaining energy fraction $z = E/E_0$:

$$f(z) = \frac{(-\ln z)^{t/\ln 2 - 1}}{\Gamma\left(t/\ln 2\right)}. \tag{4.119}$$

It can be seen that $-\ln z$ is Gamma-distributed. The expectation and the variance can be computed explicitly [24]:

$$\mathsf{E}\left[z\right] = \exp\left(-t\right), \quad \mathsf{var}\left[z\right] = \exp\left(-t \ln 3/\ln 2\right) - \exp\left(-2t\right). \tag{4.120}$$

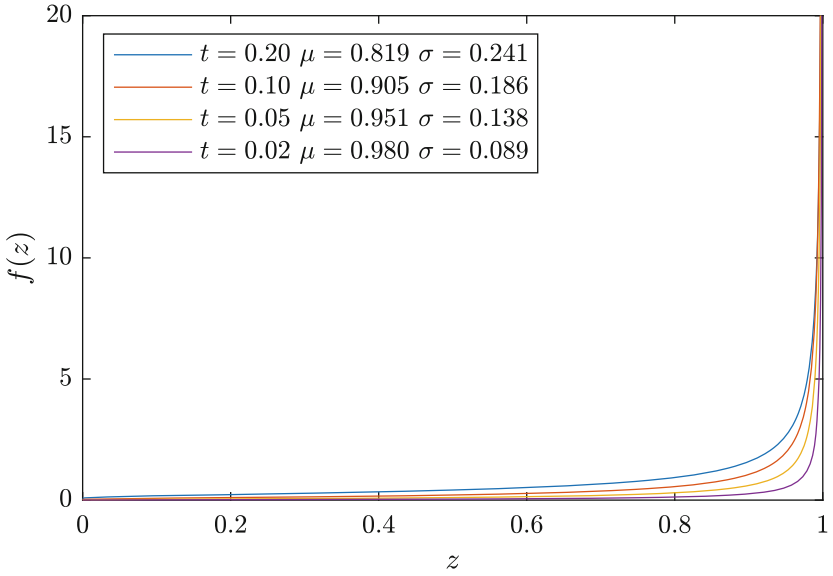

Fig. 4.8 Probability density function of the Bethe–Heitler model of bremsstrahlung in Eq. (4.119), with mean μ and standard deviation σ, for $t = 0.2, 0.1, 0.05, 0.02$

Figure 4.8 shows that the shape of the PDF is very far from being Gaussian; thus, the distribution cannot be adequately described by merely its mean and variance.

4.5.3.2 Approximation by Gaussian Mixtures

For electron reconstruction with the Gaussian-sum filter (GSF; see Sect. 6.2.3) the model PDF in Eq. (4.119) is approximated by a normal mixture PDF with n_c components. The parameters of the mixture are determined by minimizing some measure of distance between the two distributions. In [25], two distances have been used: D_{KL}, the Kullback–Leibler distance, and D_{CDF}, the integral over the absolute difference of the respective cumulative distribution functions (CDFs):

$$D_{KL} = \int_0^1 \ln[f(z)/g(z)]\, f(z)\, dz, \tag{4.121}$$

$$D_{CDF} = \int_{-\infty}^{\infty} |F(z) - G(z)|\, dz, \tag{4.122}$$

where $f(z)$ and $F(z)$ are the PDF and CDF of the model distribution, and $g(z)$ and $G(z)$ are the PDF and CDF of the normal mixture, respectively. Using D_{KL} and $n_c = 1$ the single Gaussian with the correct first two moments is recovered. In all other cases, the mixtures do not have the same moments as the model. The quality

of the approximating mixtures has been investigated in detail in [25]. Software in the form of a MATLAB® function can be downloaded from the URL in [26].

References

1. R. Frühwirth et al., *Data Analysis Techniques for High-Energy Physics*, 2nd edn. (Cambridge University Press, Cambridge, 2000)
2. S. Gorbunov, I. Kisel, Nucl. Instrum. Methods Phys. Res. A **559**(1), 148 (2006)
3. L. Bugge, J. Myrheim, Nucl. Instrum. Methods **179**, 365 (1981)
4. A. Strandlie, W. Wittek, Nucl. Instrum. Methods **A566**(2), 687 (2006)
5. E. Fehlberg, Classical Fifth-, Sixth-, Seventh-, and Eighth-Order Runge-Kutta Formulas with Stepsize Control. Technical Report NASA-TR-R-287, NASA, NASA Marshall Space Flight Center; Huntsville (1968). https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19680027281.pdf
6. C. Tsitouras, S.N. Papakostas, SIAM J. Sci. Comput. **20**(6), 2067 (1999)
7. E. Lund, L. Bugge, I. Gavrilenko, A. Strandlie, J. Instrum. **4**, P04001 (2009)
8. W. Press et al., *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007). http://apps.nrbook.com/empanel/index.html
9. J. Jackson, *Classical Electrodynamics*, 3rd edn. (Wiley, Hoboken, 1999)
10. R. Frühwirth, M. Regler, Nucl. Instrum. Methods Phys. Res. A **456**(3), 369 (2001)
11. G. Molière, Z. Naturforschung A **3**(2), 78 (1948). https://tinyurl.com/MoliereTheoryII
12. H.A. Bethe, Phys. Rev. **89**(6), 1256 (1953)
13. B. Gottschalk et al., Nucl. Instrum. Methods Phys. Res. B **74**(4), 467 (1993)
14. M. Tanabashi et al., Phys. Rev. D **98**, 030001 (2018). http://pdg.lbl.gov/2018/reviews/rpp2018-rev-passage-particles-matter.pdf
15. V.L. Highland, Nucl. Instrum. Methods **129**(2), 497 (1975)
16. G.R. Lynch, O.I. Dahl, Nucl. Instrum. Methods Phys. Res. B **58**(1), 6 (1991)
17. M. Gupta, Calculation of radiation length in materials. Technical Report PH-EP-Tech-Note-2010-013, CERN, Geneva (2010). https://cds.cern.ch/record/1279627
18. R. Frühwirth, M. Liendl, Comput. Phys. Commun. **141**, 230 (2001)
19. GEANT4 Collaboration, GEANT4 A Simulation Toolkit (2017). https://geant4.web.cern.ch
20. CMS Collaboration, S. Chatrchyan et al., J. Instrum. **3**(08), S08004 (2008). https://doi.org/10.1088/1748-0221/3/08/S08004
21. S. Chatzidakis, Z. Liu, J.P. Hayward, J.M. Scaglione, J. Appl. Phys. **123**(12), 124903 (2018)
22. E. Lund, L. Bugge, I. Gavrilenko, A. Strandlie, J. Instrum. **4**, P04016 (2009)
23. H. Bethe, W. Heitler, Proc. R. Soc. Lond. A **146**, 83 (1934). https://doi.org/10.1098/rspa.1934.0140
24. D. Stampfer, M. Regler, R. Frühwirth, Comput. Phys. Commun. **79**(2), 157 (1994)
25. R. Frühwirth, Comput. Phys. Commun. **154**(2), 131 (2003)
26. R. Frühwirth, MATLAB® function get_mixture. https://cernbox.cern.ch/index.php/s/K87fA77XQMcA821

Chapter 5
Track Finding

Abstract There is no systematic theory of track finding yet. Therefore, the first section of this chapter presents a list of basic techniques which have been successfully used, stand-alone or in combination, in past and present experiments. Among them are the conformal transformation, the Hough and the Legendre transform, cellular automata and neural networks, pattern matching, and track following by the combinatorial Kalman filter. The following section gives a brief excursion into online or real-time track finding in the collider experiments CDF, ATLAS, and CMS. As track finding in most cases delivers some candidates that do not correspond to actual particle tracks, the concluding section discusses methods for an efficient selection of valid candidates.

5.1 Basic Techniques

5.1.1 Conformal Transformation

Circles in the plane passing through the origin can be transformed into straight lines by the following mapping [1]:

$$u = \frac{x}{x^2 + y^2}, \quad v = \frac{y}{x^2 + y^2}. \tag{5.1}$$

The mapping is conformal, i.e., preserves angles between curves. Assume that a circle is defined by the equation:

$$(x - a)^2 + (y - b)^2 = R^2 = a^2 + b^2. \tag{5.2}$$

Expansion of the left hand side and division by $x^2 + y^2$ gives a linear equation in u and v:

$$2au + 2bv = 1. \tag{5.3}$$

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_5

This is the equation of a straight line in the (u, v)-plane with distance $d = 1/(2R)$ from the origin. A circle with a large radius R or small curvature κ is therefore transformed into a line that passes very close to the origin (Fig. 5.1). In the limit of zero curvature, the circle becomes a line transformed into itself by the mapping in Eq. (5.1). Both circle finding and circle fitting can be simplified by this transformation from circles to straight lines.

5.1.2 Hough Transform

The Hough transform [2] is a technique that finds clusters of points that lie on or close to a parametric curve such as a straight line or a circle. The number of parameters is usually two or three. In the simplest case, there is a set of points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ in the plane (image space) that lie on a straight line, parameterized by $y = k_0 x + d_0$, where k_0 is the slope and d_0 is the intercept of the line. A line passing through (x_i, y_i) fulfills the equation $d = -x_i k + y_i$, which is the equation of a line ℓ_i in the parameter space (Hough space) of k and d. The point of intersection of two lines ℓ_i, ℓ_j is just (k_0, d_0), the parameters of the original line. Finding straight lines in the image space is therefore equivalent to finding intersection points in the Hough space.

In practice, the measured points do not lie exactly on a straight line, and the lines in the Hough space do not intersect exactly in a single point. The usual approach is to define a binning in the Hough space and count the number of lines crossing each bin. Peaks in the 2D histogram correspond to lines that are close to many points in the image space. The size of the bins depends on the distribution of the measurement

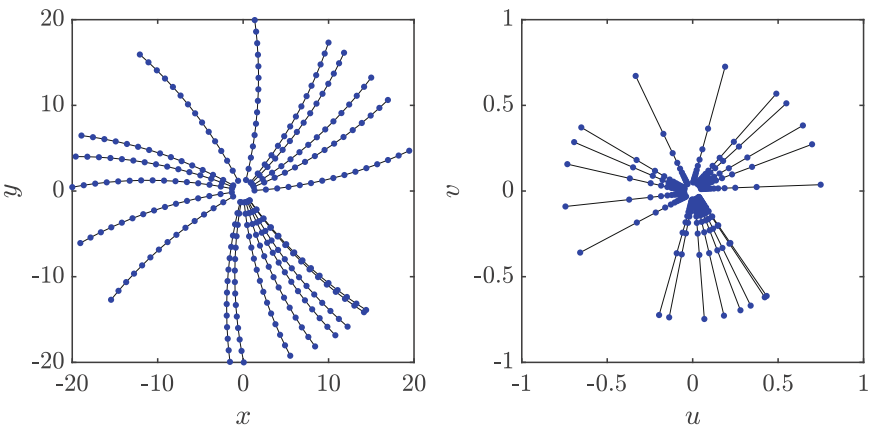

Fig. 5.1 Conformal transformation of the circles through the origin in the (x, y)-plane (left) into lines in the (u, v)-plane (right)

errors and can be tuned on simulated tracks. Alternatively, a 2D binary search can be performed using a quadtree data structure [3, 4].

The parametrization of the lines by $y = kx + d$ can be numerically problematic if very large values of the slope k are possible. A more robust parametrization of the line has the form $x \cos\varphi + y \sin\varphi - c = 0$. The curve in the Hough space of φ and c passing through the point (x_i, y_i) is a sinusoid with the equation

$$c = x_i \cos\varphi + y_i \sin\varphi = r_i \sin(\varphi + \varphi_i), \quad \text{with}$$
$$r_i = \sqrt{x_i^2 + y_i^2}, \quad \varphi_i = \arctan(x_i / y_i). \tag{5.4}$$

If the curve to be found in the image space is a circle through the origin, there are two possibilities. The problem can be reduced to the straight line case by a conformal transformation, or a circle through the origin can be parameterized in the following form:

$$x^2 - 2x R \cos\varphi + y^2 - 2y R \sin\varphi = 0, \tag{5.5}$$

where R is the circle radius and φ is the azimuth of the circle center in polar coordinates. The curve in the Hough space of $\kappa = 1/R$ and φ passing through the point (x_i, y_i) is a sinusoid with the equation

$$\kappa = \frac{2}{r_i} \sin(\varphi + \varphi_i), \tag{5.6}$$

with r_i and φ_i as above, see Fig. 5.2.

If the curve to be found in the image space is a circle in general position with the equation

$$(x - x_c)^2 + (y - y_c)^2 - R^2 = 0, \tag{5.7}$$

the constraint that the circle passes through the point (x_i, y_i) defines a second order surface in the 3D Hough space of x_c, y_c, z:

$$z = (x_i - x_c)^2 + (y_i - y_c)^2, \quad \text{with } z = R^2. \tag{5.8}$$

It follows that finding circles requires finding intersection points of surfaces in a 3D histogram, which is computationally much more expensive than the same problem in 2D. A search in 3D can be based on octrees, the 3D analogues of quadtrees [4].

An alternative is the randomized Hough transform [5], that randomly selects triplets of points. The center of the circle passing through the triplet and its radius are stored in a 3D histogram. Peak finding can be done in 3D or in the 2D histogram of the circle centers. After finding a peak, the best circle center and radius is obtained by computing the medoid [6] of the entries in the peak bin.

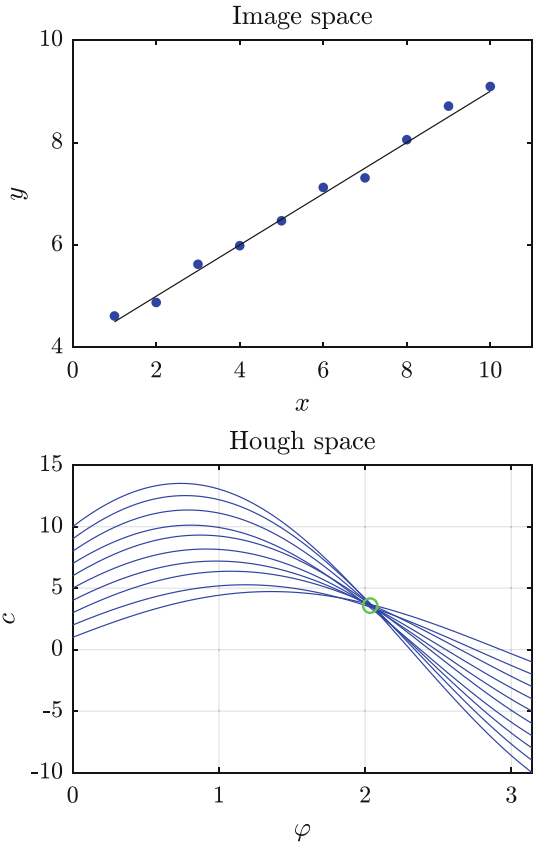

Fig. 5.2 Top: Image space (x, y); bottom: Hough space (φ, c). The circled point in the Hough space corresponds to the straight line in the image space

5.1.3 *Artificial Retina*

The concept of the "Artificial Retina" was introduced in [7]. Similar to the Hough transform, it relies on a partition of the track parameter space into cells. Figure 5.3 shows a simple example with a track in 2D, specified by slope k and intercept d.

The intensity of a cell is the sum of the responses of its associated receptors to the hits that are present in the layer. Assuming a Gaussian response, the intensity $R(k, d)$ of the cell centered at (k, d) is given by:

$$R(k,d) = \sum_{i=1}^{n} \sum_{j=1}^{m} \exp\left(-s_{ij}^2 / 2\sigma_i^2\right), \tag{5.9}$$

where $s_{ij} = y_{ij} - (kx_i + d)$ is the distance of hit j in layer i from the ideal track position in layer i, and σ_i is a scale parameter that regulates the width of the receptive field in layer i. Other response functions are of course possible, and

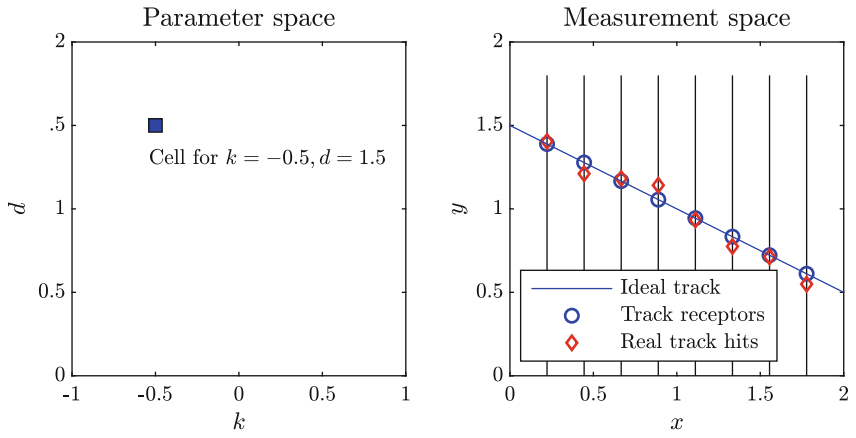

Fig. 5.3 Left: A cell in the parameter space of (k, d) corresponds to an ideal track with these parameters. Right: The corresponding track receptors represent the intercepts of this ideal track with the tracking layers. The hits of a real track are close to the track receptors within the experimental resolution. (Adapted from [11], by permission of Elsevier)

their shape and width can be adjusted for optimal performance. As with the Hough transform, track candidates correspond to the local maxima of intensity in parameter space.

The artificial retina is eminently suitable for high-speed track finding, as it can be highly parallelized and implemented on commercial FPGAs [8, 9]. For applications in the vertex locator of LHCb (Sect. 1.6.1.4) and in a test beam, see [10, 11].

5.1.4 Legendre Transform

The Legendre transform is an extension of the Hough transform, used to find common tangent lines or tangent circles through the origin to a set C of circles [12, 13]. In the context of track finding, the circles in C are drift circles in a drift chamber or a drift tube chamber [14], see Fig. 5.4. Assume a drift circle in the plane, with center $(x_\mathrm{w}, y_\mathrm{w})$ and radius ρ. A line parameterized by $x\cos\varphi + y\sin\varphi - c = 0$ is tangent to the circle if, and only if, its signed distance from the circle center is equal to $\pm\rho$:

$$x_\mathrm{w}\cos\varphi + y_\mathrm{w}\sin\varphi - c = \pm\rho_i \quad \text{or} \quad c = x_\mathrm{w}\cos\varphi + y_\mathrm{w}\sin\varphi \pm \rho_i\,. \tag{5.10}$$

The drift circle in the image space (x, y) therefore corresponds to two sinusoids in the Legendre space (φ, c), see Fig. 5.4. The further procedure is the same as with the Hough transform.

A circle through the origin can be parameterized by its radius R and the angle Φ, where the circle center is given by $x_\mathrm{c} = R\cos\Phi$, $y_\mathrm{c} = R\sin\Phi$. Such a circle

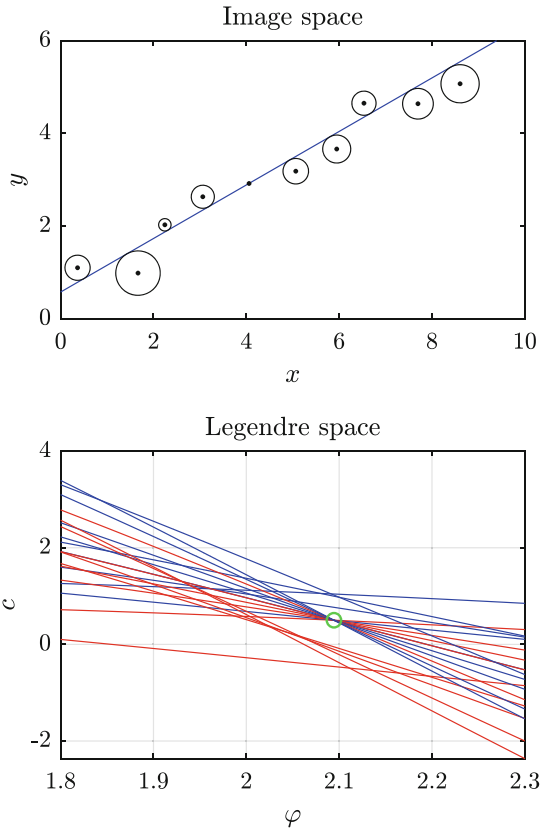

Fig. 5.4 Top: Image space (x, y); bottom: Legendre space (φ, c). The circled point in the Legendre space corresponds to the straight line in the image space

touches the drift circle with center $(x_\mathrm{w}, y_\mathrm{w})$ and radius ρ if, and only if, the squared distance of the circle centers is equal either to $(R + \rho)^2$ or to $(R - \rho)^2$:

$$2R\,(\pm\rho + x_\mathrm{w}\cos\Phi + y_\mathrm{w}\sin\Phi) = x_\mathrm{w}^2 + y_\mathrm{w}^2 - \rho^2. \tag{5.11}$$

In order to avoid large radii in the limit of the circle approaching a straight line, the Legendre space is chosen as (κ, Φ), where $\kappa = 1/R$ is the curvature of the circle. The drift circle again corresponds to two sinusoids in (κ, Φ):

$$\kappa = \frac{2(\pm\rho + x_\mathrm{w}\cos\Phi + y_\mathrm{w}\sin\Phi)}{x_\mathrm{w}^2 + y_\mathrm{w}^2 - \rho^2} \tag{5.12}$$

The task of finding a circle through the origin can be reduced to the task of finding a straight line if the Legendre transform is preceded by a conformal transformation, see Sect. 5.1.1, which transforms the circle into a straight line while transforming the drift circles into circles.

5.1.5 Cellular Automaton

A cellular automaton (CA) is a dynamical system where space, time, and variables are discrete [15]. It has five fundamental defining characteristics [16]:

1. It consists of a discrete lattice of sites.
2. It evolves in discrete time steps.
3. Each site takes on a finite set of possible values.
4. At each site, the value evolves according to the same deterministic rules.
5. The rules for the evolution of a site depend on a local neighbourhood around it.

Probably the best known CA is Conway's "Game of Life" [17]. Like many, but not all, subsequently proposed cellular automata, it assumes that the cells are located on a regular 2D rectangular lattice. However, this assumption is too restrictive for the application to track finding. Instead, the cells are represented by the nodes of a graph, and the neighbourhood of a cell C_i is the set of all cells connected by an edge to C_i.

The neighbours of a cell are divided into "inner" and "outer" neighbours such that if C_i is an inner neighbour of C_j, then C_j is an outer neighbour of C_i. The possible states of a cell are the non-negative integers. In practice, the states are bounded by some upper limit depending on the number of detector layers. The initial states of all cells are set to zero.

The earliest applications of the CA to track finding are described in [18–21]. With the exception of [18], cells are defined as short track segments connecting hits in adjacent detector layers; segments that skip a layer are sometimes allowed as well [22, 23]. Each cell has an inner hit and an outer hit according to the arrangement of the detector layers. Two cells are neighbours if the outer hit of one cell is the inner hit of the other cell. In addition, the neighbourhood relation can be further restricted by imposing a cut on the angle spanned by the two cells (segments). It is the task of the CA to find chains of neighbouring segments that correspond to actual tracks. This is achieved by the following rule of evolution. At each time step, the state of each cell is augmented by 1 if it has the same state as its inner neighbour. The states of all cells are updated synchronously. When no state changes anymore, the evolution is stopped. At this point, the state of a cell is the length of the longest unbroken chain of segments terminating in this cell. An illustration of the CA is shown in Fig. 5.5.

The actual search for track candidates starts with the cells with the highest state. If such a cell has an inner neighbour with a state that is smaller by 1, it is attached to the track candidate. This procedure is repeated until a cell with no inner neighbour is reached. If, at any point, several neighbours can be attached to the track candidate, either the "best" cell according to some criterion is selected, or the track candidate is split into two, and each candidate is followed independently. This inevitably results in candidates that share hits, requiring a final selection of compatible (non-overlapping) candidates. This is the topic of Sect. 5.3.

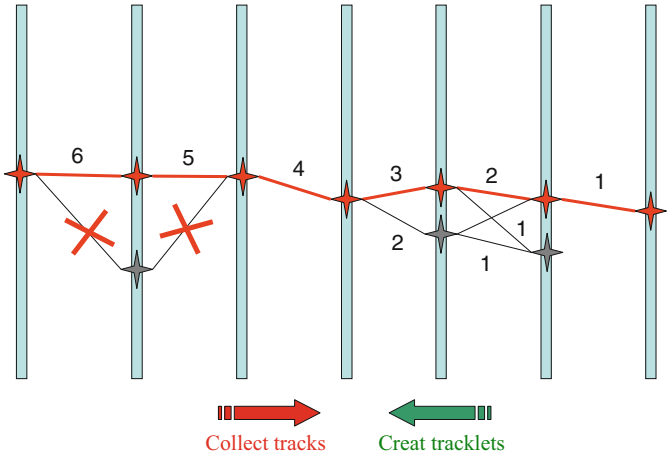

Fig. 5.5 Illustration of the cellular automaton algorithm. It creates tracklets and links, numbers them as possibly situated on the same trajectory, and collects tracklets into track candidates. (From [24], by permission of Elsevier)

In tracking detectors with few layers and little redundancy [25], the CA can be complemented by prior information stored in a sector map [26, 27]. In this concept, the sensors are partitioned into sectors. In order to create a sector map, a large training sample of simulated tracks in a chosen angular and momentum region is generated. Two sectors are declared as "friends" if a sufficiently large number of tracks passes through them without hitting any other sector in between. The friendship relation defines an acyclic directed graph that is stored in the sector map.

In the track-finding phase, the hits are sorted into their sectors, and the segment finder is activated. It creates pairs of hits in friendly sectors and keeps those pairs (segments) that pass various cuts, which are also stored in the sector map. If two segments share a hit such that the outer hit of the inner segment is the same as the inner hit of the outer segment, they are passed to the neighbour finder, which applies additional cuts, also stored in the sector map. Finally, all surviving segments form the cells of the CA.

Different sector maps for different geometrical or kinematical regions can be created and applied sequentially. For instance, high-momentum, high-quality tracks can be found first by using a sector map with tighter cuts. After removing their hits, a second sector map with less selective cuts can be used to find the remaining tracks.

Another extension of the CA, termed the 4D CA, is described in [28, 29]. Pairs of segments or triplets of hits are accepted only if the time stamps of the hits are consistent with the hypothesis that they have been created by the same charged particle. For the application of the 4D CA in the CBM experiment, see Sect. 11.2.

5.1.6 *Neural Networks*

The application of neural networks to track finding was first proposed independently in [30] and [31]. In this approach, the network is of the Hopfield type [32], the neurons being track segments that connect observations in adjacent or nearby layers of the detector; see Sect. 5.1.6.1. More recently, the HEP.TrkX pilot project was established with the aim to develop deep neural networks for track finding in high-multiplicity environments typical for the LHC era [33]. Two deep networks are described in Sects. 5.1.6.2–5.1.6.3. A follow-up project, called Exa.TrkX, was started with a kick-off meeting in June 2019 [34]; a second workshop was held in April 2020 [35]. A reference to first published results can be found in Sect. 5.1.6.3.

5.1.6.1 Hopfield Network

A Hopfield network is a fully connected network with a single layer of neurons. In the simplest case, the neurons are binary with two states: $s_i = \pm 1,\ i = 1, \ldots, n$. Each pair (i, j) of neurons has a fixed connection weight w_{ij} with $w_{ij} = w_{ji}$ and $w_{ii} = 0$. The states of the neurons evolve in discrete time steps according to the following prescription:

$$s_i(t) = \text{sign}\left[\sum_{j=1}^{n} w_{ij}\, s_j(t-1)\right]. \tag{5.13}$$

The update can be synchronous (the states are recomputed in parallel) or asynchronous (the states are recomputed sequentially). The network has an associated function $E(\boldsymbol{s})$, defined as:

$$E(\boldsymbol{s}) = -\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\, s_i\, s_j, \tag{5.14}$$

where $\boldsymbol{s} = (s_1, \ldots, s_n)$ is the state of the network. In analogy to the theory of spin glasses, $E(\boldsymbol{s})$ is called the energy function of the network. It can be shown that $E(\boldsymbol{s})$ is a non-increasing function of the time t and that the update rule Eq. (5.13) leads to a local minimum of $E(\boldsymbol{s})$ [36].

In most applications, including the one discussed here, the aim is to find the global minimum rather than a local one. To this end, thermal noise is introduced in the network. At temperature T, the state $\boldsymbol{s}$ is Boltzmann distributed with the probability function

$$P(\boldsymbol{s}) = \frac{1}{Z}\exp\left[-E(\boldsymbol{s})/T\right], \quad \text{with} \quad Z = \sum_{\boldsymbol{s}} \exp\left[-E(\boldsymbol{s})/T\right]. \tag{5.15}$$

As the number of possible states rises exponentially with the number of neurons, the partition function Z is computed in the mean-field approximation [37], and the thermal average v_i of s_i is given by:

$$v_i = \langle s_i \rangle_T = \tanh\left(-\frac{1}{T}\,\frac{\partial E}{\partial v_i}\right), \tag{5.16}$$

where the states $\boldsymbol{v} = (v_1, \ldots, v_n)$ are now continuous in the interval $(-1, 1)$. The definition of the energy function is analogous to Eq. (5.14):

$$E(\boldsymbol{v}) = -\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\, v_i\, v_j, \tag{5.17}$$

and the update is modified accordingly:

$$v_i(t) = \tanh\left[\frac{1}{T}\sum_{j=1}^{n} w_{ij}\, v_j(t-1)\right]. \tag{5.18}$$

Finding the global minimum or at least a low local minimum of the energy function is facilitated by deterministic annealing [38]. First, the energy function is minimized at high temperature; the temperature is then lowered according to a predefined cooling or annealing schedule. At low temperature, the states of the network are close to either 1 (active) or -1 (inactive).

For the purpose of track finding, the problem has to be mapped on the Hopfield network such that the final state of the network corresponds to the solution of the problem. Similar to the CA, the neurons are short track segments connecting space points in adjacent or nearby layers of the tracking detector. To keep the number of neurons manageable, geometric cuts ensure that only segments that can be part of an actual track in the momentum range of interest are used as neurons. The sector map introduced in Sect. 5.1.5 can be used to store the cuts. A track can be considered as an unbroken chain of segments, so only pairs of segments sharing a point qualify for having a positive weight. Consider a triple of points (k, l, m) in consecutive layers, defining two neurons v_{kl} and v_{lm}. The weight w_{klm} depends on the angle between the segments and on their length. In [31], it is defined as follows:

$$w_{klm} = \frac{\cos^{\lambda}\theta_{klm}}{d_{kl} + d_{lm}}, \tag{5.19}$$

where λ is an odd exponent and d_{ij} is the length of neuron v_{ij}, either in space or in the projection to the bending plane. This definition of the weights favours combinations of short segments with small angles. Small weights can be set to zero.

Constraints on the possible final configurations of the active neurons can be included by adding a cost or penalty term to the energy function of the network. This

serves to prevent association of a point to several tracks and to get approximately the expected number of active neurons in the final state. For example, in [39] the following cost term was used:

$$C(\boldsymbol{v}) = \alpha \left(\sum_{k,l,n,l \neq n} v_{kl} v_{kn} + \sum_{k,l,m,k \neq m} v_{kl} v_{ml} \right) - b, \tag{5.20}$$

where α is a Lagrange multiplier and b is a small constant bias term. Minimizing the first term leads to a competition between neurons starting or ending in the same point, so that in the end at most one should survive. The final update equation is given by Eq. (5.16).

For an evaluation of the performance of the Hopfield network on real data from an experiment at the LEP electron-positron collider, see [39]. For an application to tracking in the ALICE experiment at the LHC, see [40].

5.1.6.2 Recurrent Neural Network

The dynamics of a particle track can be modelled by a nonlinear state-space model; see Sect. 3.2.3. A track is, thus, very similar to a discrete time series, the main difference being that the observations are not progressing in time but in space. Recurrent neural networks (RNNs), in particular networks of long short-term memory (LSTM) neurons [41, 42], are routinely used for forecasting time series arising in various contexts: financial, industrial, weather, traffic, etc. It is, therefore, to be expected that RNNs can be trained to learn the dynamics of a track and follow it in a way similar to the extended Kalman filter. The crucial difference is that in the Kalman filter, the dynamics is explicitly coded in the system equation and in the distribution of the process noise, whereas in the RNN, the dynamics is implicit in the weights learned by the network on a training set of examples.

The first successful attempt to train an RNN for track finding is described in [43]. Two models are presented. The first is a sequential hit predictor that, given a sequence of past hits, predicts the position of the next hit. The second model augments the first one by predicting the covariance matrix of the hit, using a Gaussian distribution. Both predictors are implemented as an LSTM layer, followed by a fully connected (FC) layer. The scheme of the Gaussian predictor is shown in Fig. 5.6. The training data for the RNN network and the GNN network in Sect. 5.1.6.3 were generated by the ACTS package [44].

5.1.6.3 Graph Neural Network

Like the CA described in Sect. 5.1.5, track finding with the graph neural network (GNN) is based on the representation of the tracking data by a graph [43]. The detector hits are the nodes (vertices) of the graph, and two nodes are connected by

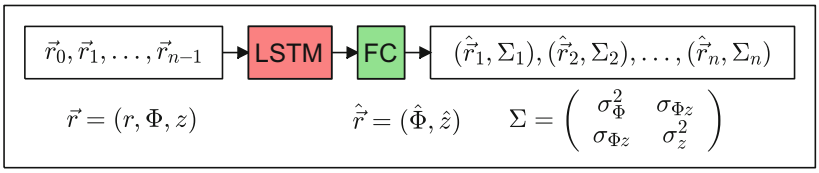

Fig. 5.6 Diagram of the Gaussian hit predictor model that takes a sequence of 3D coordinates as input and produces bivariate Gaussian probability distributions as next-step predictions. The architecture is the same as the basic hit predictor, but the model provides additional output that parameterizes the Gaussian covariance matrix. (Adapted from [43], by permission of the author)

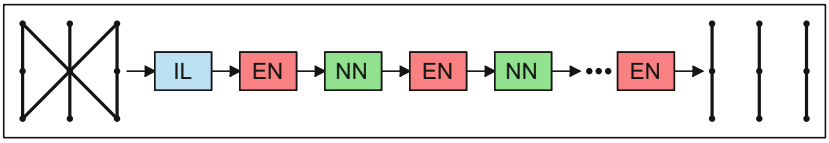

Fig. 5.7 Diagram of the Graph Neural Network model which begins with an input transformation layer (IL) and has a number of recurrent iterations of alternating EdgeNet (EN) and NodeNet (NN) units. In this case, the final unit is an EN, making this a segment classifier model. (Adapted from [43], by permission of the author)

an edge if they are compatible according to some criterion. Such a criterion can be stored in a sector map as the one described in Sect. 5.1.5. The graph is fed into the GNN that consists of an input transformation layer (IL) followed by alternating units of edge networks (ENs) and node networks (NNs), each implemented as a multi-layer perceptron with two layers, see Fig. 5.7.

An EN computes a new weight for every edge from the features of the end nodes while an NN computes new features for every node from the current features and the edge-weighted aggregated features of all connected nodes in the adjacent layers. The network can be used to classify whether nodes/hits or edges belong to a track or not. The network in Fig. 5.7 classifies edges, as the last unit is an EN. An extension of the work in [43] to track seeding and hit labelling with GNNs is described in [45].

5.1.7 Track Following and the Combinatorial Kalman Filter

In track finding methods such as the Hough transform or the CA, the complete set of all hits serves as the primary input. Such methods have, therefore, been dubbed "global" [46]. This is in contrast to methods that find tracks locally or sequentially, one after the other. The most prominent example of a sequential method is track following.

In track following, a track candidate starts from a "seed", i.e., a short track segment. This seed can in principle be anywhere in the tracking detector. Generating seeds in the outer layers of the trackers has the advantage of smaller occupancy and

less background from low-momentum tracks that spiral in the inner layers of the tracker. Generating seeds in the inner layers, which are frequently pixel layers, has the advantage of using 3D hits with higher resolution both in the bending plane and in the longitudinal direction. As will be described in more detail in Chap. 10, ATLAS and CMS, the two general-purpose experiments at the LHC, have opted for the second solution.

The generation of seeds is often a simple combinatorial search for compatible triplets or quadruplets of hits, potentially assisted by a CA [47], and includes information about the size and position of the beam spot; see Sect. 7.1. Some examples of seed generation algorithms will be given in Chap. 10.

Once the seeds have been found, each seed is then followed through the tracker by extrapolating it toward the outside of the tracker or toward the production region, depending on where the seed is situated. After each extrapolation step, compatible hits are searched for and attached to the track candidate.

The progressive track recognition described in [48] can be extended to a combinatorial Kalman filter (CKF), introduced in [49, 50] under the name "Concurrent Track Evolution", see Fig. 5.8. First, each seed is fitted with one of the methods described in Chap. 6. The parameters and the covariance matrix of the seed are then extrapolated to the nearest tracker layer, taking into account interactions with the detector material; see Sects. 4.3, 4.4 and 4.5. The hits in the sensor in which the extrapolated trajectory intersects with the layer are tested for compatibility with the predicted track parameters using a chi-square statistic; see Sects. 3.2.3 and 6.1.2. If n compatible hits are found, n copies of the predicted state, i.e., its track parameters and its covariance matrix, are generated, and each one of them is updated with one of the n hits according to the Kalman filter, Eqs. (3.29) and (3.30) or Eqs. (3.31) and (3.32). The original state is also kept and marked as having a missing hit, giving a

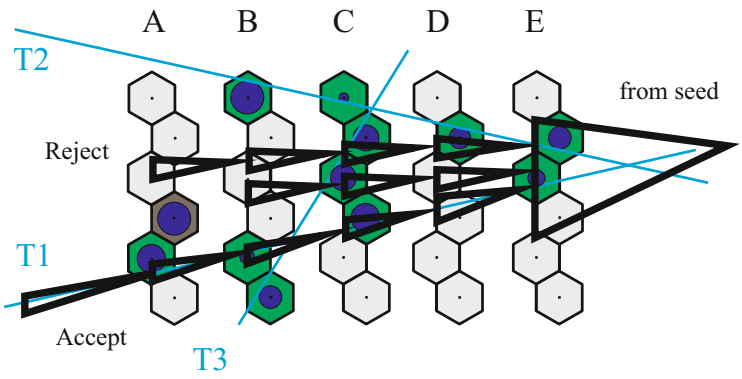

Fig. 5.8 Schematic view of concurrent track evolution in a five-layered part of a tracking system with hexagonal drift cells, which is traversed by three particles, labelled T1, T2 and T3. The simulated drift time isochrones are indicated by circles. The propagation proceeds upstream from the right to the left and starts with a seed of hits from track T1 outside of the picture. (From [49], by permission of Elsevier)

total of $n + 1$ track candidates. This procedure is iterated on each track candidate until the last layer of the tracker is reached or the count of missing hits in a candidate exceeds a preset threshold, typically one or two.

In the course of the combinatorial Kalman filter, it may be necessary to limit the number of active candidates for reasons of memory and/or speed. In this case, the "worst" track candidates are discarded and not followed anymore. The quality of a track candidate can be measured by a combination of its total number of hits, its number of missing hits, and its total chi-square χ^2_{tot} (Sect. 6.4). The tuning of the combination is usually performed on simulated data, where the correct association of hits to tracks is known.

Avoiding a combinatorial explosion is an important issue in experiments with high track multiplicities. Therefore, the CKF in, for instance, ATLAS and CMS starts with seeds in the pixel detector with its very high resolution in all three spatial dimensions. As a consequence, the compatibility test in the first non-pixel layer rejects wrong hits with a high probability. As the CKF proceeds, the state becomes more precisely known, and the probability of attaching a wrong hit becomes even smaller. For more details, see Sects. 10.2 and 10.3.

5.1.8 Pattern Matching

Pattern or template matching is mostly used in real-time track finding for the purpose of triggering on charged tracks (see Sect. 5.2). It can be applied to detectors with a layer structure, each layer being segmented into sectors or bins [51, 52]; see also Sect. 5.1.5. A charged particle crossing the detector generates hits in certain bins of the layers, thereby creating a pattern of "on" and "off" bins, which can be coded as strings of zeros and ones. The set of physically meaningful patterns is generated by extensive simulations and stored in a pattern bank.

The number of patterns to be stored depends on the geometry and size of the detector, on the characteristics of the tracks to be found, and on the granularity of the binning. For the purpose of triggering a lower bound on the momentum is usually imposed; therefore, only the patterns of tracks with momentum above the threshold need to be stored. The granularity of the binning determines how well two nearby tracks can be separated, and therefore depends on the occupancy of the layer. If the binning is very coarse, for instance, only one bin per module in a silicon strip tracker, fewer patterns have to be stored, but two nearby tracks cannot be resolved (see Fig.1 in [52]). If the binning is very fine, for instance, a bin for each strip in the extreme case, nearby tracks can be resolved almost perfectly, but the number of patterns is far too large to store. Also, higher track multiplicity implies smaller track separation on average, which in turn requires finer granularity. In any case, the granularity has to be optimized by extensive simulation studies.

In an event, a particular configuration of hits is generated and translated into a pattern. This pattern is compared to all patterns in the bank, and matching patterns constitute track candidates. The sketch in Fig. 5.9 shows an example of a pattern

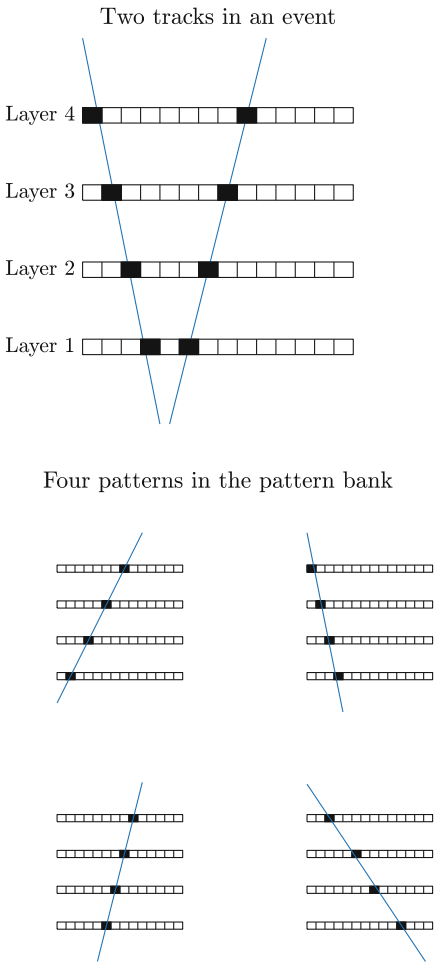

Fig. 5.9 Top: Two tracks in a detector with four layers creating two patterns. Bottom: Four patterns in the pattern bank

generated by two tracks and some patterns stored in the pattern bank. To cope with inefficiencies of the tracking detector, it may be necessary to accept partial matches.

As the number of patterns that have to be stored can be very large, the method is feasible only if there is sufficient memory and if the comparison can be made very fast and in a highly parallel mode. The matching has therefore to be implemented in VLSI hardware, using content-addressable or associative memories [51, 52]. The pattern can be arranged in a tree structure, starting with coarse granularity of the sectors and proceeding to finer granularity. It is also possible to store patterns with variable resolution [53].

Pattern matching was used for real-time track finding both in the vertex detector and in the drift chamber of the CDF experiment; see Sect. 5.2.1. Later applications include the FTK (Fast Track Trigger) for the ATLAS experiment, see Sect. 5.2.2, and a proposed track trigger for the new CMS tracker that will be installed for the HL-LHC, see Sect. 5.2.3.

5.2 Online Track Finding

An early proposal for online track finding by dedicated hardware is the one described in [52]. It is based on matching hit patterns in the tracking detector with a pattern bank stored in associative memory; see Sect. 5.1.8. As field programmable gate arrays (FPGAs) were still in their infancy at the time of the publication, the associative memory (AM) is an array of 400 custom VLSI chips [51] that can hold $O(10^5)$ patterns [51]. The pattern matching is organized as a tree search through different levels of spatial resolution [52]. This was soon followed by the actual implementation in the CDF experiment at the Tevatron collider.

5.2.1 CDF Vertex Trigger

The Silicon Vertex Tracker (SVT, [54]) was designed to provide track impact parameter information for the level-2 trigger of the CDF experiment [55]. It was realized in custom hardware [56, 57]. Track finding is done by an AM, refining the information from the XFT track processor (see below) that finds tracks in the central drift chamber for the level-1 trigger. Tracks are fitted by a farm of digital signal processors using a linearized fit that requires only scalar products; see Sect. 6.1.6.

The upgraded SVT, now renamed Silicon Vertex Trigger, is described in [58, 59]. If an event passes the level-1 trigger, the SVT extrapolates the XFT tracks, associates hits in the silicon vertex detector, and computes the transverse impact parameter. Its average latency is 24 μs. The hit association is performed by custom AM chips, the linearized track fit in FPGAs.

The eXtremely Fast Tracker (XFT, [60, 61]) is a track processor that finds tracks with high transverse momentum in the central drift chamber of the experiment [62]. It is highly parallel and reports its results every 132 ns, in time for the trigger level-1 decision. The XFT works with hits in the four axial superlayers. Track identification is done in two stages called the Finder and the Linker. The Finder searches for high-p_T segments in each of the superlayers, and the Linker searches for high-p_T track candidates by combining segments from at least three (out of four possible) segments. Both stages use pattern matching to accomplish their tasks.

5.2.2 ATLAS Fast Tracker

The Fast Tracker (FTK) system of the ATLAS experiment is designed for global track reconstruction after each level-1 trigger [63, 64]. It enables the level-2 trigger to gain rapid access to tracking results. The system is based on the Silicon Vertex Trigger of the CDF experiment; see the preceding subsection. It uses hit data from four pixel layers and from both sides of four silicon strip layers, twelve in total. The

tracker volume is split into 64 regions or towers, which are processed independently. The sensors are divided into "superstrips" with a coarser resolution.

Data processing starts with clustering in the pixel and in the strip sensors. The clustering algorithm is optimized for execution in an FPGA [65]. After clustering, a track is represented by a list of superstrips that corresponds to a pattern in the custom AM chip [66]. Pattern matching produces track candidates at coarse resolution; these are then refined by a high resolution track fit in an FPGA. Missing layers are allowed in both stages. The number of patterns that have to be stored is currently in the order of a billion. This number would have to rise by a factor of the order of 10 at the HL-LHC [63], because of the larger number of channels in the tracker and of the higher track multiplicity, which requires finer granularity (see Sect. 5.1.8). For this and other reasons, the FTK will not be upgraded for operation at the HL-LHC; instead, the focus will be on the acceleration of the track reconstruction software [67, 68]; see also Sect. 10.2.

The fitted tracks are sent to the Second Stage Board wherein they are extrapolated to the remaining silicon layers and fitted again. Finally, duplicate tracks are removed based on the number of common hits and χ^2 [64].

5.2.3 CMS Track Trigger

Starting in 2026, the luminosity of the LHC is expected to increase by a factor of about ten above the current design value. The current CMS silicon tracker, having been in operation since 2009, cannot withstand the radiation level predicted for the HL-LHC and has no triggering capability. It will, therefore, be replaced by a newly designed tracker [69]. The new design features so-called p_T modules as the basic sensing devices. A p_T module consists of two closely stacked sensors, either a pixel and a strip sensor (PS) or two strip sensors (2S); see [70] and Fig. 5.10. A charged particle crossing the stack generates a stub that consists of two clusters. Tracks of sufficiently high transverse momentum ($p_T > 2\,\text{GeV}$) have little curvature and a small offset in the sensor stack, in contrast to tracks with smaller p_T, which are bent

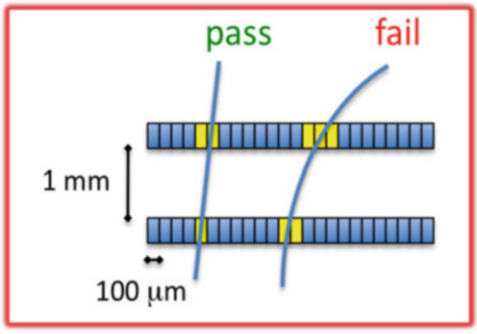

Fig. 5.10 A p_T module of the new CMS tracker. High-momentum tracks pass the p_T cut, low-momentum tracks fail. (From [70], reproduced under License CC-BY-3.0)

more strongly and have a larger offset. Stubs that pass the cut on p_T are the input to the track trigger.

Three concepts are explored for reconstructing tracks at the level-1 trigger, two using an all-FPGA system [71], the third one using a combination of AM and FPGAs [72].

5.2.3.1 Time Multiplexing

The all-FPGA system is based on the principle of time multiplexing [73]. The fundamental idea is that several sources send their information from a given bunch crossing to a single destination for processing. The architecture of the system has two layers: the first extracts and preprocesses the stubs and sends them to the second layer, which contains the track finding processors.

Two track finding algorithms are investigated in the time-multiplexed track trigger, tracklets and Hough transform. The tracklet algorithm has the following stages:

1. *Stub organization:* The stubs are sorted into sectors in ϕ.
2. *Seeding:* Tracklets are formed from stubs in adjacent layers.
3. *Projection:* Tracklets are projected to other layers to search for matching stubs, both inside-out and outside-in.
4. *Fit:* Track fit of stubs matched to the tracklet.
5. *Duplicate removal:* Candidate selection based on χ^2.

The second algorithm [71] has the following stages:

1. *Hough transform:* Stubs on the same trajectory are transformed into lines that meet in the vicinity of a single point; see Sect. 5.1.2.
2. *Fit:* Combinatorial Kalman filter, see Sect. 5.1.7.
3. *Duplicate Removal:* Tracks are removed whose parameters do not correspond to the bin of the Hough transform where they were found.

5.2.3.2 Pattern Matching

Pattern matching is done in parallel in 48 regions called trigger towers [72]. Each tower has two boards for pattern recognition and track fitting. Pattern recognition is done with lower resolution data, which are compared to predefined patterns by a content addressable memory. Only patterns corresponding to tracks with $p_T > 2\,\text{GeV}$ are stored. If a match is found, the corresponding high-resolution data are sent to the track fitting module. The linearized track fit (Sect. 6.1.6) runs in the FPGA and computes the helix parameters and the χ^2.

5.3 Candidate Selection

After track finding, track candidates may share hits. If two candidates share more hits than is deemed acceptable, for instance more than one, the track candidates are called incompatible. The incompatibility relation can be represented by an undirected graph (V, E), where the n vertices $v_i \in V,\ i = 1, \ldots, n$ are the track candidates. Two incompatible track candidates v_i and v_j are connected by the edge $e_{ij} = e_{ji}$, which is defined as the unordered pair (v_i, v_j). The number of compatible track candidates can be maximized by finding an independent set of vertices of maximal size, i.e., a subset $V_1 \subseteq V$ of vertices, no two of which are connected by an edge.

Alternatively, the graph can represent the compatibility relation, in which case two compatible tracks/vertices are connected by an edge. The problem is then to find a maximum clique, i.e., a fully connected subset $V_2 \subseteq V$ of vertices of maximal size.

Both problems are NP-hard [74] so that finding a maximum independent set or maximum clique can be very time-consuming for large graphs. A set of C routines for finding cliques in a compatibility graph can be found in [75]. The fastest exact algorithm for finding independent sets published up to now is the one in [76]. An independent set can also be obtained by finding a vertex cover, i.e., a set $V_3 \subseteq V$ of vertices the removal of which leaves an independent set. In [77], it was shown that there is a one-to-one correspondence between minimal vertex covers or maximal independent sets and steady states of Hopfield networks [32] with nonpositive weights. In addition, such a network converges to its steady state in at most $2n$ steps. There may be many minimal vertex covers of different size, and the steady state reached depends on the initial state of the network, of which there are 2^n. Finding the minimal vertex cover by an exhaustive or random search of all initial states is, therefore, computationally infeasible for large n.

In any case, finding the largest independent set is not necessarily the best approach for finding an "optimal" set of track candidates, as the quality of the track candidates (see Sect. 6.4) should be taken into account, too. If the quality of the candidate v_i is quantified by a positive weight w_i, the problem is now to find an independent set that maximizes the sum of weights (MWIS). Like its unweighted counterpart, the MWIS problem is NP-hard. For a recent approximative solution and numerous references to previous work, see [78].

If the weight w_i is mapped to a quality indicator $q_i \in [0, 1]$, the network in [77] can be generalized to a recurrent network with annealing that aims to find the set of compatible vertices with the largest sum of weights [79]; see also Sect. 11.1.

References

1. M. Hansroul, H. Jeremie, D. Savard, Nucl. Instrum. Methods Phys. Res. A **270**(2), 498 (1988)
2. P. Hough, in *Proceedings of the International Conference on High Energy Accelerators and Instrumentation* (CERN, Geneva, 1959), p. 554. http://inspirehep.net/record/919922
3. V. Trusov, Development of Pattern Recognition Algorithms for the Central Drift Chamber of the Belle II Detector. Ph.D. thesis, KIT, Karlsruhe (2016). https://publikationen.bibliothek.kit.edu/1000063754
4. S. Aluru, in *Handbook of Data Structures and Applications*, chap. 20, ed. by D. Mehta, S. Sahni, 2nd edn. (Chapman & Hall/CRC, Boca Raton, 2018)
5. L. Xu, E. Oja, P. Kultanen, Pattern Recogn. Lett. **11**(5), 331 (1990)
6. A. Struyf, M. Hubert, P. Rousseeuw, J. Stat. Softw. **1**(4), 1 (1997). https://www.jstatsoft.org/v001/i04
7. L. Ristori, Nucl. Instrum. Methods Phys. Res. A **453**(1), 425 (2000)
8. N. Neri et al., PoS **TIPP2014**, 209 (2015). https://pos.sissa.it/213/199
9. A. Abba et al., Nucl. Instrum. Methods Phys. Res. A **824**, 260 (2016)
10. A. Abba et al., Nucl. Part. Phys. Proc. **273–275**, 2488 (2016)
11. N. Neri et al., Nucl. Instrum. Methods Phys. Res. A **845**, 607 (2017)
12. D. Primor, O. Kortner, G. Mikenberg, H. Messer, J. Instrum. **2**(01), P01009 (2007). https://doi.org/10.1088/1748-0221/2/01/P01009
13. T. Alexopoulos, M. Bachtis, E. Gazis, G. Tsipolitis, Nucl. Instrum. Methods Phys. Res. A **592**, 456 (2008)
14. L. Rolandi, W. Riegler, W.Blum, *Particle Detection with Drift Chambers*. Particle Acceleration and Detection (Springer, Berlin/Heidelberg, 2008)
15. G.Y. Vichniac, Phys. D: Nonlinear Phenom. **10**(1), 96 (1984)
16. S. Wolfram, Phys. D: Nonlinear Phenom. **10**(1), vii (1984)
17. M. Gardner, Sci. Am. **223**(10), 120 (1970). https://web.stanford.edu/class/sts145/Library/life.pdf
18. A. Glazov, I. Kisel, E. Konotopskaya, G. Ososkov, Nucl. Instrum. Methods Phys. Res. A **329**, 262 (1993)
19. M.P. Bussa et al., Nuovo Cimento **109**(3), 327 (1996)
20. M.P. Bussa et al., Comput. Math. Appl. **34**(7), 695 (1997)
21. I. Kisel et al., Nucl. Instrum. Methods Phys. Res. A **387**, 433 (1997)
22. I. Abt et al., Nucl. Instrum. Methods Phys. Res. A **489**(1), 389 (2002)
23. I. Abt et al., Nucl. Instrum. Methods Phys. Res. A **490**(3), 546 (2002)
24. I. Kisel, Nucl. Instrum. Methods Phys. Res. A **566**(1), 85 (2006)
25. R. Frühwirth et al., Nucl. Instrum. Methods Phys. Res. A **732**, 95 (2013)
26. J. Lettenbichler, Real-time pattern recognition in the central tracking detector of the Belle II Experiment. Ph.D. thesis, TU Wien (2016). http://repositum.tuwien.ac.at/obvutwhs/download/pdf/1656041
27. T. Bilka et al., EPJ Web Conf. **150**, 00007 (2017). https://doi.org/10.1051/epjconf/201715000007
28. V. Akishina, I. Kisel, J. Phys.: Conf. Ser. **599**(1), 012024 (2015). https://doi.org/10.1088/1742-6596/599/1/012024
29. V. Akishina, I. Kisel, EPJ Web Conf. **127**, 00003 (2016). https://doi.org/10.1051/epjconf/201612700003
30. B.H. Denby, Comput. Phys. Commun. **49**, 429 (1988)
31. C. Peterson, Nucl. Instrum. Methods Phys. Res. A **279**, 537 (1989)
32. J.J. Hopfield, Proc. Nat. Acad. Sci. **79**, 2554 (1982). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238/pdf/pnas00447-0135.pdf
33. S. Farrell et al., EPJ Web Conf. **150**, 00003 (2017). https://doi.org/10.1051/epjconf/201715000003
34. Exa.TrkX kickoff meeting (2019). https://indico.physics.lbl.gov/event/898/

35. Exa.TrkX Collaboration Meeting (2020). https://indico.fnal.gov/event/23974/
36. J.J. Hopfield, D. Tank, Biol. Cybern. **52**, 141 (1985)
37. C. Peterson, J.R. Anderson, Complex Syst. **2**, 995 (1987)
38. C. Peterson, B. Söderberg, Neural Optimization. Technical Report LU TP 02-30, Lund University (2002). https://tinyurl.com/Peterson-NeuralOptimization
39. G. Stimpfl-Abele, L. Garrido, Comput. Phys. Commun. **64**, 46 (1991)
40. A. Pulvirenti et al., Nucl. Instrum. Methods Phys. Res. A **533**, 543 (2004)
41. S. Hochreiter, J. Schmidhuber, Neural Comput. **9**(8), 1735 (1997). http://tinyurl.com/Long-Short-Term-Memory
42. F. Gers, J. Schmidhuber, F. Cummins, Neural Comput. **12**(10), 2451 (2000)
43. S. Farrell et al., in *4th International Workshop Connecting the Dots 2018 (CTD2018)* Seattle, Washington (2018). https://arxiv.org/abs/1810.06111
44. A. Salzburger et al., A Common Tracking Software. https://acts.web.cern.ch/ACTS
45. N. Choma et al., Track Seeding and Labelling with Embedded-Space Graph Neural Networks (2020). https://arxiv.org/pdf/2007.00149.pdf
46. A. Strandlie, R. Frühwirth, Rev. Mod. Phys. **82**, 1419 (2010)
47. F. Pantaleo, New Track Seeding Techniques for the CMS Experiment. Ph.D. thesis, Universität Hamburg (2017). https://cds.cern.ch/record/2293435
48. P. Billoir, Comput. Phys. Commun. **57**, 390 (1989)
49. R. Mankel, Nucl. Instrum. Methods Phys. Res. A **395**(2), 169 (1997)
50. R. Mankel, A. Spiridonov, Nucl. Instrum. Methods Phys. Res. A **426**(2), 268 (1999)
51. M. Dell'Orso, L. Ristori, Nucl. Instrum. Methods Phys. Res. A **278**(2), 436 (1989)
52. M. Dell'Orso, L. Ristori, Nucl. Instrum. Methods Phys. Res. A **287**(3), 436 (1990)
53. A. Annovi et al., in *Proceedings of the 2nd International Conference on Advancements in Nuclear Instrumentation, Measurement Methods and Their Applications* (2011). https://cds.cern.ch/record/1352152
54. S. Belforte et al., IEEE Trans. Nucl. Sci. **42**(4), 860 (1995)
55. F. Abe et al., Nucl. Instrum. Methods Phys. Res. A **271**(3), 387 (1988)
56. S.R. Amendolia et al., IEEE Trans. Nucl. Sci. **39**(4), 795 (1992)
57. S.R. Amendolia et al., Nucl. Instrum. Methods Phys. Res. A **315**(1–3), 446 (1992)
58. B. Ashmanskas et al., Nucl. Instrum. Methods Phys. Res. A **518**(1), 532 (2004)
59. J. Adelman et al., Nucl. Instrum. Methods Phys. Res. A **572**(1), 361 (2007)
60. C. Ciobanu et al., IEEE Trans. Nucl. Sci. **46**(4), 933 (1999)
61. E.J. Thomson, IEEE Trans. Nucl. Sci. **49**(3), 1063 (2002)
62. T. Affolder et al., Nucl. Instrum. Methods Phys. Res. A **526**(3), 249 (2004)
63. M. Shochet et al., Fast TracKer (FTK) Technical Design Report. Technical Report ATLAS-TDR-021, CERN, Geneva (2013). http://cds.cern.ch/record/1552953
64. V. Cavaliere et al., J. Instrum. **11**(02), C02056 (2016). https://doi.org/10.1088/1748-0221/11/02/C02056
65. A. Annovi, M. Beretta, Nucl. Instrum. Methods Phys. Res. A **617**(1), 254 (2010)
66. A. Annovi et al., Associative Memory Design for the FastTrack Processor (FTK) at ATLAS. Technical Report ATL-DAQ-PROC-2011-045, CERN, Geneva (2011). https://cds.cern.ch/record/1402465
67. The ATLAS Collaboration, Fast Track Reconstruction for HL-LHC. Technical Report ATL-PHYS-PUB-2019-041, CERN, Geneva (2019). http://cds.cern.ch/record/2693670/files/ATL-PHYS-PUB-2019-041.pdf
68. F. Klimpel (On behalf of the ATLAS collaboration), Fast Tracking for the HL-LHC ATLAS Detector. Technical Report ATL-PHYS-PROC-2020-044, CERN, Geneva (2020). URL: https://cds.cern.ch/record/2721751
69. S. Mersi (On behalf of the CMS collaboration), Nucl. Part. Phys. Proc. **273–275**, 1034 (2016)
70. M. Jeitler (For the CMS collaboration), J. Instrum. **9**(08), C08002 (2014). https://doi.org/10.1088/1748-0221/9/08/C08002
71. R. Aggleton et al., J. Instrum. **12**(12), P12019 (2017). https://doi.org/10.1088/1748-0221/12/12/P12019

72. G. Fedi (On behalf of the CMS collaboration), EPJ Web Conf. **127**, 00008 (2016). https://doi.org/10.1051/epjconf/201612700008
73. G. Hall (On behalf of the CMS TMTT Team), Nucl. Instrum. Methods Phys. Res. A **824**, 292 (2016)
74. M. Sipser, *Introduction to the Theory of Computation*, 2nd edn. (Thomson, Boston, 2006)
75. S. Niskanen, P. Östergård, Cliquer. https://tinyurl.com/MaximumClique
76. M. Xiao, H. Nagamochi, Inf. Comput. **255**, 126 (2017)
77. Y. Shrivastava, S. Dasgupta, S.M. Reddy, IEEE Trans. Neural Netw. **3**(6), 951 (1992)
78. P. Du, Y. Zhang, Math. Probl. Eng. **2016** (2016). http://downloads.hindawi.com/journals/mpe/2016/9790629.pdf
79. R. Frühwirth, Comput. Phys. Commun. **78**(1), 23 (1993)

Chapter 6
Track Fitting

Abstract Track fitting is an application of established statistical estimation procedures with well-known properties. For a long time, estimators based on the least-squares principle were—with some notable exceptions—the principal methods for track fitting. More recently, robust and adaptive methods have found their way into the reconstruction programs. The first section of the chapter presents least-squares regression, the extended Kalman filter, regression with breakpoints, general broken lines and the triplet fit. The following section discusses robust regression by the M-estimator, the deterministic annealing filter, and the Gaussian-sum filter for electron reconstruction. The next section deals with linearized fits of space points to circles and helices. The chapter concludes with a section on track quality and shows how to test the track hypothesis, how to detect outliers, and how to find kinks in a track.

6.1 Least-Squares Fitting

In this section, three methods are described that are based on the least-squares (LS) principle for estimation of the track parameters. They are linear or linearized regression, the (extended) Kalman filter, and regression with breakpoints. In the case of a strictly linear model, they are mathematically equivalent. With a nonlinear model, there may be small differences because of different choices of the expansion point(s). In the following, the more frequent case of nonlinear models will be described, which contains the linear model as a special case.

6.1.1 Least-Squares Regression

Assume that track finding has produced a track candidate, i.e., a collection of n measurements $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n$ in different layers of the tracking detector, along with their respective covariance matrices $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_n$. The measurements may have

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_6

different dimensions m_i and usually have different covariance matrices, resulting in a heteroskedastic model. The initial parameters of the track to be fitted to the measurements are denoted by $\boldsymbol{p}$. They are assumed to be tied to a reference surface (layer 0). The regression model has the following form:

$$\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{p}) + \boldsymbol{\varepsilon}, \quad \mathsf{E}[\boldsymbol{\varepsilon}] = \boldsymbol{0}, \quad \mathsf{Var}[\boldsymbol{\varepsilon}] = \boldsymbol{V}, \tag{6.1}$$

where $\boldsymbol{m} = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n)^\mathsf{T}$ and $\boldsymbol{f} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n)^\mathsf{T}$. The function $\boldsymbol{f}_k$ maps the initial parameters $\boldsymbol{p}$ to the measurement $\boldsymbol{m}_k$ in layer k. It is the composition of the track propagators up to layer k (see Sect. 4.3) and the function that maps the track state to the measurement (see Sect. 3.2.3):

$$\boldsymbol{f}_k = \boldsymbol{h}_k \circ \boldsymbol{f}_{k|k-1} \circ \boldsymbol{f}_{k-1|k-2} \circ \ldots \circ \boldsymbol{f}_{2|1} \circ \boldsymbol{f}_{1|0}, \tag{6.2}$$

Its Jacobian $\boldsymbol{F}_k$ is given by the product of the respective Jacobians:

$$\boldsymbol{F}_k = \boldsymbol{H}_k \boldsymbol{F}_{k|k-1} \boldsymbol{F}_{k-1|k-2} \ldots \boldsymbol{F}_{2|1} \boldsymbol{F}_{1|0}. \tag{6.3}$$

The covariance matrix $\boldsymbol{V}$ is the sum of two parts, $\boldsymbol{V} = \boldsymbol{V}_\mathrm{M} + \boldsymbol{V}_\mathrm{S}$. The first part is the joint covariance matrix of all measurement errors. These can virtually always be assumed to be uncorrelated across different layers, so that $\boldsymbol{V}_\mathrm{M}$ is block-diagonal:

$$\boldsymbol{V}_\mathrm{M} = \mathrm{blkdiag}(\boldsymbol{V}_1, \ldots, \boldsymbol{V}_n), \quad \text{with} \quad \boldsymbol{V}_i = \mathsf{Var}[\boldsymbol{\varepsilon}_i], \quad i = 1, \ldots, n. \tag{6.4}$$

The second part $\boldsymbol{V}_\mathrm{S}$ is the joint covariance matrix of the process noise caused by material effects, mainly multiple Coulomb scattering; see Sect. 4.5. As in Sect. 3.2.3, the process noise encountered during the propagation from layer $k-1$ to layer k is denoted by $\boldsymbol{\gamma}_k$ and its covariance matrix by $\boldsymbol{Q}_k$. The integrated process noise up to layer k is denoted by $\boldsymbol{\Gamma}_k$. Linearized error propagation along the track gives the following expression for the covariance matrix of $\boldsymbol{\Gamma}_k$:

$$\mathsf{Var}[\boldsymbol{\Gamma}_k] = \sum_{i=1}^{k} \boldsymbol{F}_{k|i} \, \boldsymbol{Q}_i \, \boldsymbol{F}_{k|i}^\mathsf{T}, \tag{6.5}$$

with

$$\boldsymbol{F}_{k|i} = \boldsymbol{F}_{k|k-1} \boldsymbol{F}_{k-1|k-2} \cdots \boldsymbol{F}_{i+1|i} \quad \text{for} \quad i < k \quad \text{and} \quad \boldsymbol{F}_{k|k} = \boldsymbol{I}. \tag{6.6}$$

If $i < k$, $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Gamma}_k$ are correlated with the following cross-covariance matrix:

$$\mathsf{Cov}[\boldsymbol{\Gamma}_i, \boldsymbol{\Gamma}_k] = \sum_{j=1}^{i} \boldsymbol{F}_{i|j} \, \boldsymbol{Q}_j \, \boldsymbol{F}_{k|j}^\mathsf{T}. \tag{6.7}$$

Error propagation from the track states to the measurements gives the final block structure of $\boldsymbol{V}_S$:

$$\boldsymbol{V}_S = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} & \cdots & \boldsymbol{C}_{1n} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} & \cdots & \boldsymbol{C}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{n1} & \boldsymbol{C}_{n2} & \cdots & \boldsymbol{C}_{nn} \end{pmatrix}, \quad \text{with} \quad \boldsymbol{C}_{ik} = \begin{cases} \boldsymbol{H}_k \mathsf{Var}\,[\boldsymbol{\Gamma}_k]\, \boldsymbol{H}_k^{\mathsf{T}}, & \text{if } i = k \\ \boldsymbol{H}_i \mathsf{Cov}\,[\boldsymbol{\Gamma}_i, \boldsymbol{\Gamma}_k]\, \boldsymbol{H}_k^{\mathsf{T}}, & \text{if } i < k \\ \boldsymbol{C}_{ki}^{\mathsf{T}}, & \text{if } i > k \end{cases} \tag{6.8}$$

Estimation of the initial state $\boldsymbol{p}$ is usually done by the Gauss-Newton method; see Sect. 3.2.2. A first approximation $\tilde{\boldsymbol{p}}_0$ of $\boldsymbol{p}$ has to be delivered by the track finder to be used as the expansion point to compute the Jacobians in Eq. (6.3). The updated estimate $\tilde{\boldsymbol{p}}_1$ is obtained via:

$$\tilde{\boldsymbol{p}}_1 = \tilde{\boldsymbol{p}}_0 + (\boldsymbol{F}_0^{\mathsf{T}} \boldsymbol{G}\, \boldsymbol{F}_0)^{-1}\, \boldsymbol{F}_0^{\mathsf{T}} \boldsymbol{G}\, [\boldsymbol{m} - \boldsymbol{f}\,(\tilde{\boldsymbol{p}}_0)], \quad \boldsymbol{F}_0 = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\right|_{\tilde{\boldsymbol{p}}_0}. \tag{6.9}$$

The corresponding chi-square statistic is given by:

$$\chi_1^2 = [\boldsymbol{m} - \boldsymbol{f}\,(\tilde{\boldsymbol{p}}_1)]^{\mathsf{T}} \boldsymbol{G}\, [\boldsymbol{m} - \boldsymbol{f}\,(\tilde{\boldsymbol{p}}_1)], \quad \text{with} \quad \boldsymbol{G} = \boldsymbol{V}^{-1}. \tag{6.10}$$

It is approximately χ^2 distributed with $M - m$ degrees of freedom, where $M = \dim(\boldsymbol{G})$ is the sum of all m_i, and m is the number of estimated track parameters, usually five.

The Jacobians are recomputed at the new expansion point $\tilde{\boldsymbol{p}}_1$, and an updated estimate $\tilde{\boldsymbol{p}}_2$ is computed. This two-step procedure is iterated until the absolute difference $\left|\chi_{k+1}^2 - \chi_k^2\right|$ is below a predefined threshold or a maximal number of iterations is reached.

The p-value of the chi-square statistic (see Sect. 3.2.1) is the primary quality indicator of the track fit; see Sect. 6.4. A small p-value indicates a misspecification of the model or at least one outlying measurement that does not fit to the track. The standardized residuals or pulls can be used to look for outliers. The residuals $\boldsymbol{r}$ of the fit are defined by:

$$\boldsymbol{r} = \boldsymbol{m} - \boldsymbol{f}\,(\tilde{\boldsymbol{p}}), \tag{6.11}$$

where $\tilde{\boldsymbol{p}}$ is the final estimate after convergence. Their covariance matrix is obtained by linearized error propagation and is approximately equal to:

$$\boldsymbol{R} = \mathsf{Var}\,[\boldsymbol{r}] \approx \boldsymbol{V} - \boldsymbol{F}\,(\boldsymbol{F}^{\mathsf{T}} \boldsymbol{G}\, \boldsymbol{F})^{-1}\, \boldsymbol{F}^{\mathsf{T}}. \tag{6.12}$$

Note that $\boldsymbol{R}$ has rank $M - m$ and thus cannot be inverted. The standardized residuals $\boldsymbol{s}$ are given by:

$$\boldsymbol{s} = \boldsymbol{r}\, ./ \sqrt{\mathrm{diag}(\boldsymbol{R})}, \tag{6.13}$$

where ./ denotes the point-wise division of two vectors or matrices (MATLAB® convention). They are approximately distributed according to a standard normal distribution. Outliers are characterized by an unusually large value of $\boldsymbol{s}$.

The residuals $\boldsymbol{r}$, i.e., the differences of the measured track and the fitted track are a superposition of measurement noise and process noise (mainly multiple scattering). The vector $\boldsymbol{r}$ of residuals can be further decomposed into two parts corresponding to the two types of noise [1]:

$$\boldsymbol{r} = \boldsymbol{r}_\mathrm{M} + \boldsymbol{r}_\mathrm{S}, \quad \text{with} \tag{6.14}$$

$$\boldsymbol{r}_\mathrm{M} = \boldsymbol{V}_\mathrm{M}\boldsymbol{G}'\boldsymbol{m}, \quad \boldsymbol{R}_\mathrm{M} = \mathsf{Var}\,[\boldsymbol{r}_\mathrm{M}] = \boldsymbol{V}_\mathrm{M}\boldsymbol{G}'\boldsymbol{V}_\mathrm{M}, \tag{6.15}$$

$$\boldsymbol{r}_\mathrm{S} = \boldsymbol{V}_\mathrm{S}\boldsymbol{G}'\boldsymbol{m}, \quad \boldsymbol{R}_\mathrm{S} = \mathsf{Var}\,[\boldsymbol{r}_\mathrm{S}] = \boldsymbol{V}_\mathrm{S}\boldsymbol{G}'\boldsymbol{V}_\mathrm{S}, \tag{6.16}$$

$$\boldsymbol{G}' = \boldsymbol{G}\boldsymbol{R}\boldsymbol{G}. \tag{6.17}$$

The two noise contributions can thus be checked independently via their standardized residuals $\boldsymbol{s}_\mathrm{M}$ and $\boldsymbol{s}_\mathrm{S}$, given by:

$$\boldsymbol{s}_\mathrm{M} = \boldsymbol{r}_\mathrm{M} ./ \sqrt{\mathrm{diag}(\boldsymbol{R}_\mathrm{M})} \tag{6.18}$$

$$\boldsymbol{s}_\mathrm{S} = \boldsymbol{r}_\mathrm{S} ./ \sqrt{\mathrm{diag}(\boldsymbol{R}_\mathrm{S})} \tag{6.19}$$

6.1.2 Extended Kalman Filter

A "progressive" or recursive version of the LS regression for track fitting was first proposed in [2]. Soon, it was realized that this was the same as an (extended) Kalman filter [3] in the state space model of the track dynamics. The Kalman filter has the advantage that only small matrices have to be inverted, that the track fit follows the track as closely as possible, and that material effects such as multiple scattering and energy loss can be treated locally in each measurement or material layer. In addition, the filter can be complemented by the smoother; see Sect. 3.2.3.

In track fitting with the extended Kalman filter, it is assumed that the trajectory crosses a number of surfaces or layers with well-defined positions and orientations. A layer can be a measurement layer, a material layer, or both. At the intersection point of the trajectory with layer k, the state vector $\boldsymbol{q}_k$ contains information about the local position, the local direction, and the local momentum of the track. The uncertainty of the information is specified by the associated covariance matrix $\boldsymbol{C}_k$. Different possible parameterizations of the track state are discussed in Sect. 4.2.

The Kalman filter is a sequence of alternating prediction and update steps; see Sect. 3.2.3. In the prediction step, the estimate $\tilde{\boldsymbol{q}}_{k-1}$ of the track state in layer $k-1$ is extrapolated to layer k, along with its covariance matrix $\boldsymbol{C}_{k-1}$:

$$\tilde{\boldsymbol{q}}_{k|k-1} = \boldsymbol{f}_{k|k-1}(\tilde{\boldsymbol{q}}_{k-1}), \quad \boldsymbol{C}_{k|k-1} = \boldsymbol{F}_{k|k-1}\boldsymbol{C}_{k-1}\boldsymbol{F}_{k|k-1}^\mathsf{T}, \tag{6.20}$$

where $\boldsymbol{f}_{k|k-1}$ is the track propagator from layer $k-1$ to layer k, and $\boldsymbol{F}_{k|k-1}$ is its Jacobian matrix; see Sect. 4.3.

The update step is different in material and detector layers. In a material layer, multiple scattering is taken into account by inflating the covariance matrix elements of the track directions and, in a thick scatterer, of the track position; see Sect. 4.5. Energy loss by ionization is taken into account by decreasing the track momentum. For the treatment of electron bremsstrahlung, see Sect. 6.2.3.

The update step in a detector layer is given by Eqs. (3.29) and (3.30) or Eqs. (3.31) and (3.32). The associated chi-square statistic χ^2_k, Eq. (3.35), can be used to test the compatibility of the observation $\boldsymbol{m}_k$ with the predicted state $\tilde{\boldsymbol{q}}_{k|k-1}$ in the combinatorial Kalman filter; see Sect. 5.1.7. A large value of χ^2_k or a small p-value indicates that the observation does not belong to the track.

In track fitting, the initial state $\boldsymbol{q}_0$ is obtained from the track finder and therefore contains information from the observations. In order not to use the information twice, its covariance matrix is set to a large diagonal matrix. As a consequence, the initial chi-square statistics χ^2_k have zero degrees of freedom, until the covariance matrix of the state vector has full rank. For instance, if all measurements are 2D, and the state is 5D, χ^2_1 and χ^2_2 have zero degrees of freedom, χ^2_3 has one degree of freedom, all subsequent χ^2_k have two degrees of freedom and the total chi-square statistic χ^2_{tot} has $2n-5$ degrees of freedom.

The smoother can be implemented either according to Eqs. (3.36) and (3.37) or by running a second filter in the opposite direction and combining the states of the two filters by a weighted mean (Eq. (3.38)):

$$\tilde{\boldsymbol{q}}_{k|n} = \boldsymbol{C}_{k|n}\left[\boldsymbol{C}_k{}^{-1}\tilde{\boldsymbol{q}}_k + \left(\boldsymbol{C}^{\text{b}}_{k|k+1}\right)^{-1}\tilde{\boldsymbol{q}}^{\text{b}}_{k|k+1}\right],\quad \boldsymbol{C}^{-1}_{k|n} = \boldsymbol{C}^{-1}_k + \left(\boldsymbol{C}^{\text{b}}_{k|k+1}\right)^{-1}, \tag{6.21}$$

where $\tilde{\boldsymbol{q}}^{\text{b}}_{k|k+1}$ is the predicted state from the backward filter and $\boldsymbol{C}^{\text{b}}_{k|k+1}$ its covariance matrix. Alternatively, the predicted state from the forward filter and the updated state from the backward filter can be combined. The associated chi-square statistic $\chi^2_{k|n}$ (Eqs. (3.39) and (3.40)) can be used to test the compatibility of the observation $\boldsymbol{m}_k$ with the smoothed state $\tilde{\boldsymbol{q}}_{k|n}$, using the entire track information. A large value of $\chi^2_{k|n}$ or a small p-value indicates that the observation does not belong to the track and is an outlier; see Sect. 6.4.2. A simplified version of the chi-square statistic $\chi^2_{k|n}$ is used in the deterministic annealing filter; see Sect. 6.2.2.

6.1.3 Regression with Breakpoints

Instead of absorbing the effects of multiple scattering in the covariance matrix of the process noise, the scattering angles at certain points, called breakpoints, can be explicitly incorporated into the track model as additional parameters. At

these breakpoints, the scattering angles are estimated [4, 5], using their known expectation (zero) and known covariance matrix in the curvilinear parameterization; see Sect. 4.5.1. The breakpoint fit is mathematically equivalent both to LS regression in the linear approximation of the model and to the Kalman filter, unless the initial state of the latter has non-negligible information.

Let $\boldsymbol{\theta}_j = (\theta_{j1}, \theta_{j2}), j = 1, \ldots, m$ denote two uncorrelated multiple scattering angles at the breakpoint j, and $\boldsymbol{Q}_j$ their covariance matrix. Then the regression model in Eq. (6.1) can be modified to:

$$\boldsymbol{m}_k = \hat{\boldsymbol{f}}_k(\boldsymbol{p}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{j_k}) + \boldsymbol{\varepsilon}_k, \quad \mathsf{E}[\boldsymbol{\varepsilon}_k] = \boldsymbol{0}, \quad \mathsf{Var}[\boldsymbol{\varepsilon}_k] = \boldsymbol{V}_k, \quad k = 1, \ldots, n, \tag{6.22}$$

where j_k is the index of the last breakpoint before measurement layer k. All measurement errors $\boldsymbol{\varepsilon}_k$ are now independent, and their joint covariance matrix is block-diagonal, as is the joint covariance matrix of all $\boldsymbol{\theta}_j$. The full regression model, which includes the scattering angles as additional parameters, now reads:

$$\begin{pmatrix} \boldsymbol{m}_1 \\ \vdots \\ \boldsymbol{m}_n \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{f}}_1(\boldsymbol{p}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{j_1}) \\ \vdots \\ \hat{\boldsymbol{f}}_n(\boldsymbol{p}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m) \\ \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_m \end{pmatrix} + \boldsymbol{\delta}, \quad \text{with} \tag{6.23}$$

$$\mathsf{E}[\boldsymbol{\delta}] = \boldsymbol{0}, \quad \mathsf{Var}[\boldsymbol{\delta}] = \mathrm{blkdiag}(\boldsymbol{V}_1, \ldots, \boldsymbol{V}_n, \boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_m). \tag{6.24}$$

If the functions $\hat{\boldsymbol{f}}_k$ are Taylor-expanded to first order, a linear regression model with the following structure is obtained:

$$\begin{pmatrix} \boldsymbol{m}_1 \\ \vdots \\ \boldsymbol{m}_n \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_1 & \boldsymbol{H}_{1|1} & \cdots & \boldsymbol{H}_{1|j_1} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{F}_n & \boldsymbol{H}_{n|1} & \cdots & \cdots & \cdots & \cdots & \boldsymbol{H}_{n|m} \\ \boldsymbol{O} & \boldsymbol{I} & \boldsymbol{O} & \cdots & \cdots & \cdots & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{O} & \cdots & \cdots & \cdots & \cdots & \boldsymbol{O} & \boldsymbol{I} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_m \end{pmatrix} + \boldsymbol{c}, \tag{6.25}$$

where $\boldsymbol{H}_{k|j}, j \leq j_k$ is the Jacobian matrix of the function that describes the dependence of $\boldsymbol{m}_k$ on $\boldsymbol{\theta}_j$ and $\boldsymbol{F}_k$ is as in Eq. (6.3).

The following two subsections describe two efficient implementations of the breakpoint concept.

6.1.4 General Broken Lines

The general broken lines (GBL) algorithm [6] is a fast track refit based on the breakpoint concept. It is particularly useful for track-based alignment and calibration with the Millepede-II package [7]. It is assumed that only thin scatterers are present, or that thick scatterers are divided into several thin scatterers. The algorithm needs an initial trajectory as a reference.

At each measurement plane and thin scatterer, a local orthonormal coordinate system (u, v, w) is defined. The natural choice of the w-axis is perpendicular to the sensor plane for a measurement and parallel to the track direction for a scatterer. At each thin scatterer, the offset (u, v) is a fit parameter, as is the global signed inverse momentum q/p. The prior information on the scattering angles is the same as above, obtained from multiple scattering theory; see Sect. 4.5.1. The transformations from the curvilinear frame following the track to the local frames are given in Sect. 4.4.1.

The corrections to the track parameters in a measurement plane depend only on the (inverse) momentum and the adjacent offsets; therefore, their estimation requires the inversion of a bordered band matrix, the computing time of which is proportional to the number of breakpoints. The GBL is available in a dedicated software package [8] and is also implemented in GENFIT [9–13].

A comparative study in [6] shows that track fitting with the GBL is a little faster than the extended Kalman filter and up to three times faster than the Kalman filter plus smoother.

6.1.5 Triplet Fit

A track fit for situations in which the principal source of stochastic uncertainty is multiple scattering is described in [14]. The fit is an independent combination of fits to triplets of hits, where the middle point of a triplet is a breakpoint. As the triplet fits are fast and can be parallelized, the method is well suited for online reconstruction. The fit has been designed for low momentum tracks with several turns in the detector, but performs also very well for tracks in a high-resolution pixel tracker with momenta up to 10 GeV. For a comparison with a full helix fit and the GBL fit (Sect. 6.1.4) see [14]. The fit is implemented in a software package called WATSON [15] which is available on request from the authors of [14].

6.1.6 Fast Track Fit by Affine Transformation

Online track reconstruction in the first-level trigger, see Sect. 5.2, requires an ultra-fast track fit that can be implemented in high-speed hardware such as FPGAs [16]. One possibility to achieve the required speed is to fit an affine model to a training

sample of simulated tracks that expresses the track parameters $\boldsymbol{p}$ as an affine function of the measurements $\boldsymbol{m}$ [17]:

$$\tilde{\boldsymbol{p}} = \boldsymbol{A}\boldsymbol{m} + \boldsymbol{c}. \tag{6.26}$$

The matrix $\boldsymbol{A}$ and the vector $\boldsymbol{c}$ are estimated by minimizing the objective function

$$\mathcal{S}(\boldsymbol{A}, \boldsymbol{c}) = \sum_{i=1}^{N} \left(\boldsymbol{A}\boldsymbol{m}_i + \boldsymbol{c} - \boldsymbol{p}_i\right)^{\mathsf{T}} \left(\boldsymbol{A}\boldsymbol{m}_i + \boldsymbol{c} - \boldsymbol{p}_i\right), \tag{6.27}$$

where the $\boldsymbol{p}_i$ are the true parameters of the N tracks in the training sample. The solution is given by:

$$\boldsymbol{A} = \left[\langle \boldsymbol{p}\boldsymbol{m}^{\mathsf{T}} \rangle - \langle \boldsymbol{p} \rangle \langle \boldsymbol{m}^{\mathsf{T}} \rangle\right] \boldsymbol{C}^{-1}, \; \boldsymbol{c} = \langle \boldsymbol{p} \rangle - \boldsymbol{A} \langle \boldsymbol{m} \rangle, \; \boldsymbol{C} = \langle \boldsymbol{m}\boldsymbol{m}^{\mathsf{T}} \rangle - \langle \boldsymbol{m} \rangle \langle \boldsymbol{m}^{\mathsf{T}} \rangle, \tag{6.28}$$

where $\boldsymbol{C}$ is the sample covariance matrix of the measurement vectors, and the angle brackets denote the average over the training sample. The goodness-of-fit can be judged by the chi-square statistic

$$\chi^2 = \left[\boldsymbol{m} - \langle \boldsymbol{m} \rangle\right]^{\mathsf{T}} \boldsymbol{C}^{-1} \left[\boldsymbol{m} - \langle \boldsymbol{m} \rangle\right], \tag{6.29}$$

which is approximately χ^2-distributed with $M - m$ degrees of freedom, where $M = \dim(\boldsymbol{m})$ and $m = \dim(\boldsymbol{p})$. If the track model is exactly linear, i.e., $\boldsymbol{m} = \boldsymbol{F}\boldsymbol{p}$ with some matrix $\boldsymbol{F}$, the matrix $\boldsymbol{A}$ is equal to the Moore-Penrose pseudoinverse of $\boldsymbol{F}$, and the rank of $\boldsymbol{C}$ is m, so that $\boldsymbol{C}$ has m positive eigenvalues while the remaining ones are equal to 0. If the linear approximation to the track model is valid in the entire training sample, $\boldsymbol{C}$ has m large and $M - m$ small eigenvalues. The training sample, therefore, has to be chosen such that this condition is satisfied. In practice, this means that the detector volume has to be partitioned into many small regions, each with its own training sample. The training automatically takes into account material effects, measurement errors, misalignment, and the configuration of the magnetic field.

6.2 Robust and Adaptive Fitting

6.2.1 Robust Regression

The LS regression described in Sect. 6.1.1 is not robust in the sense that outliers in the track candidate lead to a significant distortion of the estimated parameters. One way to cope with this problem is to look for outliers in the standardized residuals of

the regression and to remove them; see Sect. 6.4.2. A more elegant way is to make the regression robust by minimizing an objective function that is different from the sum of squared differences between the model and the measurements.

Three important approaches to robust regression are Least Median of Squares (LMS), Least Trimmed Squares (LTS), and M-estimators [18, 19]. LMS regression is difficult to compute in the context of track fitting; it is statistically less efficient than either LS or LTS, and there is no simple prescription for the computation of the covariance matrix of the estimated parameters. The computation of the LTS estimator requires the solution of a combinatorial optimization problem and is therefore rather time-consuming. The M-estimator, on the other hand, can be implemented as an iterated re-weighted LS regression and is therefore an excellent method for a robust track fit. The re-weighting can be done on single components of the measurement vectors $\boldsymbol{m}_i$ or on entire measurement vectors.

An M-estimator is characterized by a function $\psi(z)$ that determines the corresponding weight function $\omega(z) = \psi(z)/z$, where z is the standardized residual of a measurement. Setting $\psi(z) = z$ yields the LS estimator. Table 6.1 and Fig. 6.1 show three examples of ψ and weight functions from the literature.

The constant c controls the shape of the weight function. The weight function of Huber's M-estimator [19] is equal to one in the interval $[-c, c]$ and slowly drops to zero outside the interval (see Fig. 6.1a). Tukey's biweight [20] is a redescending estimator for which the weight function is equal to zero outside the interval $[-c, c]$ (see Fig. 6.1b). For the adaptive M-estimator [21], c is the point where the weight function is equal to 0.5. This estimator has an additional control parameter T that modifies the transition from large to small weight and can be used for implementing an annealing procedure; see Sect. 6.2.2. In the limit of $T \to 0$, the estimator is redescending, as the weight function approaches a step function that drops from 1 to 0 at $z = c$ (see Fig. 6.1c). The computation of the M-estimator is summarized in Table 6.2. In order to be less sensitive to multiple scattering, it uses only the measurement component $\boldsymbol{s}_{\mathrm{M}}$ of the standardized residuals; see Eq. (6.18).

Table 6.1 The ψ functions and the corresponding weight functions ω of three M-estimators. c and T are constants. Further explanations can be found in the text

Name	$\psi(z)$	$\omega(z)$
Huber	z, if $\lvert z\rvert \le c$ $c \cdot \mathrm{sign}(z)$, if $\lvert z\rvert > c$	1, if $\lvert z\rvert \le c$ $c/\lvert z\rvert$, if $\lvert z\rvert > c$
Tukey's biweight	$z\,(1 - z^2/c^2)^2$, if $\lvert z\rvert \le c$ 0, if $\lvert z\rvert > c$	$(1 - z^2/c^2)^2$, if $\lvert z\rvert \le c$ 0, if $\lvert z\rvert > c$
Adaptive	$\dfrac{z \exp(-z^2/2T)}{\exp(-z^2/2T) + \exp(-c^2/2T)}$	$\dfrac{\exp(-z^2/2T)}{\exp(-z^2/2T) + \exp(-c^2/2T)}$

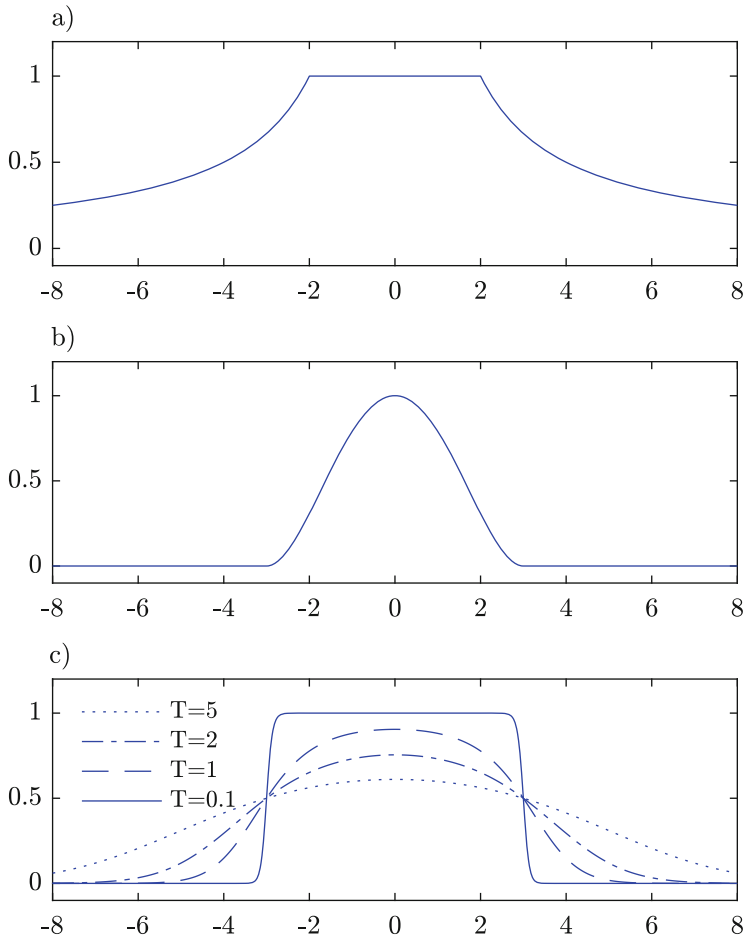

Fig. 6.1 The weight functions of the three M-estimators in Table 6.1. (**a**) Huber type with $c = 2$. (**b**) Turkey's biweight with $c = 3$. (**c**) Adaptive with $c = 3$

6.2.2 Deterministic Annealing Filter

The deterministic annealing filter (DAF) is an adaptive version of the standard Kalman filter [22]. It can modify the influence of outlying measurements by assigning them a smaller weight. It can also deal with the case that two (or more) measurements in the same detector layer are tagged by the track finder as valid candidates for inclusion in the track. This is particularly relevant in the LHC experiments, given the high track density in the central tracker. It is then up to the track fit to decide which of the competing measurements, if any, is most compatible with the local track state. Another use case is the choice between two mirror hits in a drift chamber. The DAF is implemented in GENFIT [9–12].

The measurements in layer k are denoted by $\boldsymbol{m}_k^j$, $j = 1, \ldots, M_k$, and their covariance matrices by $\boldsymbol{V}_i^j$. The DAF is implemented as an iterated Kalman filter plus smoother with annealing; see Table 6.3 for the basic sequence and [23, 24] for implementation details. In each layer, the state vector is updated with a weighted mean of the measurements, using the covariance matrices of the current iteration.

Table 6.2 Algorithm: Track fit with M-estimator

TRACK FIT WITH M-ESTIMATOR

1. Set $\varepsilon = 10^{-6}$, $\widehat{\boldsymbol{V}} = \boldsymbol{V}$, $\widehat{\boldsymbol{V}}_\mathrm{M} = \boldsymbol{V}_\mathrm{M}$, and compute the LS regression.
2. Compute the standardized residuals $\boldsymbol{s}_\mathrm{M}$ in Eq. (6.18).
3. Compute the weights w_j of the components $s_{\mathrm{M},j}$ of $\boldsymbol{s}_\mathrm{M}$ according to:

$$w_j = \max[\varepsilon, \omega(s_{\mathrm{M},j})], \quad j = 1, \ldots, M. \tag{6.30}$$

   Set $\boldsymbol{V}_\mathrm{M} = \widehat{\boldsymbol{V}}_\mathrm{M}./\mathrm{diag}(\boldsymbol{w})$ and $\boldsymbol{V} = \boldsymbol{V}_\mathrm{M} + \boldsymbol{V}_\mathrm{S}$.
4. Recompute the LS regression, set $\boldsymbol{V} = \widehat{\boldsymbol{V}}$ and repeat from step 2 until convergence.

Table 6.3 Algorithm: Deterministic annealing filter

DETERMINISTIC ANNEALING FILTER

1. Set $\varepsilon = 10^{-6}$, set the iteration counter $I = 0$ and $\boldsymbol{V}_k^j(0) = \boldsymbol{V}_k^j$, $j = 1, \ldots, M_k$, $k = 1, \ldots, n$. Choose a threshold χ_c^2 and an initial temperature T_0.
2. Run a Kalman filter plus smoother, using the covariance matrices $\boldsymbol{V}_k^j(I)$ in all updates.
3. For all layers $k = 1, \ldots, n$, compute the weights w_k^j of measurements $\boldsymbol{m}_k^j$, $j = 1, \ldots, M_k$:

$$w_k^j = \max\left[\varepsilon, \frac{\exp(-X_k^j/2T_k)}{\exp(-\chi_\mathrm{c}^2/2T_k) + \sum_{\ell=1}^{M_k} \exp(-X_k^\ell/2T_k)}\right],$$

$$X_k^j = \left[\boldsymbol{m}_k^j - \boldsymbol{h}_k(\boldsymbol{q}_{k|n})\right]^\mathsf{T} \left[\boldsymbol{V}_k^j(0)\right]^{-1} \left[\boldsymbol{m}_k^j - \boldsymbol{h}_k(\boldsymbol{q}_{k|n})\right].$$

   X_k^j is the χ^2-distance of $\boldsymbol{m}_k^j$ from the smoothed state in layer k in the metric defined by the inverse of $\boldsymbol{V}_k^j(0)$, see Eq. (3.39).
4. Increase the iteration counter I by 1 and set $\boldsymbol{V}_k^j(I) = \boldsymbol{V}_k^j(0)/w_k^j$, $j = 1, \ldots, M_k$, $k = 1, \ldots, n$.
5. Set the new temperature T_I according to the chosen annealing schedule and go to step 2.
6. When the final temperature is reached, iterate until convergence.

6.2.3 Gaussian-Sum Filter

The Kalman filter is (near) optimal if the track model is (approximately) linear, and both system and measurement noise are (approximately) Gaussian; see Sect. 3.2.3. If the noise is long-tailed or highly asymmetric, the Gaussian-sum filter (GSF) is an alternative estimator. It can be applied to a broad class of non-Gaussian distributions by allowing all densities involved to be mixtures of normal PDF s or Gaussian sums. The GSF can be applied in the following use cases:

1. *Long-tailed measurement errors.* The distribution of the measurement errors is contaminated by frequent outliers and can be modelled by a mixture of two Gaussians, the "core" and the "tails" [25].
2. *Thin scatterers.* The distribution of the multiple scattering angle in thin layers is non-Gaussian because of its long tails [26], but can be approximated by a Gaussian sum with two components [27]; see Sect. 4.5.1.
3. *Inhomogeneous scatterers.* Multiple scattering in an inhomogeneous material is often treated by computing an average thickness and an average radiation length (Eq. (4.79)). With the GSF, it is possible to describe the angular distribution by a Gaussian sum, with one or two components for each type of material [28].
4. *Energy loss by bremsstrahlung.* The distribution of the energy loss caused by bremsstrahlung of electrons is very far from being Gaussian. While the Kalman filter is restricted to using the first two moments (mean and variance) of the distribution, an approximation by a normal mixture allows the GSF to take into account more details of the shape of the energy loss PDF [29, 30]; see Sect. 4.5.3.

In the GSF, the PDF of the state vector can be a Gaussian sum at every surface. First assume that the surface is a material surface and that the predicted state vector $\boldsymbol{q}$ at the entry of the surface has the following normal mixture PDF with J components:

$$p_0(\boldsymbol{q}) = \sum_{j=1}^{J} \pi_j \varphi(\boldsymbol{q}; \boldsymbol{q}_j, \boldsymbol{C}_j), \quad \sum_{j=1}^{J} \pi_j = 1, \tag{6.31}$$

where $\varphi(\boldsymbol{q}; \boldsymbol{q}_j, \boldsymbol{C}_j)$ is the normal PDF with mean $\boldsymbol{q}_j$ and covariance matrix $\boldsymbol{C}_j$, $j = 1, \ldots, J$. The process noise $\boldsymbol{\gamma}$ in the surface is modeled by a normal mixture as well (see also Sect. 3.2.3.2):

$$g(\boldsymbol{\gamma}) = \sum_{m=1}^{M} \omega_m \varphi(\boldsymbol{\gamma}; \boldsymbol{g}_m, \boldsymbol{Q}_m), \quad \sum_{m=1}^{M} \omega_m = 1. \tag{6.32}$$

Note that both J and M may be equal to one. Then the PDF of the state vector at the exit of the surface is given by the following normal mixture with $J \times M$ components:

$$p_1(\boldsymbol{q}) = \sum_{j=1}^{J} \sum_{m=1}^{M} \pi_j \omega_m \varphi(\boldsymbol{q}; \boldsymbol{q}_j + \boldsymbol{g}_m, \boldsymbol{C}_j + \boldsymbol{Q}_m). \tag{6.33}$$

Now assume that the surface is a measurement surface and that the distribution of the measurement error of measurement $\boldsymbol{m}$ is modeled by a normal mixture:

$$g(\boldsymbol{\varepsilon}) = \sum_{m=1}^{M} \omega_m \varphi(\boldsymbol{\varepsilon}; \mathbf{0}, \boldsymbol{V}_m), \quad \sum_{m=1}^{M} \omega_m = 1. \tag{6.34}$$

Again, J and M may be equal to one. The PDF of the updated state vector is given by:

$$p_1(\boldsymbol{q}) = \sum_{j=1}^{J} \sum_{m=1}^{M} \eta_{jm} \varphi(\boldsymbol{q}; \boldsymbol{q}_{jm}, \boldsymbol{C}_{jm}), \tag{6.35}$$

with

$$\eta_{jm} \propto \pi_j \omega_m \varphi(\boldsymbol{m}; \boldsymbol{h}(\boldsymbol{q}_j), \boldsymbol{V}_m + \boldsymbol{H}\boldsymbol{C}_j\boldsymbol{H}^\mathsf{T}), \quad \sum_{j=1}^{J} \sum_{m=1}^{M} \eta_{jm} = 1. \tag{6.36}$$

The mean $\boldsymbol{q}_{jm}$ and the covariance matrix $\boldsymbol{C}_{jm}$ are obtained by the Kalman filter update of component j of the predicted PDF $p_0(\boldsymbol{q})$ with component m of the measurement error PDF $g(\boldsymbol{\varepsilon})$, see Eqs. (3.29) and (3.30) or Eqs. (3.31) and (3.32).

Finally, assume that there are M measurements $\boldsymbol{m}_m$ in the measurement surface, with weights ω_m, $m = 1, \ldots, M$. In the absence or prior information, all weights are set to $1/M$. The observation $\boldsymbol{m}$ can then be modeled by the following normal mixture PDF :

$$g(\boldsymbol{m}) = \sum_{m=1}^{M} \omega_m \varphi(\boldsymbol{m}; \boldsymbol{m}_m, \boldsymbol{V}_m), \quad \sum_{m=1}^{M} \omega_m = 1. \tag{6.37}$$

The PDF of the updated state vector is given by:

$$p_1(\boldsymbol{q}) = \sum_{j=1}^{J} \sum_{m=1}^{M} \eta_{jm} \varphi(\boldsymbol{q}; \boldsymbol{q}_{jm}, \boldsymbol{C}_{jm}), \tag{6.38}$$

with

$$\eta_{jm} \propto \pi_j \omega_m \varphi(\boldsymbol{m}_m; \boldsymbol{h}(\boldsymbol{q}_j), \boldsymbol{V}_m + \boldsymbol{H}\boldsymbol{C}_j\boldsymbol{H}^\mathsf{T}), \quad \sum_{j=1}^{J} \sum_{m=1}^{M} \eta_{jm} = 1. \tag{6.39}$$

The mean $\boldsymbol{q}_{jm}$ and the covariance matrix $\boldsymbol{C}_{jm}$ are obtained by a Kalman filter update of component j of the predicted PDF $p_0(\boldsymbol{q})$ with component m of the observation PDF $g(\boldsymbol{m})$. The resulting GSF is basically a combinatorial Kalman filter in which

each track candidate has an additional weight which shows how likely it is compared to the other ones. This version of the GSF can be used for track finding; in this case, a missing hit with large errors should be added in each layer [31]. Its weight can reflect the hit efficiency of the measurement device.

In principle, the number of components rises exponentially in the course of the GSF, and it is necessary to reduce their number whenever it exceeds a threshold K set by the user. The simplest way of reducing the number of components is to keep the K components with the largest weights, drop the remaining ones, and renormalize the weights to a sum of one. A more sophisticated approach is to search for clusters among the components and to collapse the components in a cluster to a single Gaussian with the same mean and covariance matrix. Clustering can be based on the similarity between components, as measured by, e.g., the Kullback–Leibler divergence [30]. For a brief review of clustering algorithms, see Sect. 3.3. The choice of the threshold K and the clustering procedure have to be optimized by simulation studies.

Even for moderate values of K, the GSF is significantly slower than the Kalman filter; it is, therefore, used mainly for special applications such as the track fit of electrons with non-negligible bremsstrahlung [30, 32, 33].

6.3 Linear Approaches to Circle and Helix Fitting

6.3.1 Conformal Mapping Method

The conformal transformation described in Sect. 5.1.1 can be generalized to deal with circles passing close the origin [34]. The conformal transformation maps such a circle to a circle with a very small curvature, which in turn can be well approximated by a parabola:

$$v = \frac{1}{2b} - \frac{a}{b} \cdot u - \epsilon \left(\frac{R}{b}\right)^3 \cdot u^2, \tag{6.40}$$

where $\epsilon = R - \sqrt{a^2 + b^2}$ is the impact parameter. A standard parabola fit to the measurements in the transformed (u, v)-coordinates yields the parameters A, B and C according to

$$v = A + Bu + Cu^2, \tag{6.41}$$

and the circle parameters are therefore given by

$$b = \frac{1}{2A}, \quad a = -bB, \quad \epsilon = -C \cdot \frac{b^3}{(a^2 + b^2)^{3/2}}, \tag{6.42}$$

using the approximation $R \approx \sqrt{a^2 + b^2}$ in the expression of ϵ [34]. Following Gluckstern [35], it is also possible to obtain expressions of the estimated errors of the circle parameters [34].

6.3.2 Chernov and Ososkov's Method

The task of fitting a circular track to a set of measurements is tantamount to minimizing the function

$$\chi^2 = \sum_{i=1}^{n} {d_i}^2, \tag{6.43}$$

where d_i are measurement residuals orthogonal to the particle trajectory:

$$d_i = \pm \left[\sqrt{(x_i - a)^2 + (y_i - b)^2} - R \right], \; i = 1, \ldots, n, \tag{6.44}$$

where a, b, and R are the coordinates of the circle centre and the radius. The approach of Chernov and Ososkov [36] is to simplify this non-linear minimization problem by introducing an approximate expression for the residuals d_i,

$$d_i \approx \pm \left[(x_i - a)^2 + (y_i - b)^2 - R^2 \right] / 2R, \tag{6.45}$$

which holds true with high precision as long as the residuals are small compared to the circle radius. The equations obtained by differentiating χ^2 with respect to the circle parameters and setting these to zero are quartic (polynomial equations of degree 4) and can be solved efficiently by a standard Newton iteration procedure.

6.3.3 Karimäki's Method

Karimäki's approach [37] starts from the simplified expression of the residuals d_i introduced by Chernov and Ososkov [36] and considers a χ^2 with weighted residuals:

$$\chi^2 = \sum_{i=1}^{n} w_i {d_i}^2, \tag{6.46}$$

The weights can for instance contain measurement uncertainties if they are not the same for all measurements (x_i, y_i).

The χ^2 function is minimized with respect to a set of circle parameters with Gaussian behaviour: the curvature κ (the inverse radius of curvature), the impact parameter ϵ (the distance from the origin to the point of closest approach of the fitted circle), and the direction ϕ of the tangent of the circle at the point of closest approach. Using this set of parameters, the simplified residuals are expressed as

$$d_i = \frac{1}{2}\kappa {r_i}^2 - (1 + \kappa\epsilon) r_i \sin(\phi - \phi_i) + \frac{1}{2}\kappa\epsilon^2 + \epsilon, \tag{6.47}$$

where r_i and ϕ_i are the polar coordinates of measurement i. The residuals can be written as $d_i = (1 + \kappa\epsilon)\,\eta_i$, with

$$\eta_i = \gamma {r_i}^2 - r_i \sin(\phi - \phi_i) + \delta, \tag{6.48}$$

and

$$\gamma = \frac{\kappa}{2(1 + \kappa\epsilon)}, \quad \delta = \frac{1 + \kappa\epsilon/2}{1 + \kappa\epsilon}\epsilon. \tag{6.49}$$

Using these definitions, the χ^2 can be written as

$$\chi^2 = (1 + \kappa\epsilon)^2 \tilde{\chi}^2, \tag{6.50}$$

where $\tilde{\chi}^2 = \sum_i w_i \eta_i^2$. With the approximation $1 + \kappa\epsilon \approx 1$, $\tilde{\chi}^2$ can be minimized instead of χ^2, leading to the attractive feature of a set of equations with explicit solutions:

$$\phi = \frac{1}{2}\arctan(2q_1/q_2), \tag{6.51}$$

$$\gamma = \left(\sin\phi \cdot C_{xz} - \cos\phi \cdot C_{yz}\right)/C_{zz}, \tag{6.52}$$

$$\delta = -\gamma\langle z\rangle + \sin\phi\langle x\rangle - \cos\phi\langle y\rangle, \tag{6.53}$$

where $q_1 = C_{zz}C_{xy} - C_{xz}C_{yz}$ and $q_2 = C_{zz}(C_{xx} - C_{yy}) - C_{xz}^2 + C_{yz}^2$ and the angle brackets $\langle\rangle$ denote a weighted average, e.g., $\langle x\rangle = \sum_i w_i x_i / \sum_i w_i$. The variances and covariances of the measurements x, y and $z = x^2 + y^2$ are given by

$$\begin{aligned} &C_{xx} = \langle x^2\rangle - \langle x\rangle^2, \quad C_{xy} = \langle xy\rangle - \langle x\rangle\langle y\rangle, \; C_{yy} = \langle y^2\rangle - \langle y\rangle^2, \\ &C_{xz} = \langle xz\rangle - \langle x\rangle\langle z\rangle, \; C_{yz} = \langle yz\rangle - \langle y\rangle\langle z\rangle, \; C_{zz} = \langle r^4\rangle - \langle z\rangle^2. \end{aligned} \tag{6.54}$$

The curvature κ and impact parameter ϵ are given by

$$\kappa = \frac{2\gamma}{\sqrt{1 - 4\delta\gamma}}, \quad \epsilon = \frac{2\delta}{1 + \sqrt{1 - 4\delta\gamma}}. \tag{6.55}$$

Expressions of the uncertainties of the estimated parameters are available and can be found in [37].

6.3.4 Riemann Fit

The Riemann circle fit [38] is based on the fundamental theorem in complex analysis that circles and lines in the plane correspond one-to-one to circles on the Riemann sphere. Since a circle on a sphere is the intersection of a plane with the sphere, there is a one-to-one correspondence between circles and lines in the plane and planes in space. The problem of fitting a circle to a set of measurements in the plane can, therefore, be transformed into the problem of fitting the transformed measurements to a plane in space. The latter problem can be solved directly by non-iterative methods.

The mapping of a point (u_i, v_i) in the plane to the transformed point (x_i, y_i, z_i) on the Riemann sphere is given by

$$\begin{aligned} x_i &= u_i/(1 + {u_i}^2 + {v_i}^2), \\ y_i &= v_i/(1 + {u_i}^2 + {v_i}^2), \\ z_i &= ({u_i}^2 + {v_i}^2)/(1 + {u_i}^2 + {v_i}^2). \end{aligned} \tag{6.56}$$

The denominator in the expressions of the transformed measurements leads to small distances between the transformed measurements and the fitted plane for large radii $R_i = \left({u_i}^2 + {v_i}^2\right)^{1/2}$ in the plane. In an attempt to satisfy the Gauss-Markov conditions as closely as possible, a radius-dependent scaling factor was introduced in the fitting procedure in [39]. It was realized in [40] that this scaling factor could be omitted by mapping the points in the plane to a paraboloid rather than the Riemann sphere,

$$x_i = u_i, \quad y_i = v_i, \quad z_i = {u_i}^2 + {v_i}^2, \tag{6.57}$$

leading to the same values of the estimated parameters if the measurements are at fixed radial positions.

Fitting a plane in space to the n measurements on the paraboloid is tantamount to minimizing the objective function

$$\mathcal{S}(c, \boldsymbol{n}) = \sum_{i=1}^{n} \frac{(c + n_1 x_i + n_2 y_i + n_3 z_i)^2}{\sigma_i^2} = \sum_{i=1}^{n} \frac{{d_i}^2}{\sigma_i^2} \tag{6.58}$$

with respect to c and $\boldsymbol{n} = (n_1, n_2, n_3)^\mathsf{T}$ with the constraint that $\boldsymbol{n}$ is a unit vector. This is achieved by choosing $\boldsymbol{n}$ as the unit eigenvector corresponding to the smallest eigenvalue of the sample covariance matrix $\boldsymbol{A}$ of the measurements:

$$\boldsymbol{A} = \frac{1}{N} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \left(\boldsymbol{r}_i - \boldsymbol{r}_{\mathrm{cg}}\right) \left(\boldsymbol{r}_i - \boldsymbol{r}_{\mathrm{cg}}\right)^\mathsf{T}. \tag{6.59}$$

The constant c is equal to $c = -\boldsymbol{n}^{\mathsf{T}}\boldsymbol{r}_{\mathrm{cg}}$ with the centre of gravity vector $\boldsymbol{r}_{\mathrm{cg}} = \sum_i w_i \boldsymbol{r}_i$ and the weights

$$w_i = \frac{1/\sigma_i^2}{\sum_{j=1}^{n} 1/\sigma_j^2}, \; i = 1, \ldots, n. \tag{6.60}$$

Given the parameters of the fitted plane, a suitable set of circle parameters can be derived. For example, the parameters chosen in Sect. 6.3.3 are given by:

$$\phi = \arctan\left(\frac{n_2}{n_1}\right), \tag{6.61}$$

$$\kappa = s \cdot \frac{2n_3}{\sqrt{1 - n_3^2 - 4cn_3}}, \tag{6.62}$$

$$\epsilon = s \cdot \frac{\sqrt{1 - n_3^2 - 4cn_3} - \sqrt{1 - n_3^2}}{2n_3}, \tag{6.63}$$

up to a sign $s = \pm 1$. One possible convention of determining s is given in [41].

Expressions of the uncertainties of the estimated circle parameters are given in [41] for measurement uncertainties both in the transverse and in the radial direction. Effects of multiple Coulomb scattering can also be included in this approach, essentially by modifying Eq. (6.59) to include correlations between all measurements due to multiple scattering [40]. A robust version of the Riemann fit based on LMS regression is proposed in [42].

6.3.5 *Helix Fitting*

Linearized helix fitting can be done by first estimating the parameters of the circle that results from the projection of the helix on the transverse (bending) plane. Any of the methods described above can be used for this. If the detector system at hand is of a barrel-type, so that the radial positions of the measurements are known to a very high precision, the path lengths to the intersections between the fitted circle and the detector elements can be obtained from the circle parameters. For the Riemann circle fit, these path lengths can be found directly from the knowledge of the parameters of the fitted plane [43]. The longitudinal (non-bending) plane parameters can then be found by solving the linear regression model:

$$\boldsymbol{z} = \boldsymbol{A}\boldsymbol{p} + \boldsymbol{\varepsilon}, \quad \boldsymbol{A} = \begin{pmatrix} 1 & s_1 \\ \vdots & \vdots \\ 1 & s_n \end{pmatrix}, \tag{6.64}$$

where $\boldsymbol{z}$ is the vector of z measurements and the parameter vector $\boldsymbol{p}$ is given by:

$$\boldsymbol{p} = \begin{pmatrix} z \\ \tan\lambda \end{pmatrix}, \quad \mathsf{Var}\left[\boldsymbol{\varepsilon}\right] = \boldsymbol{V}_z, \tag{6.65}$$

where the dip angle $\lambda = \pi/2 - \theta$ is the complement of the polar angle θ, and $\boldsymbol{V}_z$ is the covariance matrix of the measurements (containing contributions from multiple scattering, if desired). From the fitted parameters, θ and z in the innermost layer can be immediately obtained.

In a forward-type detector, the z positions are known very precisely, whereas the radial positions of the measurements are observed. In this case, the regression is s on z. A suitable regression model is then:

$$\boldsymbol{s} = \boldsymbol{A}\boldsymbol{p} + \boldsymbol{\varepsilon}, \quad \boldsymbol{A} = \begin{pmatrix} 1 & z_1 \\ \vdots & \vdots \\ 1 & z_n \end{pmatrix}, \quad \boldsymbol{p} = \begin{pmatrix} s \\ \tan\theta \end{pmatrix}, \quad \mathsf{Var}\left[\boldsymbol{\varepsilon}\right] = \boldsymbol{V}_R, \tag{6.66}$$

since the covariance matrix of s is a very good approximation to the covariance matrix of R. From the fitted parameters, the polar angle θ of the track and the s-values in all layers can be immediately determined, and, if desired, the predicted radial positions of all measurements [43]. If higher precision is needed, the circle and line fits can be iterated.

6.4 Track Quality

6.4.1 Testing the Track Hypothesis

The principal test statistic of the track hypothesis, i.e., of the hypothesis that all measurements in the track are generated by the same charged particle, is the total χ^2 of the track. It is exactly χ^2-distributed if, and only if, the following conditions are met:

1. the track model is exactly linear;
2. the measurement errors are normally distributed with mean zero and have the correct covariance matrix;
3. the material effects are normally distributed and have the correct covariance matrix;
4. the estimator is the LS estimator and thus a linear function of the measurements.

Obviously, these conditions are met very rarely, if ever, in the experiment. In most circumstances, the track model is the linear approximation of a non-linear one; the measurement errors are not strictly normal, and the calibration of their covariance matrix is not perfect; the distribution of the multiple scattering angle has tails that

contradict the assumption of normality; the estimated track parameters may be distorted by outliers; and the estimator may be a robust version of the usual LS estimator. As a consequence, the best one can hope for is that the total χ^2 is at least approximately χ^2-distributed. Its distribution for a sample of tracks can be visualized by a histogram of the p-values, defined by:

$$p = \int_{\chi^2}^{\infty} g_d(x)\mathrm{d}x, \tag{6.67}$$

where $g_d(x)$ is the PDF of the χ^2-distribution with d degrees of freedom. The number of degrees of freedom is the sum M of the measurement dimensions m_i minus the dimension of the track parameter vector. In the ideal case, the p-values are uniformly distributed in the interval [0, 1]. In practice, one frequently observes a fairly uniform distribution with a peak at zero, where defective and fake tracks accumulate.

Besides the total chi-square statistic, the track length and the number of holes or missing measurements is an indication of the track quality. As the number of degrees of freedom of the track fit is the same as the number of geometrical constraints imposed on the measurements, a long track is much less likely to be a fake track than a short track. On the other hand, an outlier has less effect on the total χ^2 in a long track than it has in a short track. This can be demonstrated by a simple example. Assume a perfect sample of tracks with four measurements of dimension two. The fit of five track parameters leaves three degrees of freedom. A χ^2-cut at the 99%-quantile $q_{0.99,3} = 11.345$ rejects 1% of the tracks. Assume that one of measurements is replaced by an outlier, thereby increasing the total χ^2 of every track by 3. Now the same cut rejects 4% of the tracks. Under the same assumptions, but with ten measurements and 15 degrees of freedom, the cut rejects only 2.4% of the tracks with outliers.

If the efficiency of the tracking detectors or sensors is known with good precision, a rigorous test on the allowed number of holes can be constructed. If, for the sake of simplicity, it is assumed that the efficiency ϵ is the same for all sensors contributing hits to a track candidate, and that the occurrence of holes is independent across sensors, the number h of holes in a track with n measurements is distributed according to a binomial distribution:

$$P(h) = \binom{n}{h} (1-\epsilon)^h \epsilon^{n-h}. \tag{6.68}$$

If $\epsilon = 0.98$ and $n = 15$, then $P(1) = 0.23$ and $P(2) = 0.032$, so a single hole is not suspicious at all, and two holes are unlikely, but not impossible. If $n = 6$, then $P(1) = 0.11$ and $P(2) = 0.0055$, so a single hole is possible, but the occurrence of two holes just by chance is very unlikely, in which case the suspicion of a fake track or a contaminated track is well-founded.

6.4.2 Detection of Outliers

In the context of track fitting, an outlier is defined as a measurement that does not follow the expected behaviour. This may be put into statistical terms by saying that a measurement is considered as an outlier whenever its distance from the locally estimated track position is too large under the assumption of normal measurement errors, the distance being expressed in terms of the covariance matrix attached to the measurement.

Outliers can be classified into track-correlated and track-uncorrelated ones [44]. Some sources of track-correlated outliers are:

- *Ambiguous measurements.* Some tracking detectors, in particular drift chambers, give rise to ambiguous information; see also Sect. 1.2.3. The track search, being less restrictive than a rigorous track fit, is not always able to decide which of the two possible solutions is the correct one, and the decision must be deferred to the track fit. In the track fit, the wrong solution is regarded as an outlier which has to be spotted or suppressed.
- *Delta rays.* Delta rays are energetic ionization electrons that leave a trail of secondary ionization in the detector and can cause a shift in the measured position.
- *Cluster merging.* In a silicon sensor or in a gaseous detector, two clusters belonging to particles that are close in space may merge to a single cluster that is biased with respect to both true positions.
- *Cluster decay.* Similarly, a large cluster may decay into two clusters, which are both biased with respect to the true position.
- *Non-normal measurement errors.* Although the bulk of the measurements follows a normal distribution in most tracking detectors, there is nearly always a small fraction of the data that deviate from the normal law. These data show up as long tails in the error distribution and look like outliers.
- *Faulty covariance matrix.* The errors attached to the measurement are too small, because of insufficient calibration, fluctuations of the signal, wrong assumptions about the track angle with respect to the sensor, dead channels, or other detector problems.

Track-uncorrelated outliers are signals that are not caused by the track, but are nevertheless picked up by the track search. They may be, for instance, signals from adjacent tracks, ghost hits in double-sided silicon sensors, see Sect. 1.3.1, or noise.

Whatever the source, an outlier can be detected by a test based on the residuals of the measurements with respect to the estimated track position. In the case of a single outlier, the test is most powerful if the estimate contains the information of all the other measurements. This is done most easily in the state space model of the track; see Sects. 3.2.3 and 6.1.2. Let $\boldsymbol{m}_k$ be measurement under scrutiny, $\boldsymbol{q}_{k|n}$ the smoothed residual and $\boldsymbol{C}_{k|n}$ its covariance matrix, see Eqs. (3.39) and (3.40) and end of Sect. 6.1.2. The compatibility of $\boldsymbol{m}_k$ with $\boldsymbol{q}_{k|n}$ can be checked component-wise on the basis of the standardized residuals, or globally on the basis of the chi-square statistic:

$$\chi^2_{k|n} = \left(\boldsymbol{r}_{k|n}\right)^{\mathsf{T}} \left(\boldsymbol{R}_{k|n}\right)^{-1} \boldsymbol{r}_{k|n}. \tag{6.69}$$

If there are no outliers, the standardized residuals should be compatible with a standard normal distribution, and $\chi^2_{k|n}$ should be compatible with a χ^2-distribution with m_k degrees of freedom, where m_k is the dimension of $\boldsymbol{m}_k$. If $\boldsymbol{m}_k$ is an outlier, this should be visible in the values of $\boldsymbol{r}_{k|n}$ and $\chi^2_{k|n}$. There is, however, a problem with this approach. Even a single outlier at position k introduces a bias in all of the states $\boldsymbol{q}_{i|n}, i \neq k$, so that also an inlier at position $i \neq k$ can show abnormal values of the residuals and $\chi^2_{i|n}$, especially if i is close to k. As a consequence, it is by no means obvious that $\boldsymbol{m}_k$ can be correctly identified as the outlier. The situation is even worse if there are several outliers. In this case, a robust track fit that down-weights outliers, instead of trying to find and remove them, is a better solution; see Sect. 6.2.

6.4.3 Kink Finding

A charged particle decay that produces a single charged daughter particle plus some neutral ones manifests itself as a sudden change of the track direction and/or curvature, often called a kink or breakpoint.[1] Typical examples are the muonic decays of charged π and K mesons. Another source of kinks is hard elastic scattering on the material of the detector [45]. Collinear energy loss of an electron by bremsstrahlung does not result in a kink, but only in a change of curvature.

It is characteristic for a kink that the track segments in front of and behind the kink both give a good fit to their respective track model; however, there is a significant difference between the two sets of track parameters estimated from the two track segments. In the Kalman filter framework, this difference and its covariance matrix are readily available at any layer k from the forward and the backward filter:

$$\boldsymbol{\Delta}_k = \boldsymbol{q}_k - \tilde{\boldsymbol{q}}^{\,\mathrm{b}}_{k|k+1}, \quad \mathsf{Var}\left[\boldsymbol{\Delta}_k\right] = \boldsymbol{C}_{\Delta,k} = \boldsymbol{C}_k + \boldsymbol{C}^{\,\mathrm{b}}_{k|k+1}. \tag{6.70}$$

The associated chi-square statistic is given by:

$$\chi^2_{\Delta,k} = \boldsymbol{\Delta}_k^{\mathsf{T}} \boldsymbol{C}_{\Delta,k}{}^{-1} \boldsymbol{\Delta}_k. \tag{6.71}$$

In [44] a χ^2-test statistic X^2 for kink finding is investigated:

$$X^2 = \max_{k \in K} \chi^2_{\Delta,k}, \tag{6.72}$$

[1] In the literature on time series analysis it is called a change point.

where the range K of layers is restricted by the requirement that the respective track segments from the forward and the backward filter are well defined. Results from the simulation of π and K decays in a simplified setup are given in [44].

This simple test does not take into account the specific features of the process leading to the kink. In energy loss by bremsstrahlung, only the curvature changes; in hard elastic scattering, only the direction changes; in a decay, the direction changes and the momentum decreases, or curvature increases. In all three processes, the position of the two track segments has to be compatible. These features are taken into account by the modified track fit described in [45]. At each possible breakpoint, an extended set of track parameters $\boldsymbol{\alpha}$ is defined that allows for sudden changes in a subset of the track parameters. Three cases are considered:

1. *Energy loss by bremsstrahlung.* The curvature is allowed to change; therefore, $\boldsymbol{\alpha}$ contains two curvature parameters instead of one, for instance, κ_f and κ_b.
2. *Hard elastic scattering.* Only the direction is allowed to change; therefore, $\boldsymbol{\alpha}$ contains two sets of direction parameters instead of one, for instance $\tan\lambda_\mathrm{f}$, ϕ_f and $\tan\lambda_\mathrm{b}$, ϕ_b.
3. *Muonic decays of π and K mesons.* Direction and curvature are allowed to change; therefore, $\boldsymbol{\alpha}$ contains two sets of the corresponding parameters instead of one, for instance, $\tan\lambda_\mathrm{f}$, ϕ_f, κ_f and $\tan\lambda_\mathrm{b}$, ϕ_b, κ_b.

At layer k, $\boldsymbol{\alpha}$ can be estimated by a linear regression in which $\tilde{\boldsymbol{q}}_k$ and $\tilde{\boldsymbol{q}}^{\,\mathrm{b}}_{k|k+1}$ play the role of the observations. This is equivalent to the minimization of the following objective function:

$$\begin{aligned}\mathcal{S}(\boldsymbol{\alpha}) = &\left(\tilde{\boldsymbol{q}}_k - \boldsymbol{H}_\mathrm{f}\boldsymbol{\alpha}\right)^\mathsf{T} \boldsymbol{C}_k{}^{-1} \left(\tilde{\boldsymbol{q}}_k - \boldsymbol{H}_\mathrm{f}\boldsymbol{\alpha}\right) + \\ &+ \left(\tilde{\boldsymbol{q}}^{\,\mathrm{b}}_{k|k+1} - \boldsymbol{H}_\mathrm{b}\boldsymbol{\alpha}\right)^\mathsf{T} \left(\boldsymbol{C}^{\,\mathrm{b}}_{k|k+1}\right)^{-1} \left(\tilde{\boldsymbol{q}}^{\,\mathrm{b}}_{k|k+1} - \boldsymbol{H}_\mathrm{b}\boldsymbol{\alpha}\right), \end{aligned} \tag{6.73}$$

where $\boldsymbol{H}_\mathrm{f}$ and $\boldsymbol{H}_\mathrm{b}$ are the matrices that project $\boldsymbol{\alpha}$ on $\tilde{\boldsymbol{q}}_k$ and $\tilde{\boldsymbol{q}}^{\,\mathrm{b}}_{k|k+1}$, respectively.

From the estimated vector $\boldsymbol{\alpha}$ and its covariance matrix, standardized forward-backward differences of the relevant parameters can be computed. In addition an F-test can be performed to test whether additional parameters result in a significant reduction of the total chi-square statistic. The location of the breakpoint can be determined by the location of the largest discrepancy between forward and backward parameters, as measured by the value of $\mathcal{S}(\boldsymbol{\alpha})$ at the minimum. Results of studies of simulated pion decays in the NOMAD detector are shown in [45].

The breakpoint finder in [46] is based on the autocorrelation function of the residuals of the track fit. In an undisturbed track with many measurements, typically in a TPC, the residuals between the measured coordinates and the fitted trajectory are only weakly correlated so that the autocorrelation function of the residuals is close to zero for arbitrary lags. A breakpoint in the track introduces correlated shifts in all subsequent position measurements, resulting in an autocorrelation function that is significantly different from zero. Assuming 1D position measurements, the average autocorrelation of lag ℓ is defined by:

$$\rho_\ell = \left(\sum_{i=1}^{n-\ell} r_i r_{i+\ell}\right) \left(\sum_{i=1}^{n-\ell} {r_i}^2 \sum_{i=1}^{n-\ell} r_{i+\ell}^2\right)^{-1/2}, \tag{6.74}$$

where $r_i = \delta_i/\sigma_i$ is the residual of measurement i divided by the standard error of measurement i, and n is the total number of measurements. The test statistic λ used in [46] is a weighted average of the autocorrelations up to a maximal lag L such that small lags have larger weight:

$$\lambda = \sum_{\ell=1}^{L} w_\ell \, \rho_\ell, \quad w_\ell = \frac{2\,(L-\ell)}{L\,(L-1)}, \quad \sum_{\ell=1}^{L} w_\ell = 1. \tag{6.75}$$

For the simulated data used in [46], setting L equal to the nearest integer to $n/8$ gives the largest power of the test. In general, the maximal lag L and the weights w_ℓ must be tuned on simulated data. The threshold of λ above which the null hypothesis (no breakpoint) is rejected, is set according to the tolerated percentage of undisturbed tracks that are rejected, i.e., to the probability of an error of the first kind.

References

1. R. Frühwirth, Nucl. Instrum. Methods Phys. Res. A **324**, 317 (1993)
2. P. Billoir, Nucl. Instrum. Methods **225**(2), 352 (1984)
3. R. Frühwirth, Nucl. Instrum. Methods Phys. Res. A **262**, 444 (1987)
4. P. Billoir, R. Frühwirth, M. Regler, Nucl. Instrum. Methods Phys. Res. A **241**, 115 (1985)
5. R. Frühwirth et al., *Data Analysis Techniques for High-Energy Physics*, 2nd edn. (Cambridge University Press, Cambridge, 2000)
6. C. Kleinwort, Nucl. Instrum. Methods Phys. Res. A **673**, 107 (2012)
7. V. Blobel, C. Kleinwort, F. Meier, Comput. Phys. Commun. **182**(9), 1760 (2011)
8. C. Kleinwort, General Broken Lines. http://www.terascale.de/wiki/generalbrokenlines
9. C. Höppner et al., Nucl. Instrum. Methods Phys. Res. A **620**(2), 518 (2010)
10. J. Rauch, T. Schlüter, J. Phys.: Conf. Ser. **608**(1), 012042 (2015). https://doi.org/10.1088/1742-6596/608/1/012042
11. J. Rauch, T. Schlüter, GENFIT—A Generic Track-Fitting Toolkit (2016). https://arxiv.org/abs/1410.3698v2
12. GENFIT—a generic toolkit for track reconstruction for experiments in particle and nuclear physics. http://genfit.sourceforge.net
13. T. Bilka et al., Nucl. Instrum. Methods Phys. Res. A (2019, Submitted) https://arxiv.org/pdf/1902.04405.pdf
14. N. Berger et al., Nucl. Instrum. Methods Phys. Res. A **844**, 135 (2017)
15. M. Kiehn, Pixel Sensor Evaluation and Track Fitting for the Mu3e Experiment. Ph.D. thesis, Universität Heidelberg (2016). http://www.ub.uni-heidelberg.de/archiv/20493
16. E. Clement et al., A High-Performance Track Fitter for Use in Ultra-fast Electronics. Technical Report FERMILAB-PUB-18-496-PPD-TD, FERMILAB (2018). https://arxiv.org/pdf/1809.01467.pdf
17. M. Shochet et al., Fast TracKer (FTK) Technical Design Report. Technical Report ATLAS-TDR-021, CERN, Geneva (2013). http://cds.cern.ch/record/1552953
18. P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection* (Wiley, Hoboken, 1987)
19. P.J. Huber, E.M. Ronchetti, *Robust Statistics*, 2nd edn. (Wiley, Hoboken, 2009)

20. F. Mosteller, J.W. Tukey, *Data Analysis and Regression: A Second Course in Statistics* (Addison-Wesley Publishing Company, Reading, 1977)
21. R. Frühwirth, W. Waltenberger, Austrian J. Stat. **37**(3&4), 301 (2008). https://tinyurl.com/RedescendingMestimators
22. R. Frühwirth, A. Strandlie, Comput. Phys. Commun. **120**, 197 (1999)
23. M. Winkler, A comparative study of track reconstruction methods in the context of CMS physics. Ph.D. thesis, TU Wien (2002). http://cds.cern.ch/record/583851
24. E. Brondolin, Track reconstruction in the CMS experiment for the high luminosity LHC. Ph.D. thesis, Technische Universität Wien (2018). http://cds.cern.ch/record/2308020
25. R. Frühwirth, Comput. Phys. Commun. **100**, 1 (1997)
26. R. Frühwirth, M. Regler, Nucl. Instrum. Methods Phys. Res. A **456**(3), 369 (2001)
27. R. Frühwirth, M. Liendl, Comput. Phys. Commun. **141**, 230 (2001)
28. A. Strandlie, R. Frühwirth, IEEE Trans. Nucl. Sci. **53**, 3842 (2006)
29. R. Frühwirth, Comput. Phys. Commun. **154**(2), 131 (2003)
30. W. Adam, R. Frühwirth, A. Strandlie, T. Todorov, J. Phys. G: Nucl. Part. Phys. **31**(9), N9 (2005)
31. R. Frühwirth, A. Strandlie, Nucl. Instrum. Methods Phys. Res. A **559**, 162 (2006)
32. The ATLAS Collaboration, Improved electron reconstruction in ATLAS using the Gaussian Sum Filter-based model for bremsstrahlung. Technical Report ATLAS-CONF-2012-047, CERN, Geneva (2012). https://cds.cern.ch/record/1449796
33. ATLAS Collaboration, Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton-proton collision data at $\sqrt{s} = 13$ TeV. Technical Report CERN-EP-2018-273, CERN, Geneva (2019). https://cds.cern.ch/record/2657964
34. M. Hansroul, H. Jeremie, D. Savard, Nucl. Instrum. Methods Phys. Res. A **270**(2), 498 (1988)
35. R.L. Gluckstern, Nucl. Instrum. Methods **24**, 381 (1963)
36. N. Chernov, G. Ososkov, Comput. Phys. Commun. **33**, 329 (1984)
37. V. Karimäki, Nucl. Instrum. Methods Phys. Res. A **305**, 187 (1991)
38. B. Lillekjendlie, Comput. Vis. Image Underst. **67**, 311 (1997)
39. A. Strandlie, J. Wroldsen, R. Frühwirth, B. Lillekjendlie, Comput. Phys. Commun. **131**, 95 (2000)
40. A. Strandlie, J. Wroldsen, R. Frühwirth, Nucl. Instrum. Methods Phys. Res. A **488**, 332 (2002)
41. A. Strandlie, R. Frühwirth, Nucl. Instrum. Methods Phys. Res. A **480**, 734 (2002)
42. R. Frühwirth, A. Strandlie, J. Phys.: Conf. Ser. **1085**(4), 042004 (2018). https://doi.org/10.1088/1742-6596/1085/4/042004
43. R. Frühwirth, A. Strandlie, W. Waltenberger, Nucl. Instrum. Methods Phys. Res. A **490**, 366 (2002)
44. R. Frühwirth, Application of Filter Methods to the Reconstruction of Tracks and Vertices in Events of Experimental High Energy Physics. Ph.D. thesis, TU Wien (1988). http://www.ub.tuwien.ac.at/diss/VL/AC00099086.pdf
45. P. Astier et al., Nucl. Instrum. Methods Phys. Res. A **450**(1), 138 (2000)
46. R. Kowalewski, P. Jackson, Nucl. Instrum. Methods Phys. Res. A **457**(3), 640 (2001)

Part III
Vertex Reconstruction

Chapter 7
Vertex Finding

Abstract Vertex finding is the search for clusters of tracks that originate at the same point in space. The chapter discusses a variety of methods for finding primary vertices, first in one and then in three dimensions. Details are given on model-based clustering, the EM algorithm and clustering by deterministic annealing in 1D, and greedy clustering, iterated estimators, topological vertex finding, and a vertex finder based on medical imaging in 3D.

7.1 Introduction

Vertex finding is the process of dividing all or a subset of the reconstructed tracks in an event into classes such that presumably all tracks in a class are produced at the same vertex. Vertices in an event can be classified as primary vertices or secondary vertices. In a fixed target experiment, a primary vertex is the point where a beam particle collides with a target particle; in a collider experiment, a primary vertex is the point where two beam particles collide. In the LHC experiments CMS and ATLAS, there are many primary vertices in each event because in each bunch crossing many collisions may and do occur. As a rule, however, at most one of these primary vertices is of interest to the subsequent analysis; this is called the signal vertex. The signal vertex is distinguished from the other primary vertices (pile-up vertices) by using kinematic criteria such as the transverse momenta of the tracks forming the vertex.

A secondary vertex is the point where an unstable particle decays, or where a particle interacts with the material of the detector. As the search for secondary vertices is often based on a well-reconstructed primary vertex, this chapter deals only with primary vertex finding; secondary vertex finding is deferred to Chap. 9.

In collider experiments, the position of the beam spot and the size of the luminous region contains precise prior information about the transverse position of primary vertices, see Fig. 7.1. The prior information on the position along the beam axis is much weaker, as the primary vertices can be anywhere in a zone defined by the bunch length, which is a couple of centimeters in the LHC. In fixed-target

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_7

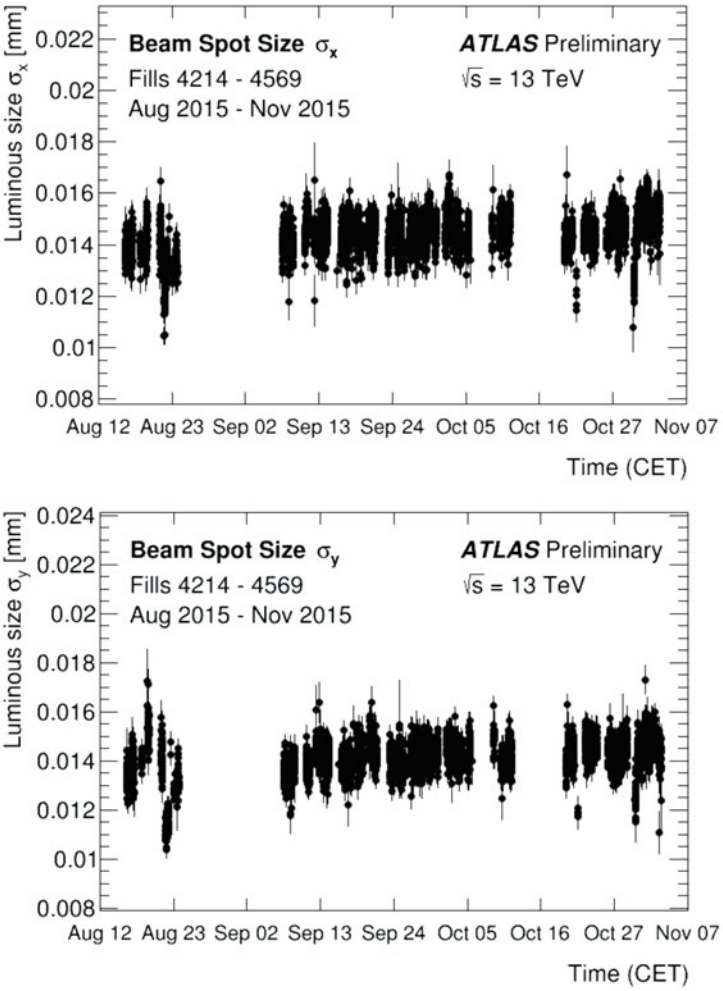

Fig. 7.1 The transverse size of the luminous region of the LHC as determined by ATLAS in 2017 [1]. Left: horizontal size; right: vertical size

experiments, the prior information is given by the beam profile and the position and size of the target.

Vertex finding methods can be roughly divided into three main types: generic clustering algorithms, topological methods, and iterated estimators. The latter can be considered as a special model-based clustering method. For a brief introduction into methods of clustering, see Sect. 3.3.

7.2 Primary Vertex Finding in 1D

In the LHC experiments ATLAS and CMS, there are many beam-beam interactions during a single bunch crossing at the typical luminosity level of the collider. In order to find all primary vertices, first primary tracks are selected by a cut on their distance to the z-axis, which is the beam line. The selected tracks are then clustered on the basis of their z-coordinates at their point of closest approach to the centre of the beam spot. The vertex finding problem is thus reduced to clustering in a single spatial dimension.

7.2.1 Divisive Clustering

A simple divisive clustering can be performed by requiring a gap of at least d between adjacent clusters/vertices. The threshold d depends on the shape of the beam profile along z, on the expected number of interactions per bunch crossing, and on the precision of the z-coordinate used for clustering. It has to be optimized by studying simulated events. If subsequent validation of a cluster fails, it can be further divided by a more refined method (see Sect. 7.3.3).

7.2.2 Model-Based Clustering

Assume that there are n tracks with z-coordinates $z_i,\ i = 1, \ldots, n$, sorted in ascending order: $z_1 < z_2 < \ldots < z_n$. The z_i are assumed to be sampled from a Gaussian mixture with the following PDF:

$$f(z) = \sum_{k=1}^{K} \omega_k \varphi(z; v_k, \sigma_k^2), \quad \sum_{k=1}^{K} \omega_k = 1, \tag{7.1}$$

where K is the number of mixture components, φ is the PDF of the normal distribution, ω_k is the component weight, v_k is the mean value, and σ_k^2 is the variance of component $k,\ k = 1, \ldots, K$. As the association of the tracks represented by the points z_i to the vertices represented by the component means v_k is unknown, latent (unobserved) variables $y_i,\ i = 1, \ldots, n$ are introduced that encode the association:

$$y_i = k \iff z_i \text{ belongs to component } k,\ i = 1, \ldots, n. \tag{7.2}$$

The latent variables and the unknown parameters of the mixture (weights, means, variances) can be estimated by the Expectation-Maximization (EM) algorithm [2–4]. The EM algorithm is iterative, and each iteration consists of two steps. In the E-step (expectation step), the latent variables are estimated, given the observations and

the current estimate of the mixture parameters. In the M-step (maximization step), the maximum likelihood estimate of the mixture parameters is computed, using the estimates of the latent variables from the E-step. Convergence is guaranteed as the likelihood increases in every iteration. There is no guarantee, however, to reach the global maximum of the likelihood function.

In the special case of a normal mixture, explicit formulas can be obtained. Assume that the M-step of iteration j gives the estimates $\omega_k(j), v_k(j), \sigma_k^2(j),\ k = 1, \ldots, K$. The E-step of the next iteration $j+1$ computes the association probabilities or ownership weights $p_{i,k}$ of all points with respect to all components:

$$p_{i,k} = \frac{\omega_k(j)\,\varphi(z_i; v_k(j), \sigma_k^2(j))}{\sum_{l=1}^{K} \omega_l(j)\,\varphi(z_i; v_l(j), \sigma_l^2(j))},\ i = 1, \ldots, n,\ k = 1, \ldots, K. \tag{7.3}$$

In the M-step the mixture parameters are updated:

$$\begin{aligned} \omega_k(j+1) &= \frac{1}{n} \sum_{i=1}^{n} p_{i,k}, \\ v_k(j+1) &= \frac{\sum_{i=1}^{n} p_{i,k}\, z_i}{\sum_{i=1}^{n} p_{i,k}}, \\ \sigma_k^2(j+1) &= \frac{\sum_{i=1}^{n} p_{i,k}\, [z_i - v_k(j+1)]^2}{\sum_{i=1}^{n} p_{i,k}}, \quad k = 1, \ldots, K. \end{aligned} \tag{7.4}$$

The EM algorithm suffers from the fact that the number K of clusters has to be selected in advance. A possible solution to this problem is to set K to the largest value that can reasonably be expected, and to merge components that are sufficiently similar after the EM algorithm has converged.

The selection of the optimal number of components can be automatized by sparse model-based clustering [5, 6]. Sparsity of the final mixture is achieved by an appropriate prior on the mixture weights. As explicit formulas are no longer available, estimation of the mixture parameters has to be done via Markov Chain Monte Carlo (MCMC). A comparison with the EM algorithm can be found in [6].

7.2.3 EM Algorithm with Deterministic Annealing

Assume that there are n tracks with z-coordinates $z_i,\ i = 1, \ldots, n$, again in ascending order, with their associated standard errors $\sigma_i,\ i = 1, \ldots, n$. The EM algorithm described in Sect. 7.2.2 is sensitive to the initial values of the component parameters. This can be cured by introducing deterministic annealing (DA), see [7]. DA introduces a temperature parameter T, which is used to scale the standard errors of the $z_i,\ i = 1, \ldots, n$. Annealing starts at high temperature, corresponding to

large errors, and the temperature is lowered according to a predefined annealing schedule. At each temperature, the association probabilities of the data points z_i to the current cluster centers are computed. If there are data points not associated to any of the clusters, indicated by very low association probabilities, one of these points is chosen as a new cluster center. The number of clusters is thus determined dynamically. The algorithm is summarized in Table 7.1. Note that in contrast to the model-based clustering in Sect. 7.2.2 the association probabilities are computed using the uncertainties of the track positions instead of the vertex positions.

Two examples with ten randomly generated clusters within a space of 1 cm and $\sigma_i = 0.01$ cm are shown in Fig. 7.2. In the example on the top, the found clusters perfectly match the true clusters while in the example on the bottom, two true clusters merge into a single found cluster, and one of the found clusters is spurious with a single data point.

7.2.4 *Clustering by Deterministic Annealing*

Assume further that there are K vertex positions v_k, $k = 1, \ldots, K$, called prototypes, that represent K clusters of tracks. The expected discrepancy between data points and prototypes is given by:

Table 7.1 Algorithm: Vertex finding with EM algorithm and deterministic annealing

CLUSTER FINDING WITH EM ALGORITHM AND DETERMINISTIC ANNEALING

1. Choose an initial temperature $T_0 > 1$, a cooling schedule $T_0, \ldots, T_m < 1$, a threshold δ, and a robustness constant $c \in [2, 3]$.
2. Select the smallest data point z_1 as the first cluster center v_1, set $K = 1,\ t = 0$, and $T = T_t$.
3. Compute the association probabilities $p_{i,k}$ according to:

$$p_{i,k} = \frac{\exp\left[-\beta\,(z_i - v_k)^2/\sigma_i^2\right]}{\exp\left(-\beta\,c^2\right) + \sum_{j=1}^{K} \exp\left[-\beta\,(z_i - v_j)^2/\sigma_i^2\right]}, \quad \text{with } \beta = 1/T. \tag{7.5}$$

4. Update the cluster centers:

$$v_k = \frac{\sum_{i=1}^{n} p_{i,k}\, z_i}{\sum_{i=1}^{n} p_{i,k}},\ k = 1, \ldots, K. \tag{7.6}$$

5. For each data point, compute $p_i = \max_k p_{i,k}$ and find the index j of the smallest p_i. If $p_j < \delta$, set $K := K + 1,\ v_K = z_j$ and go to 3.
6. *Optional*: Split "large" clusters by checking their width or their multiplicity. Assign data point z_i to cluster k, if $p_{i,k} > 0.5$. If cluster k is to be split, set $K := K + 1$, set v_k to the smallest data point in the cluster, set v_K to the largest data point in the cluster, and go to 3.
7. If $t < m$, set $t := t + 1,\ T = T_t$, and go to 3.
8. Assign data point z_i to cluster k, if $p_{i,k} > 0.9,\ i = 1, \ldots, n$.

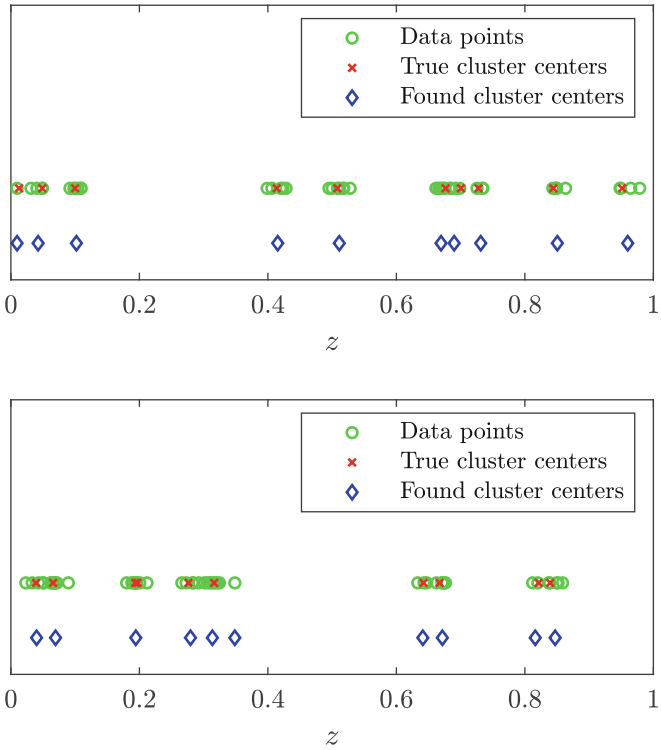

Fig. 7.2 Examples of cluster finding with EM algorithm and Deterministic Annealing. The results are discussed in the text

$$E = \sum_{i=1}^{n} \sum_{k=1}^{K} P(z_i \in C_k)\, d(z_i, v_k), \tag{7.7}$$

where C_k is the cluster represented by v_k and $d(z_i, v_k)$ is a measure of distance between data point z_i and the prototype v_k [8]. A typical choice of $d(z_i, v_k)$ is the weighted squared difference:

$$d(z_i, v_k) = \left(\frac{z_i - v_k}{\sigma_i} \right)^2, \tag{7.8}$$

where σ_i is again the standard error of track position z_i. As there is no prior knowledge on the probabilities $P(z_i \in C_k)$, the principle of maximum entropy can be applied, giving a Gibbs distribution:

$$P(z_i \in C_k) = \frac{\exp\left(-\beta\, d(z_i, v_k)\right)}{\sum_{l=1}^{K} \exp\left(-\beta\, d(z_i, v_l)\right)} \tag{7.9}$$

The parameter $\beta = 1/T$ is the inverse of the "temperature" of the system.

Finding the most probable configuration of the prototypes at a given temperature is equivalent to minimizing the "free energy" F, which is given by [8]:

$$F = -\frac{1}{\beta} \sum_{i=1}^{n} \ln \sum_{k=1}^{K} \exp\left(-\beta\, d(z_i, v_k)\right). \tag{7.10}$$

Minimization of F has to be done by numerical methods; see Sect. 3.1.

At infinite temperature, i.e., $\beta = 0$, there is a single cluster containing all data points. The temperature is now lowered according to a predefined annealing schedule. At some positive value of β, the cluster will undergo a phase transition and split into smaller clusters. Annealing is continued until the end of the schedule. At every temperature, a characteristic number of effective prototypes emerges at distinct positions, independent of the number of prototypes.

Instead of using a large number of prototypes, many of which may coincide at a given temperature, one can work with weighted prototypes, where the weight ρ_k corresponds to the fraction of unweighted prototypes that coincide at v_k. The weights ρ_k always sum to 1, and the free energy is slightly modified:

$$F = -\frac{1}{\beta} \sum_{i=1}^{n} \ln \sum_{k=1}^{K} \rho_k \exp\left(-\beta\, d(z_i, v_k)\right). \tag{7.11}$$

The association probabilities and the cluster weight are given by:

$$p_{i,k} = P(z_i \in C_k) = \frac{\rho_k \exp\left(-\beta\, d(z_i, v_k)\right)}{\sum_{l=1}^{K} \rho_l \exp\left(-\beta\, d(z_i, v_l)\right)}, \quad \rho_k = \frac{1}{n} \sum_{i=1}^{n} p_{i,k}. \tag{7.12}$$

The annealing is started at high temperature with a single prototype of weight $\rho_1 = 1$. If the distance function is chosen as in Eq. (7.8), the minimum of F is at the weighted mean of the data points. During annealing, the temperature is gradually decreased and the local minima of F emerge. The critical temperature T_k^{c} of cluster k is the point where a local minimum turns into a saddle point and is given by:

$$T_k^{\mathrm{c}} = 2 \sum_{i=1}^{n} \frac{p_{i,k}}{\sigma_i^2} \left(\frac{z_i - v_k}{\sigma_i}\right)^2 \Bigg/ \sum_{i=1}^{n} \frac{p_{i,k}}{\sigma_i^2}. \tag{7.13}$$

Whenever the temperature falls below the critical temperature of a cluster, the prototype of this cluster is replaced by two nearby representatives. The association probabilities and the new cluster weights are recomputed according to Eq. (7.12). If the final temperature is small enough, the soft assignments of data points to clusters turns into a hard assignment.

7.3 Primary Vertex Finding in 3D

In principle, the clustering methods described above for vertex finding in 1D can also be applied to vertex finding in 3D. It has to be noted, though, that the shortest distance in space between two tracks is peculiar insofar as it does not satisfy the triangle inequality: if tracks a and b are close, and tracks b and c are close, it does not follow that tracks a and c are close as well. The distance between two clusters of tracks should therefore be defined as the maximum of the individual pairwise distances, known as complete linkage in the clustering literature. Alternatively, the distance between two clusters can be the distance between the two vertices fitted from the clusters.

7.3.1 Preclustering

Because of the high track and vertex multiplicity in typical collider experiments at the LHC, preliminary clusters of tracks can be formed by selecting a primary vertex or a small group of primary vertices found in 1D. All tracks that are compatible with these are put in a preliminary cluster. This reduces the combinatorics and opens the possibility of processing these preliminary clusters in parallel. Tracks that are not compatible with any primary vertex are reserved for secondary vertex finding. In low-multiplicity experiments, this preliminary clustering can be omitted.

7.3.2 Greedy Clustering

Greedy clustering is agglomerative and starts with a single track, preferably a high-quality track with many hits and good χ^2 (see Sect. 6.4.1). It is combined with its nearest neighbour in 3D, and a vertex is fitted from the two tracks. If the fit is successful, the vertex is stored. The track nearest to vertex is added, for instance, by means of an extended Kalman filter (see Sect. 8.1.2.2). This procedure is continued until the vertex fit fails. Clustering is then resumed with an unused track.

The greedy clustering does not guarantee the globally best assignment of tracks to vertices, as tracks that are attached to a vertex remain attached forever. This can be cured by using a robust vertex fit throughout (see Sect. 8.2), allowing a track to be removed from a vertex if it is tagged as an outlier.

7.3.3 Iterated Estimators

This is a divisive clustering algorithm with the following steps:

1. Perform a (preferably robust) vertex fit with all tracks.
2. Discard all incompatible tracks.
3. Repeat step 1 with all discarded tracks.

The iteration stops when no vertex with at least two tracks can be successfully fitted. Step 2 might itself be iterative, especially if the vertex fit is not robust, so that the incompatible tracks have to be removed sequentially. An iterative vertex finder, based on an adaptive fit (Sect. 8.2.2) and called the Adaptive Vertex Reconstructor (AVR, [9]) is implemented in the RAVE toolbox [10, 11]; see Appendix C.

7.3.4 Topological Vertex Finder

A general topological vertex finder called ZVTOP was proposed in [12]. It is related to the Radon transform, which is a continuous version of the Hough transform used for track finding (see Sect. 5.1.2). The search for vertices is based on a function $V(\boldsymbol{v})$ that quantifies the probability of a vertex at location $\boldsymbol{v}$. For each track a Gaussian probability tube $f_i(\boldsymbol{v})$ is constructed. The function $V(\boldsymbol{v})$ is defined taking into account that the value of $f_i(\boldsymbol{v})$ must be significant for at least two tracks:

$$V(\boldsymbol{v}) = \sum_{i=1}^{n} f_i(\boldsymbol{v}) - \frac{\sum_{i=1}^{n} f_i^2(\boldsymbol{v})}{\sum_{i=1}^{n} f_i(\boldsymbol{v})}$$

Due to the second term on the right-hand side, $V(\boldsymbol{v}) \approx 0$ in regions where $f_i(\boldsymbol{v})$ is significant for only one track. The form of $V(\boldsymbol{v})$ can be modified to fold in known physics information about probable vertex locations. For instance, $V(\boldsymbol{v})$ can be augmented by a further function $f_0(\boldsymbol{v})$ describing the location and spread of the interaction point. In addition, $V(\boldsymbol{v})$ may be modified by a factor dependent on the angular location of the point $\boldsymbol{v}$.

Vertex finding amounts to finding the local maxima of the function $V(\boldsymbol{v})$. The search starts at the calculated maxima of the products $f_i(\boldsymbol{v}) f_j(\boldsymbol{v})$ for all track pairs. For each of these points, the nearest maximum of $V(\boldsymbol{v})$ is found. As $V(\boldsymbol{v})$ is a smooth function, any of the methods discussed in Sect. 3.1 can be employed. The found maxima are clustered together to form candidate vertex regions. The final association of the tracks to the vertex candidates can be done on the basis of the respective χ^2 contributions or by an adaptive fit (see Sect. 8.2.2). An experimental application is described in [13].

In [14], the topological vertex finder was augmented by a procedure based on the concept of the minimum spanning tree of a graph. For each track, the bins crossed by the track (or a tangent to the track at the point of closest approach) are incremented by one.

7.3.5 *Medical Imaging Vertexer*

The Medical Imaging Vertexer (MIV) [15] is similar to ZVTOP, but differs from it in two points: first, it works with a pixelized representation of the track density; second, it applies a medical imaging filter to the density before finding the maxima. The vertex finder can be summarized in the following steps:

1. All tracks are back-projected into a volume to be searched for vertices. The volume is represented by a 3D histogram. The bin size of the histogram is comparable to the tube size in ZVTOP.
2. The histogram is transformed into Fourier space, filtered by a medical imaging filter that removes artifacts and reduces blurring, and transformed back to a histogram in 3D space.
3. The filtered histogram is searched for local maxima. Clustering starts with the highest bin. If the next highest bin is adjacent it is added to the cluster, otherwise it is the seed for the next cluster. This is iterated until no bin is above a predefined threshold.
4. Clusters are split or merged using a resolution criterion similar to the one used in ZVTOP [12].
5. A cluster is accepted as a vertex candidate if its starting bin exceeds a predefined threshold. The vertex position is estimated as the center of gravity of the cluster.

The performance of the MIV has been studied and compared to the Adaptive Vertex Reconstructor (AVR) in [15]. It is shown that the MIV finds vertices with higher efficiency and higher purity at large pile-up, whereas the AVR performs better at small pile-up.

References

1. ATLAS Collaboration, Beam Spot Public Results (2017). https://tinyurl.com/ATLAS-Beamspot
2. A.P. Dempster, N.M. Laird, D.B. Rubin, J. R. Stat. Soc. Series B **39**(1), 1 (1977)
3. J.A. Bilmes, A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report TR-97-021, International Computer Science Institute (1998). http://www.icsi.berkeley.edu/ftp/global/pub/techreports/1997/tr-97-021.pdf
4. S. Borman, The Expectation Maximization Algorithm—A short tutorial. Technical Report, University of Utah (2004). https://tinyurl.com/BormanEMTutorial
5. G. Malsiner-Walli, S. Frühwirth-Schnatter, B. Grün, Stat. Comput. **26**(1), 303 (2016)
6. R. Frühwirth, K. Eckstein, S. Frühwirth-Schnatter, J. Phys. Conf. Ser. **762**(1), 012055 (2016). https://iopscience.iop.org/article/10.1088/1742-6596/762/1/012055
7. N. Ueda, R. Nakano, Neural Netw. **11**(2), 271 (1998)
8. K. Rose, E. Gurewitz, G.C. Fox, IEEE Trans. Pattern Anal. Mach. Intell. **15**, 785 (1993)
9. W. Waltenberger, Adaptive Vertex Reconstruction. Technical Report CMS-NOTE-2008-033, CERN, Geneva (2008). https://cds.cern.ch/record/1166320

10. W. Waltenberger, F. Moser, in *IEEE Nuclear Science Symposium Conference Record 2006*, vol. 1 (IEEE, 2006), p. 104
11. W. Waltenberger, W. Mitaroff, F. Moser, Nucl. Instrum. Meth. Phys. Res. A **581**, 549 (2007)
12. D.J. Jackson, Nucl. Instrum. Meth. Phys. Res. A **388**, 247 (1997)
13. D. Bailey, et al., Nucl. Instrum. Meth. Phys. Res. A **610**(2), 573 (2009)
14. S. Hillert, ZVMST: A Minimum spanning tree-based vertex finder. Technical report, University of Oxford (2008). https://arxiv.org/pdf/0811.3904.pdf
15. S. Hageböck, E. von Toerne, J. Phys. Conf. Ser. **396**(2), 022021 (2012). https://iopscience.iop.org/article/10.1088/1742-6596/396/2/022021

Chapter 8
Vertex Fitting

Abstract The methods used for vertex fitting are closely related to the ones used in track fitting. The chapter describes least-squares estimators as well as robust and adaptive estimators. Furthermore, it is shown how the vertex fit can be extended to a kinematic fit by imposing additional constraints on the tracks participating in the fit.

8.1 Least-Squares Fitting

8.1.1 Straight Tracks

8.1.1.1 Exact Fit

Assume that there are n straight tracks that have to be fitted to a common vertex. Track i is given by a point $\boldsymbol{r}_i$, a unit direction vector $\boldsymbol{a}_i$, and the joint covariance matrix $\boldsymbol{V}_i$ of $\boldsymbol{q}_i = (\boldsymbol{r}_i; \boldsymbol{a}_i),\ i = 1, \ldots, n$. The rank of $\boldsymbol{V}_i$ is usually equal to five. Here and in the entire chapter, it is assumed that there is no material between the vertex and the point or surface where the track parameters are defined.

The estimated common vertex is the point $\boldsymbol{v}$ that minimizes the sum of the weighted squared distances from the tracks. The squared distance D_i of track i from the point $\boldsymbol{v}$ is given by:

$$D_i(\boldsymbol{v}) = [(\boldsymbol{r}_i - \boldsymbol{v}) \times \boldsymbol{a}_i]^2. \tag{8.1}$$

For a given vertex $\boldsymbol{v}_0$, the variance $\sigma_i^2 = \mathsf{var}[D_i]$ is computed by linearized error propagation. The Jacobian of D_i with respect to $\boldsymbol{q}_i$ is given by:

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_8

$$\boldsymbol{J}_i = \begin{pmatrix} \dfrac{\partial D_i}{\partial \boldsymbol{r}_i} \\ \dfrac{\partial D_i}{\partial \boldsymbol{a}_i} \end{pmatrix}, \quad \frac{\partial D_i}{\partial \boldsymbol{r}_i} = 2 \cdot \begin{pmatrix} a_{i,2}\,\eta_{i,21} - a_{i,3}\,\eta_{i,13} \\ a_{i,3}\,\eta_{i,32} - a_{i,1}\,\eta_{i,21} \\ a_{i,1}\,\eta_{i,13} - a_{i,2}\,\eta_{i,32} \end{pmatrix},$$

$$\frac{\partial D_i}{\partial \boldsymbol{a}_i} = 2 \cdot \begin{pmatrix} d_{i,3}\,\eta_{i,13} - d_{i,2}\,\eta_{i,21} \\ d_{i,1}\,\eta_{i,21} - d_{i,3}\,\eta_{i,32} \\ d_{i,2}\,\eta_{i,32} - d_{i,1}\,\eta_{i,13} \end{pmatrix}, \tag{8.2}$$

with the auxiliary variables

$$d_{i,k} = r_{i,k} - v_{0,k}, \quad \eta_{i,jk} = a_{i,j}\,d_{i,k} - a_{i,k}\,d_{i,j}, \quad j,k = 1,2,3.$$

It follows that

$$\sigma_i^2 \approx \boldsymbol{J}_i^{\mathsf{T}} \cdot \boldsymbol{V}_i \cdot \boldsymbol{J}_i. \tag{8.3}$$

Minimizing the sum of the squared distances gives the fitted vertex $\hat{\boldsymbol{v}}$:

$$\hat{\boldsymbol{v}} = \arg_{\boldsymbol{v}} \min \mathcal{S}(\boldsymbol{v}), \quad \text{with} \quad \mathcal{S}(\boldsymbol{v}) = \sum_{i=1}^{n} \frac{D_i(\boldsymbol{v})}{\sigma_i^2}. \tag{8.4}$$

The minimization with the Newton–Raphson method proceeds iteratively:

1. Let $\boldsymbol{v}_0$ be an approximate initial vertex position. Compute $D_i(\boldsymbol{v}_0)$, $\boldsymbol{J}_i$ and σ_i^2 for $i = 1, \ldots, n$, according to Eqs. (8.1)–(8.3).
2. Compute the gradient of $\mathcal{S}$ with respect to $\boldsymbol{v}$ at $\boldsymbol{v}_0$:

$$\nabla \mathcal{S} = \frac{\partial \mathcal{S}}{\partial \boldsymbol{v}} = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \cdot \frac{\partial D_i}{\partial \boldsymbol{v}}, \quad \text{with} \quad \frac{\partial D_i}{\partial \boldsymbol{v}} = -\frac{\partial D_i}{\partial \boldsymbol{r}_i}. \tag{8.5}$$

3. Compute the Hessian matrix of $\mathcal{S}$ with respect to $\boldsymbol{v}$ at $\boldsymbol{v}_0$:

$$\nabla^2 \mathcal{S} = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \boldsymbol{H}_i, \quad \text{with} \quad \boldsymbol{H}_i = 2 \cdot \begin{pmatrix} a_{i,2}^2 + a_{i,3}^2 & -a_{i,1}\,a_{i,2} & -a_{i,1}\,a_{i,3} \\ -a_{i,1}\,a_{i,2} & a_{i,1}^2 + a_{i,3}^2 & -a_{i,2}\,a_{i,3} \\ -a_{i,1}\,a_{i,3} & -a_{i,2}\,a_{i,3} & a_{i,1}^2 + a_{i,2}^2 \end{pmatrix}. \tag{8.6}$$

4. Compute the solution $\boldsymbol{v}_1$ of $\nabla \mathcal{S} = \boldsymbol{0}$:

$$\boldsymbol{v}_1 = \boldsymbol{v}_0 - (\nabla^2 \mathcal{S})^{-1} \cdot \nabla \mathcal{S}. \tag{8.7}$$

5. Set $\boldsymbol{v}_0$ equal to $\boldsymbol{v}_1$ and repeat from step 2 until convergence.

The covariance matrix of the final estimate $\hat{\boldsymbol{v}}$ is given by $(\nabla^2 \mathcal{S})^{-1}$, and the χ^2-statistic of the fit is equal to $\mathcal{S}(\hat{\boldsymbol{v}})$. Its number of degrees of freedom is the sum of the ranks of all $\boldsymbol{V}_i$ minus three. Prior information on the vertex position that is independent of the track information can be included by an additional term in Eq. (8.4) or after the fit by a weighted mean.

8.1.1.2 Simplified Fit

If the uncertainty of the direction vectors $\boldsymbol{a}_i$ is neglected, the vertex fit can be further simplified [1]. Assume that track i is specified by a reference point $\boldsymbol{r}_i = (x_i, y_i, z_i)^\mathsf{T}$ in the vicinity of the vertex and a unit direction vector $\boldsymbol{a}_i$ in spherical coordinates:

$$\boldsymbol{a}_i = (\cos\varphi_i \cos\lambda_i, \sin\varphi_i \cos\lambda_i, \sin\lambda_i)^\mathsf{T}, \tag{8.8}$$

where φ_i is the azimuth and $\lambda_i = \pi/2 - \theta_i$ the dip angle, i.e., the complement of the polar angle. For the purpose of the vertex fit, a convenient choice of the coordinate system for position is a system where the $x_\parallel$-axis is parallel to the track, the $y_\perp$-axis is perpendicular to $x_\parallel$ and z, and the $z_\perp$-axis forms a right-handed orthonormal system with $x_\parallel$ and $y_\perp$. The coordinate transformation of the reference point $\boldsymbol{r}_i$ to this track-based system is given by the following rotation:

$$\boldsymbol{r}_i' = \boldsymbol{R}_i \boldsymbol{r}_i = \begin{pmatrix} \cos\varphi_i \cos\lambda_i & \sin\varphi_i \cos\lambda_i & \sin\lambda_i \\ -\sin\varphi_i & \cos\varphi_i & 0 \\ -\cos\varphi_i \sin\lambda_i & -\sin\varphi_i \sin\lambda_i & \cos\lambda_i \end{pmatrix} \boldsymbol{r}_i . \tag{8.9}$$

The coordinates $y_\perp$ and $z_\perp$ are called the transverse and the longitudinal impact parameter, respectively. The fit described in [1] assumes that $\boldsymbol{q}_i = (y_\perp, z_\perp)^\mathsf{T}$ has been estimated by the track fit, with the associated weight matrix $\boldsymbol{G}_i$, and that the direction errors are negligible. The transformation from $\boldsymbol{r}_i$ to $\boldsymbol{q}_i$ is given by the 2×3 matrix $\boldsymbol{T}_i$ consisting of the second and third line of $\boldsymbol{R}_i$.

The vertex $\boldsymbol{v}$ is estimated by minimizing the sum of the weighted distances between the reference points and the vertex, transformed to the corresponding track-based systems:

$$\mathcal{S}(\boldsymbol{v}) = \sum_{i=1}^{n} (\boldsymbol{r}_i - \boldsymbol{v})^\mathsf{T} \boldsymbol{T}_i^\mathsf{T} \boldsymbol{G}_i \boldsymbol{T}_i (\boldsymbol{r}_i - \boldsymbol{v}). \tag{8.10}$$

The estimated vertex and its covariance matrix $\boldsymbol{C}$ are therefore given by:

$$\hat{\boldsymbol{v}} = \boldsymbol{C} \sum_{i=1}^{n} \boldsymbol{W}_i \boldsymbol{r}_i, \quad \text{with} \quad \boldsymbol{C} = \left(\sum_{i=1}^{n} \boldsymbol{W}_i \right)^{-1} \quad \text{and} \quad \boldsymbol{W}_i = \boldsymbol{T}_i^\mathsf{T} \boldsymbol{G}_i \boldsymbol{T}_i, \ i = 1, \ldots, n. \tag{8.11}$$

8.1.2 Curved Tracks

The fits described in the preceding subsection can also be used with locally straight tracks for which the change of curvature and direction in the vicinity of the vertex is negligible. If this is not the case, nonlinearities in the track model have to be taken into account.

8.1.2.1 Nonlinear Regression

The general vertex fit can be formulated as a nonlinear regression model [2]. Assume that there are n tracks to be fitted to a common vertex. The tracks are specified by the estimated track parameters $\boldsymbol{q}_i$ and the associated covariance matrices $\boldsymbol{V}_i, i = 1, \ldots, n$. The parameters to be estimated are the vertex position $\boldsymbol{v}$ and the momentum vectors $\boldsymbol{p}_i$ of all tracks at the vertex, see Fig. 8.1.

The track parameters $\boldsymbol{q}_i$ are nonlinear functions of the parameters:

$$\boldsymbol{q}_i = \boldsymbol{h}_i(\boldsymbol{v}, \boldsymbol{p}_i), \; i = 1 \ldots, n. \tag{8.12}$$

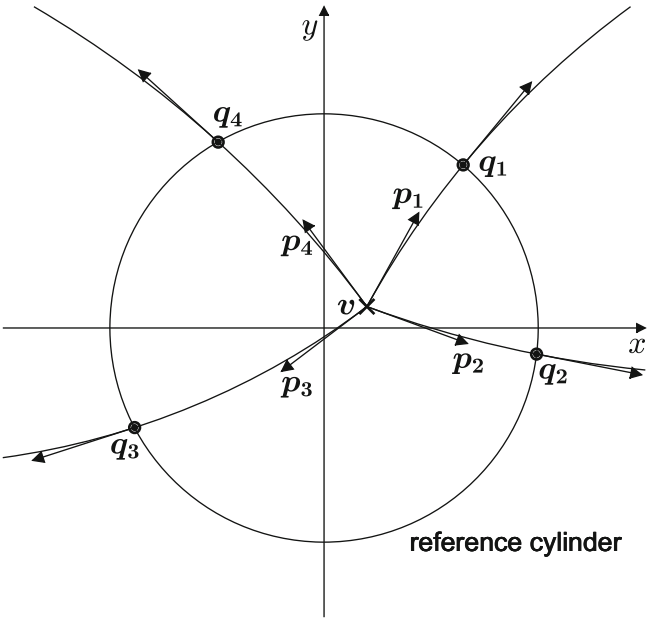

Fig. 8.1 A vertex fit with four tracks. The parameters of the fit are the vertex $\boldsymbol{v}$ and the momentum vectors $\boldsymbol{p}_i$; the observations are the estimated track parameters $\boldsymbol{q}_i$

The first-order Taylor expansion of $\boldsymbol{h}_i$ at a suitable expansion point $\boldsymbol{e}_0 = (\boldsymbol{v}_0, \boldsymbol{p}_{i,0})$ gives the following approximate linear model:

$$\boldsymbol{q}_i \approx \boldsymbol{A}_i \boldsymbol{v} + \boldsymbol{B}_i \boldsymbol{p}_i + \boldsymbol{c}_i, \; i = 1 \ldots, n, \tag{8.13}$$

with

$$\boldsymbol{A}_i = \left.\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{v}}\right|_{\boldsymbol{e}_0}, \quad \boldsymbol{B}_i = \left.\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{p}_i}\right|_{\boldsymbol{e}_0}, \quad \boldsymbol{c}_i = \boldsymbol{h}_i(\boldsymbol{v}_0, \boldsymbol{p}_{i,0}) - \boldsymbol{A}_i \boldsymbol{v}_0 - \boldsymbol{B}_i \boldsymbol{p}_{i,0}. \tag{8.14}$$

This can be written as:

$$\begin{pmatrix} \boldsymbol{q}_1 \\ \vdots \\ \boldsymbol{q}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_1 & \boldsymbol{O} & \ldots & \boldsymbol{O} \\ \boldsymbol{A}_2 & \boldsymbol{O} & \boldsymbol{B}_2 & \ldots & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_n & \boldsymbol{O} & \boldsymbol{O} & \ldots & \boldsymbol{B}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_n \end{pmatrix} + \begin{pmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_n \end{pmatrix}. \tag{8.15}$$

The LS estimates $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{p}}_i$ are obtained by:

$$\begin{pmatrix} \hat{\boldsymbol{v}} \\ \hat{\boldsymbol{p}}_1 \\ \vdots \\ \hat{\boldsymbol{p}}_n \end{pmatrix} = \boldsymbol{M}^{-1} \boldsymbol{N} \begin{pmatrix} \boldsymbol{q}_1 - \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{q}_n - \boldsymbol{c}_n \end{pmatrix}, \tag{8.16}$$

with

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{D}_0 & \boldsymbol{D}_1 & \boldsymbol{D}_2 & \ldots & \boldsymbol{D}_n \\ \boldsymbol{D}_1^\mathsf{T} & \boldsymbol{E}_1 & \boldsymbol{O} & \ldots & \boldsymbol{O} \\ \boldsymbol{D}_2^\mathsf{T} & \boldsymbol{O} & \boldsymbol{E}_2 & \ldots & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{D}_n^\mathsf{T} & \boldsymbol{O} & \boldsymbol{O} & \ldots & \boldsymbol{E}_n \end{pmatrix}, \tag{8.17}$$

$$\boldsymbol{N} = \begin{pmatrix} \boldsymbol{A}_1^\mathsf{T}\boldsymbol{G}_1 & \boldsymbol{A}_2^\mathsf{T}\boldsymbol{G}_2 & \ldots & \boldsymbol{A}_n^\mathsf{T}\boldsymbol{G}_n \\ \boldsymbol{B}_1^\mathsf{T}\boldsymbol{G}_1 & \boldsymbol{O} & \ldots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{B}_2^\mathsf{T}\boldsymbol{G}_2 & \ldots & \boldsymbol{O} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \ldots & \boldsymbol{B}_n^\mathsf{T}\boldsymbol{G}_n \end{pmatrix}, \tag{8.18}$$

$$\boldsymbol{D}_i = \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i, \quad \boldsymbol{E}_i = \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i = \boldsymbol{W}_i^{-1}, \quad \boldsymbol{G}_i = \boldsymbol{V}_i{}^{-1}, \quad i = 1, \ldots, n, \tag{8.19}$$

$$\boldsymbol{D}_0 = \sum_{i=1}^{n} \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{A}_i. \tag{8.20}$$

$\boldsymbol{C} = \boldsymbol{M}^{-1}$ can be written as a block matrix with the blocks $\boldsymbol{C}_{ij},\ i, j = 0, \ldots, n$:

$$\boldsymbol{C}_{00} = \left(\boldsymbol{D}_0 - \sum_{i=1}^{n} \boldsymbol{D}_i \boldsymbol{W}_i \boldsymbol{D}_i^{\mathsf{T}} \right)^{-1}, \tag{8.21}$$

$$\boldsymbol{C}_{0j} = -\boldsymbol{C}_{00} \boldsymbol{D}_j \boldsymbol{W}_j, \quad \boldsymbol{C}_{j0} = \boldsymbol{C}_{0j}^{\mathsf{T}}, \quad j > 0 \tag{8.22}$$

$$\boldsymbol{C}_{ij} = \delta_{ij} \boldsymbol{W}_i + \boldsymbol{W}_i \boldsymbol{D}_i^{\mathsf{T}} \boldsymbol{C}_{00} \boldsymbol{D}_j \boldsymbol{W}_j = \delta_{ij} \boldsymbol{W}_i - \boldsymbol{W}_i \boldsymbol{D}_i^{\mathsf{T}} \boldsymbol{C}_{0j}, \quad i, j > 0. \tag{8.23}$$

Substitution of Eqs. (8.21)–(8.23) into Eq. (8.16) gives the following expressions for the estimated parameters:

$$\hat{\boldsymbol{v}} = \boldsymbol{C}_{00} \sum_{j=1}^{n} \boldsymbol{A}_j^{\mathsf{T}} \boldsymbol{G}_j (\boldsymbol{I} - \boldsymbol{B}_j \boldsymbol{W}_j \boldsymbol{B}_j^{\mathsf{T}} \boldsymbol{G}_j)(\boldsymbol{q}_j - \boldsymbol{c}_j), \tag{8.24}$$

$$\hat{\boldsymbol{p}}_i = \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i (\boldsymbol{q}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \hat{\boldsymbol{v}}), \quad i = 1, \ldots, n. \tag{8.25}$$

The functions $\boldsymbol{h}_i$ are re-expanded at the new expansion point $\boldsymbol{e}_1 = (\hat{\boldsymbol{v}}, \hat{\boldsymbol{p}}_i), \quad i = 1, \ldots, n$, and the fit is iterated until convergence. After convergence, the track parameters $\boldsymbol{q}_i$ can be updated:

$$\hat{\boldsymbol{q}}_i = \boldsymbol{h}_i(\hat{\boldsymbol{v}}, \hat{\boldsymbol{p}}_i),\ i = 1 \ldots, n, \tag{8.26}$$

In the linear approximation, the joint covariance matrix of $\hat{\boldsymbol{v}}$ and all $\hat{\boldsymbol{p}}_i$ is equal to $\boldsymbol{C} = \boldsymbol{M}^{-1}$, from which the joint covariance matrix of all $\hat{\boldsymbol{q}}_i$ can be computed by linearized error propagation. The χ^2-statistic of the fit can be computed as follows:

$$\chi^2 = \sum_{i=1}^{n} (\boldsymbol{q}_i - \hat{\boldsymbol{q}}_i)^{\mathsf{T}} \boldsymbol{G}_i (\boldsymbol{q}_i - \hat{\boldsymbol{q}}_i). \tag{8.27}$$

If the errors of the estimated track parameters can be assumed to be approximately Gaussian, the chi-square statistic is approximately χ^2-distributed with

$$\text{ndf} = \sum_{i=1}^{n} \text{rank}(\boldsymbol{V}_i) - 3\,(n+1) \tag{8.28}$$

degrees of freedom.

8.1.2.2 Extended Kalman Filter

The nonlinear regression can be reformulated as an extended Kalman filter (see Sect. 6.1.2). Initially, the state vector consists only of the prior information about the vertex position $\boldsymbol{v}_0$, and its covariance matrix $\boldsymbol{C}_0$. In many instances, the prior information is given by the position and the size of the beam spot or the target. If no prior information is available, $\boldsymbol{v}_0$ is a rough guess, and $\boldsymbol{C}_0$ is set to a large diagonal matrix.

For each track i, $i = 1, \ldots, n$, the state vector is augmented by the three-momentum vector at the vertex $\boldsymbol{p}_i$. The system equation is the identity:

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1}, \quad \boldsymbol{C}_i = \boldsymbol{C}_{i-1}. \tag{8.29}$$

The measurement equation and its linearized form are the same as in Eqs. (8.12)–(8.14). The update of the vertex position and the estimation of $\boldsymbol{p}_i$ can now be written as:

$$\boldsymbol{v}_i = \boldsymbol{C}_i \left[\boldsymbol{C}_{i-1}^{-1} \boldsymbol{v}_{i-1} + \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} (\boldsymbol{q}_i - \boldsymbol{c}_i) \right], \tag{8.30}$$

$$\boldsymbol{p}_i = \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i (\boldsymbol{q}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \boldsymbol{v}_i), \tag{8.31}$$

with $\boldsymbol{W}_i = (\boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i)^{-1}$ and $\boldsymbol{G}_i^{\mathrm{B}} = \boldsymbol{G}_i - \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i$. The updated covariance and cross-covariance matrices are:

$$\mathsf{Var}\left[\boldsymbol{v}_i\right] = \boldsymbol{C}_i = \left(\boldsymbol{C}_{i-1}^{-1} + \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i \right)^{-1}, \tag{8.32}$$

$$\mathsf{Var}\left[\boldsymbol{p}_i\right] = \boldsymbol{W}_i + \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_i \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i, \tag{8.33}$$

$$\mathsf{Cov}\left[\boldsymbol{v}_i, \boldsymbol{p}_i\right] = -\boldsymbol{C}_i \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i. \tag{8.34}$$

Each update step gives rise to residuals $\boldsymbol{r}_i$ and a chi-square statistic χ_i^2:

$$\boldsymbol{r}_i = \boldsymbol{q}_i - \boldsymbol{h}_i(\boldsymbol{v}_i, \boldsymbol{p}_i), \tag{8.35}$$

$$\chi_i^2 = \boldsymbol{r}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{r}_i + (\boldsymbol{v}_i - \boldsymbol{v}_{i-1})^{\mathsf{T}} \boldsymbol{C}_{i-1}^{-1} (\boldsymbol{v}_i - \boldsymbol{v}_{i-1}). \tag{8.36}$$

The chi-square statistic has two degrees of freedom and can be used to test the compatibility of track i with the current fitted vertex. If no intermediate results are needed, the computation of the momentum vectors $\boldsymbol{p}_i$ can be deferred to the smoother, and the final vertex $\boldsymbol{v}_n$ and its covariance matrix $\boldsymbol{C}_n$ can be computed directly, cf. Eqs. (8.21) and (8.24):

$$\boldsymbol{v}_n = \boldsymbol{C}_n \left[\boldsymbol{C}_0^{-1} \boldsymbol{v}_0 + \sum_{i=1}^{n} \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} (\boldsymbol{q}_i - \boldsymbol{c}_i) \right], \tag{8.37}$$

$$\boldsymbol{C}_0 = \left(\boldsymbol{C}_0^{-1} + \sum_{i=1}^{n} \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i \right)^{-1} . \tag{8.38}$$

As there is no process noise in the system equation, the smoother is tantamount to recomputing the momentum vectors and the covariance matrices with the final vertex $\boldsymbol{v}_n$ and its covariance matrix $\boldsymbol{C}_n$, see also Eqs. (8.31)–(8.34):

$$\boldsymbol{p}_{i|n} = \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i (\boldsymbol{q}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \boldsymbol{v}_n), \tag{8.39}$$

$$\mathsf{Var}\left[\boldsymbol{p}_{i|n}\right] = \boldsymbol{W}_i + \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_n \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i , \tag{8.40}$$

$$\mathsf{Cov}\left[\boldsymbol{v}_n, \boldsymbol{p}_{i|n}\right] = -\boldsymbol{C}_n \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i , \tag{8.41}$$

$$\mathsf{Cov}\left[\boldsymbol{p}_{i|n}, \boldsymbol{p}_{j|n}\right] = \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_n \boldsymbol{A}_j^{\mathsf{T}} \boldsymbol{G}_j \boldsymbol{B}_j \boldsymbol{W}_j . \tag{8.42}$$

The update of the track parameters reads:

$$\hat{\boldsymbol{q}}_i = \boldsymbol{h}_i(\boldsymbol{v}_n, \boldsymbol{p}_{i|n}), \ i = 1 \ldots, n. \tag{8.43}$$

Their joint covariance matrix can be computed by linearized error propagation. Each track can be tested against the final vertex by computing the smoothed residuals and the corresponding chi-square statistic:

$$\boldsymbol{r}_{i|n} = \boldsymbol{q}_i - \boldsymbol{h}_i(\boldsymbol{v}_n, \boldsymbol{p}_{i|n}), \tag{8.44}$$

$$\chi^2_{i|n} = \boldsymbol{r}_{i|n}^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{r}_{i|n} + (\boldsymbol{v}_n - \boldsymbol{v}_{n|-i})^{\mathsf{T}} \boldsymbol{C}_{n|-i}^{-1} (\boldsymbol{v}_n - \boldsymbol{v}_{n|-i}), \tag{8.45}$$

where $\boldsymbol{v}_{n|-i}$ is the final vertex with track i removed, and $\boldsymbol{C}_{n|-i}$ is its covariance matrix:

$$\boldsymbol{v}_{n|-i} = \boldsymbol{C}_{n|-i} \left[\boldsymbol{C}_n^{-1} \boldsymbol{v}_n - \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} (\boldsymbol{q}_i - \boldsymbol{c}_i) \right], \tag{8.46}$$

$$\mathsf{Var}\left[\boldsymbol{v}_{n|-i}\right] = \boldsymbol{C}_{n|-i} = \left(\boldsymbol{C}_n^{-1} - \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i \right)^{-1} . \tag{8.47}$$

Searching for outliers in this way is, however, tedious and time consuming. The adaptive vertex fit described in Sect. 8.2 is better suited to and more powerful for this task, especially if there are several outliers.

8.1.2.3 Fit with Perigee Parameters

In many collider experiments, past and present, the magnetic field in the vicinity of the collision region is almost perfectly homogeneous, giving a helical track model. The "perigee" parametrization for such helical tracks was introduced in [3], with a

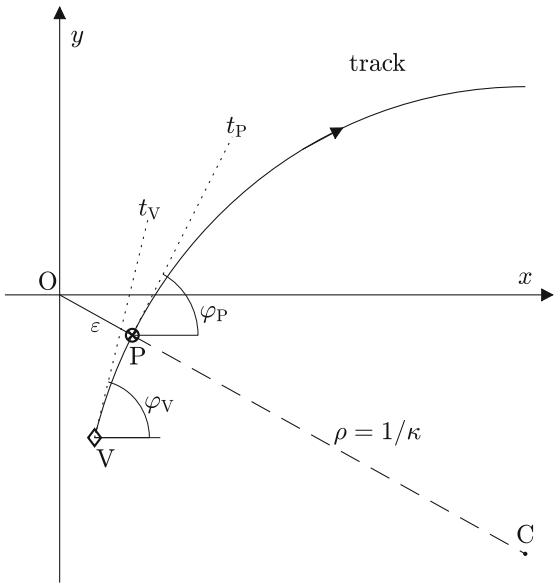

Fig. 8.2 A helical track in the projection to the (x, y)-plane. O: origin; C:circle center; P: perigee point; V: vertex; ρ: circle radius; t_P: tangent at P; φ_P: azimuth of track direction at P; t_V: tangent at V; φ_V: azimuth of track direction at V

correction in [4]. The track is parametrized around the point of closest approach, the perigee point $\boldsymbol{v}^P$, of the helix to the z-axis, which is also the direction of the beams and the magnetic field. The perigee point is expected to be close to the vertex. The five track parameters of the perigee parametrization are (see Fig. 8.2):

1. The impact parameter ϵ. By convention, the sign of ϵ is positive if the origin O is at the left side of the trajectory.
2. The azimuth φ_P of the tangent to the track at P.
3. The z-coordinate z_P of the perigee point P.
4. The polar angle ϑ of the helix with respect to the z-axis.
5. The signed curvature κ. By convention, the sign of κ is positive if the trajectory is anti-clockwise.

With these definitions, the trajectory can be approximately parametrized in terms of a running parameter s, which is the distance from P along the projected helix:

$$x \approx \epsilon \sin\varphi_P + s\cos\varphi_P - \frac{s^2\kappa}{2}\sin\varphi_P, \tag{8.48}$$

$$y \approx -\epsilon\cos\varphi_P + s\sin\varphi_P + \frac{s^2\kappa}{2}\cos\varphi_P, \tag{8.49}$$

$$z \approx z_P + s\cot\vartheta.$$

The track parameters $\boldsymbol{q} = (\epsilon, \varphi_P, \vartheta, z_P, \kappa)^T$ have to be expressed as a function of the coordinates $\boldsymbol{v} = (x_V, y_V, z_V)^T$ of the vertex V, and the track parameters $\boldsymbol{p} = (\vartheta, \varphi_V, \kappa)^T$ at V. Note that ϑ and κ are invariant along the helix. As higher

orders of κ can usually be neglected for tracks in collider experiments, the following functional dependence is obtained:

$$\begin{aligned} \epsilon &\approx -R - Q^2\kappa/2, \\ z_\mathrm{P} &\approx z_\mathrm{V} - Q(1 - R\kappa)\cot\vartheta, \\ \varphi_\mathrm{P} &\approx \varphi_\mathrm{V} - Q\kappa, \end{aligned} \tag{8.50}$$

with

$$Q = x_\mathrm{V}\cos\varphi_\mathrm{V} + y_\mathrm{V}\sin\varphi_\mathrm{V}, \quad R = y_\mathrm{V}\cos\varphi_\mathrm{V} - x_\mathrm{V}\sin\varphi_\mathrm{V}. \tag{8.51}$$

The Jacobian matrix at the lowest order is given by:

$$\frac{\partial \boldsymbol{q}}{\partial(\boldsymbol{v}, \boldsymbol{p})} = \begin{pmatrix} s & -tc & -\kappa c \\ -c & -ts & -\kappa s \\ 0 & 1 & 0 \\ 0 & Q(1+t^2) & 0 \\ Q & -Rt & 1 \\ -Q^2/2 & QRt & -Q \end{pmatrix}, \tag{8.52}$$

with

$$c = \cos\varphi_\mathrm{V}, \quad s = \sin\varphi_\mathrm{V}, \quad t = \cot\vartheta. \tag{8.53}$$

With these ingredients, the nonlinear regression (see Sect. 8.1.2.1) can be computed.

8.2 Robust and Adaptive Vertex Fitting

8.2.1 Vertex Fit with M-Estimator

Estimators are called robust if they are insensitive to outlying observations [5–7]. The M-estimator, see Sect. 6.2.1, is a well-known robust estimator that can be implemented as an iterated reweighted LS estimator that assigns smaller weights (larger uncertainties) to observations suspected to be outliers. One of the first attempts, if not the first, to make the vertex fit robust is the extension of the Kalman filter to an M-estimator [8] of the Huber type [5]. Before the fit, the parameters of each track are decorrelated by finding the orthogonal matrix $\boldsymbol{U}_i$ that transforms the covariance matrix $\boldsymbol{V}_i$ to a diagonal matrix $\boldsymbol{D}_i$:

$$\boldsymbol{D}_i = \boldsymbol{U}_i \boldsymbol{V}_i \boldsymbol{U}_i^\mathsf{T}. \tag{8.54}$$

Table 8.1 Algorithm: Vertex fit with M-estimator

VERTEX FIT WITH M-ESTIMATOR

1. Set $\varepsilon = 10^{-6}$, set the iteration counter $I = 0$ and set $\boldsymbol{V}_i{}^{(0)} = \boldsymbol{V}_i$, $i = 1, \ldots, n$. Choose a robustness constant c, usually a value between 1 and 3.
2. Compute the LS estimate $\boldsymbol{v}^{(I)}$ by one of the estimators in Sect. 8.1, using the covariance matrices $\boldsymbol{V}_i{}^{(I)}$. For each track i, compute the smoothed residuals $\boldsymbol{r}_{i|n}$.
3. For each component $r_{i|n,j}$, $j = 1, \ldots, 5$, compute a weight $w_{i,j}$ according to:

$$w_{i,j} = \max\left[\varepsilon, \frac{\psi\left(r_{i|n,j}/\boldsymbol{V}_{i,jj}\right)}{r_{i|n,j}/\boldsymbol{V}_{i,jj}}\right], \tag{8.56}$$

   where $\boldsymbol{V}_{i,jj}$ is the j-th diagonal element of the initial covariance matrix $\boldsymbol{V}_i$, and $\psi(t)$ is one of the functions in Table 6.1 or a similar one from the literature.
4. Increase the iteration counter I by 1. For each track i and all j, set $\boldsymbol{V}_{i,jj}^{(I)} = \boldsymbol{V}_{i,jj}/w_{i,j}$.
5. Check convergence; if necessary, go to step 2.

The measurement equation Eq. (8.13) is transformed by setting

$$\boldsymbol{V}_i \leftarrow \boldsymbol{D}_i, \quad \boldsymbol{q}_i \leftarrow \boldsymbol{U}_i \boldsymbol{q}_i, \quad \boldsymbol{A}_i \leftarrow \boldsymbol{U}_i \boldsymbol{A}_i, \quad \boldsymbol{B}_i \leftarrow \boldsymbol{U}_i \boldsymbol{B}_i, \quad \boldsymbol{c}_i \leftarrow \boldsymbol{U}_i \boldsymbol{c}_i. \tag{8.55}$$

The M-estimator is then computed by an iterated LS estimator, see Table 8.1. Vertex fits using M-estimators with other weight functions are described in [9, 10].

8.2.2 *Adaptive Vertex Fit with Annealing*

The adaptive vertex fit (AVF) was introduced in [11] and further investigated in [12–15]. It can be interpreted either as an EM algorithm or as an M-estimator with a specific weight function [16, 17]; see also Table 6.1. In the AVF, tracks as a whole are down-weighted, as the weight w_i is a function of the distance of track i from the current vertex, measured by the chi-square χ_i^2:

$$\chi_i^2 = \boldsymbol{r}_{i|n}^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{r}_{i|n}, \quad w_i\left(\chi_i^2\right) = \frac{\exp\left(-\chi_i^2/2T\right)}{\exp\left(-\chi_i^2/2T\right) + \exp\left(-\chi_c^2/2T\right)}, \tag{8.57}$$

where T is a temperature parameter. The weight w_i can be interpreted as the probability that track i belongs to the vertex. The chi-square cut χ_c^2 sets the threshold where the weight is equal to 0.5. Beyond this cut, a track is considered to be an outlier rather than an inlier.

The temperature T modifies the shape of the function in Eq. (8.57). At high temperature, the weight function varies very slowly; at low temperature, the weight is close to 1 for $\chi_i^2 \leq \chi_c^2$ and close to 0 for $\chi_i^2 > \chi_c^2$ (see Fig. 6.1c), with a sharp

Table 8.2 Algorithm: Adaptive vertex fit with annealing

ADAPTIVE VERTEX FIT WITH ANNEALING

1. Set the iteration counter $I = 0$ and set ${\boldsymbol{V}_i}^{(0)} = \boldsymbol{V}_i,\ i = 1, \ldots, n$. Choose a threshold χ^2_c, an initial temperature T_0, and an initial vertex estimate $\boldsymbol{v}^{(0)}$.
2. For each track i, compute the smoothed residuals $\boldsymbol{r}_{i|n}$ and the χ^2-statistic χ^2_i in Eq. (8.57), using $\boldsymbol{v}^{(I)}$ and $\boldsymbol{V}_i$. Compute the weight w_i according to:

$$w_i = \frac{\exp\left(-\chi^2_i/2T_k\right)}{\exp\left(-\chi^2_i/2T_k\right) + \exp\left(-\chi^2_c/2T_k\right)}. \tag{8.58}$$

3. Increase the iteration counter I by 1. For each track i, set ${\boldsymbol{V}_i}^{(I)} = \boldsymbol{V}_i/w_i$.
4. Compute the LS estimate $\boldsymbol{v}^{(I)}$ by one of the estimators in Sect. 8.1, using the inflated covariance matrices ${\boldsymbol{V}_i}^{(I)}$.
5. Set the new temperature T_I according to a chosen annealing schedule and go to step 2.
6. When the final temperature is reached, iterate until convergence.

drop at $\chi^2_i = \chi^2_c$. The weights at low temperature can be used to identify secondary tracks in a fit of the primary vertex; see Sect. 7.3.3.

Similar to the M-estimator in Sect. 8.2.1, the adaptive vertex fit is implemented as an iterated LS estimator, which can be one of the methods described in Sect. 8.1. The temperature T can be used to employ an annealing procedure that helps to reach the globally optimal solution. The adaptive vertex fit is summarized in Table 8.2.

In order to get reasonable initial weights, the initial vertex $\boldsymbol{v}^{(0)}$ has to be chosen carefully, preferably with a robust finder [18]. The annealing schedule and the constant χ^2_c have to be tuned on simulated data. A detailed study and useful hints can be found in [15].

8.2.3 Vertex Quality

The assessment of the quality of a fitted vertex is similar to the assessment of track quality; see Sect. 6.4. The primary criterion is the chi-square statistic of the vertex fit, or its p-value. If the p-value is unreasonably small, a search for outlying tracks can be started. There are several sources of outliers in the vertex fit:

- Fake tracks and tracks that include unrecognized extraneous measurements or noise hits.
- Tracks with an unrecognized interaction (kink) in the material.
- Tracks with a covariance matrix that does not properly reflect the statistical uncertainty of the track parameters, for instance, because of an incorrect evaluation of material effects.
- Tracks that belong to a different vertex, in particular to a nearby secondary vertex.

Just as in the case of the track fit, outliers can be detected by a test based on the residuals of the track with respect to the estimated vertex position. Outlying tracks that pass the track quality check are often candidates for inclusion in a secondary vertex; see Sects. 7.3.3 and 9.2.

8.3 Kinematic Fit

The vertex fit as described so far imposes purely geometrical constraints on the participating tracks, namely that they originate at the same point in space. Especially in the case of a secondary vertex, the vertex fit can be extended by imposing other geometrical or kinematical constraints. In the vertex fit of a photon conversion (see Sect. 9.4), the constraint can be imposed that the momentum vectors of the outgoing tracks are parallel. In the fit of a decay vertex (see Sects. 9.2 and 9.3), the laws of momentum and/or energy conservation can be imposed as constraints. In addition, assumptions about the mass of some or all of the participating particles can be included. The width of a resonance can be included by considering its mass as a virtual measurement with a standard error reflecting the width. Such fits are called kinematic fits.

In a kinematic fit, a track is most conveniently represented by a collection $\boldsymbol{u}$ of parameters that are physically meaningful and allow a simple formulation of the constraints [19]. If only kinematic constraints are present, $\boldsymbol{u}$ contains the energy and the Cartesian momentum components in the global coordinate system:

$$\boldsymbol{u} = (E, p_x, p_y, p_z)^{\mathsf{T}}. \tag{8.59}$$

For charged tracks, $\boldsymbol{u}$ is computed from the usual representation by a five-vector $\boldsymbol{q}$, and the covariance matrix $\boldsymbol{V} = \mathsf{Var}[\boldsymbol{u}]$ is obtained by linearized error propagation. If the mass of the particle is assumed to be known and fixed, the rank of $\boldsymbol{V}$ is three, as the energy is a deterministic function of the momentum p and the mass m. If there is an independent measurement of the energy, for instance by a cluster in the calorimeter, the rank is four. For neutral tracks, $\boldsymbol{u}$ is computed from the calorimeter information. As there is no independent momentum measurement, the rank of $\boldsymbol{V}$ is three.

If in addition a vertex constraint is to be enforced, the location x, y, z in space, which can vary along the track, is appended to $\boldsymbol{u}$, and the rank of $\boldsymbol{V}$ increases by two [19].

If n tracks with parameters $\boldsymbol{u}_k$ and covariance matrices $\boldsymbol{V}_k$, $k = 1, \ldots, n$, participate in the kinematic fit, their parameters are combined in a single column vector $\boldsymbol{y}$:

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_n \end{pmatrix}. \tag{8.60}$$

If the estimated $\boldsymbol{u}_k$ are uncorrelated, their joint covariance matrix $\boldsymbol{V}$ is block-diagonal:

$$\mathsf{Var}\,[\boldsymbol{y}] = \boldsymbol{V} = \mathrm{blkdiag}(\boldsymbol{V}_1, \ldots, \boldsymbol{V}_n). \tag{8.61}$$

If the momentum parts of the $\boldsymbol{u}_k$ are the results of a preceding vertex fit, they are correlated, and their joint covariance matrix is given in Eqs. (8.40) and (8.42).

The vector $\boldsymbol{y}$ is considered to be an unbiased observation of the vector $\boldsymbol{\alpha}$ that satisfies $r \leq \mathrm{rank}(\boldsymbol{V})$ kinematical or geometrical constraints:

$$\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{\varepsilon}, \quad \mathsf{E}\,[\boldsymbol{\varepsilon}] = \boldsymbol{0}, \quad \mathsf{Var}\,[\boldsymbol{\varepsilon}] = \boldsymbol{V}. \tag{8.62}$$

The constraints are expressed by r equations $h_i(\boldsymbol{\alpha}) = 0,\ i = 1, \ldots, r$, which can be written compactly in the following vector form:

$$\boldsymbol{h}(\boldsymbol{\alpha}) = \boldsymbol{0}, \quad \text{with } \boldsymbol{h} = (h_1(\boldsymbol{\alpha}), \ldots, h_r(\boldsymbol{\alpha}))^{\mathsf{T}}. \tag{8.63}$$

The function $\boldsymbol{h}$ is approximated by its first-order Taylor expansion at the expansion point $\boldsymbol{\alpha}_0 = \boldsymbol{y}$, resulting in a set of linear constraints:

$$\boldsymbol{h}(\boldsymbol{\alpha}) \approx \boldsymbol{h}(\boldsymbol{\alpha}_0) + \boldsymbol{H}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) = \boldsymbol{h}(\boldsymbol{\alpha}_0) + \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{H}\boldsymbol{\alpha}_0 = \boldsymbol{H}\boldsymbol{\alpha} + \boldsymbol{d} = \boldsymbol{0}, \tag{8.64}$$

with the Jacobian matrix $\boldsymbol{H} = \partial \boldsymbol{h}/\partial \boldsymbol{\alpha}$ evaluated at $\boldsymbol{\alpha} = \boldsymbol{\alpha}_0$ and $\boldsymbol{d} = \boldsymbol{h}(\boldsymbol{\alpha}_0) - \boldsymbol{H}\boldsymbol{\alpha}_0$. It is assumed that $\boldsymbol{H}$ has rank r in a sufficiently large neighbourhood of $\boldsymbol{y}$.

If $\boldsymbol{V}$ has full rank, the constrained LS estimate of $\boldsymbol{\alpha}$ is obtained by minimizing the following objective function:

$$\mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = (\boldsymbol{\alpha} - \boldsymbol{y})^{\mathsf{T}}\, \boldsymbol{G}\, (\boldsymbol{\alpha} - \boldsymbol{y}) + 2\, \boldsymbol{\lambda}^{\mathsf{T}}\, (\boldsymbol{H}\boldsymbol{\alpha} + \boldsymbol{d})\,, \tag{8.65}$$

where $\boldsymbol{G} = \boldsymbol{V}^{-1}$ and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Setting the gradient of $\mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ to zero results in the following system of linear equations:

$$\boldsymbol{G}\, (\boldsymbol{\alpha} - \boldsymbol{y}) + \boldsymbol{H}^{\mathsf{T}} \boldsymbol{\lambda} = \boldsymbol{0}, \tag{8.66}$$

$$\boldsymbol{H}\boldsymbol{\alpha} + \boldsymbol{d} = \boldsymbol{0}. \tag{8.67}$$

The system can be explicitly solved for $\boldsymbol{\alpha}$, the solution being the estimate $\hat{\boldsymbol{\alpha}}$:

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{y} - \boldsymbol{V}\, \boldsymbol{H}^{\mathsf{T}} \boldsymbol{G}_H\, (\boldsymbol{H}\boldsymbol{y} + \boldsymbol{d})\,, \quad \text{with } \boldsymbol{G}_H = \left(\boldsymbol{H}\, \boldsymbol{V}\, \boldsymbol{H}^{\mathsf{T}}\right)^{-1}. \tag{8.68}$$

The solution is valid also for singular $\boldsymbol{V}$ as long as $r \leq \mathrm{rank}(\boldsymbol{V})$, so that $\boldsymbol{H}\, \boldsymbol{V}\, \boldsymbol{H}^{\mathsf{T}}$ has full rank r and can be inverted. This can be proved by regularization of $\boldsymbol{V}$; see Appendix B.

The constraints are re-expanded at the new expansion point $\boldsymbol{\alpha}_0 = \hat{\boldsymbol{\alpha}}$, and the fit is iterated until convergence. The covariance matrix $\boldsymbol{V}_{\hat{\boldsymbol{\alpha}}}$ of the final estimate $\hat{\boldsymbol{\alpha}}$ and the chi-square statistic of the fit are given by:

$$\boldsymbol{V}_{\hat{\boldsymbol{\alpha}}} = \boldsymbol{V} - \boldsymbol{V}\boldsymbol{H}^{\mathsf{T}}\boldsymbol{G}_H\boldsymbol{H}\boldsymbol{V}, \tag{8.69}$$

$$\chi^2 = (\boldsymbol{H}\boldsymbol{y} + \boldsymbol{d})^{\mathsf{T}}\boldsymbol{G}_H(\boldsymbol{H}\boldsymbol{y} + \boldsymbol{d}). \tag{8.70}$$

The χ^2 has r degrees of freedom and can be used to assess the goodness of the fit, provided that the linear approximation of the constraints is satisfactory and the distribution of the error term $\boldsymbol{\varepsilon}$ in Eq. (8.62) is close to normal.

A comprehensive software package for kinematic fitting, called KWFIT, can be found in [20]. In addition to stand-alone kinematic fits as described above, it allows fitting entire decay chains with multiple vertices. For a discussion of its features and of kinematic fitting in general, see [21]. More recent decay chain fitters are described in [22] and [23].

References

1. V. Karimäki, Effective Vertex Fitting. Technical Report CMS-NOTE-1997-051, CERN, Geneva (1997). http://cdsweb.cern.ch/record/687531
2. P. Billoir, R. Frühwirth, M. Regler, Nucl. Instrum. Meth. Phys. Res. A **241**, 115 (1985)
3. P. Billoir, S. Qian, Nucl. Instrum. Meth. Phys. Res. A **311**, 139 (1992)
4. P. Billoir, S. Qian, Nucl. Instrum. Meth. Phys. Res. A **350**, 624 (1994)
5. P.J. Huber, E.M. Ronchetti, *Robust statistics*, 2nd edn. (Wiley, Hoboken, 2009)
6. F.R. Hampel, et al., *Robust Statistics: The Approach Based on Influence Functions* (Wiley, Hoboken, 2011)
7. P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection* (Wiley, Hoboken, 1987)
8. R. Frühwirth, et al., Comput. Phys. Commun. **96**, 189 (1996)
9. G. Agakichiev, et al., Nucl. Instrum. Meth. Phys. Res. A **394**, 225 (1997)
10. M. Kucharczyk, P. Morawski, M. Witek, Primary Vertex Reconstruction at LHCb. Technical Report CERN-LHCb-PUB-2014-044, CERN, Geneva (2014). https://cds.cern.ch/record/1756296
11. W. Waltenberger, Development of vertex finding and vertex fitting algorithms for CMS. Ph.D. thesis, TU Wien (2004). http://repositum.tuwien.ac.at/obvutwhs/download/pdf/1562338
12. J. D'Hondt, et al., IEEE Trans. Nucl. Sci. **51**(5), 2037 (2004)
13. E. Chabanat, et al., Nucl. Instrum. Meth. Phys. Res. A **549**, 188 (2005)
14. T. Speer, R. Frühwirth, P. Vanlaer, W. Waltenberger, Nucl. Instrum. Meth. Phys. Res. A **566**, 149 (2006)
15. W. Waltenberger, R. Frühwirth, P. Vanlaer, J. Phys. G **34**, N343 (2007)
16. R. Frühwirth, W. Waltenberger, Austrian J. Stat. **37**(3&4), 301 (2008). https://tinyurl.com/RedescendingMestimators
17. A. Strandlie, R. Frühwirth, Rev. Mod. Phys. **82**, 1419 (2010)
18. R. Frühwirth, et al., in *Proceedings of the 13th International Conference on Computing in High-Energy and Nuclear Physics (CHEP 2003)*, vol. eConf C0303241 (2003), p. TULT013. https://www.slac.stanford.edu/econf/C0303241/proc/papers/TULT013.PDF
19. P. Avery, Applied Fitting Theory VI — Formulas for Kinematic Fitting. Technical Report CBX 98-37, CLEO Collaboration (1998). https://tinyurl.com/Avery-KinematicFit

20. P. Avery, KWFIT. https://tinyurl.com/Avery-KWFIT
21. P. Avery, in *Proceedings of the International Conference on Computing in High Energy and Nuclear Physics* (2000). http://chep2000.pd.infn.it/paper/pap-a207.pdf
22. W.D. Hulsbergen, Nucl. Instrum. Meth. Phys. Res. A **552**(3), 566 (2005)
23. S.Gorbunov, I. Kisel, Reconstruction of decayed particles based on the Kalman filter. Technical Report CBM-SOFT-note-2007-003, STAR experiment, Brookhaven National Laboratory (2007). https://tinyurl.com/Gorbunov-KFParticle

Chapter 9
Secondary Vertex Reconstruction

Abstract The chapter reviews methods for the search for secondary vertices. Four types of secondary vertices are discussed in detail: decays of short-lived particles, decays of long-lived particles, photon conversions, and hadronic interactions in the detector material.

9.1 Introduction

A vertex is called a secondary vertex or displaced vertex if it is outside the beam profile in a collider experiment or outside the target region in a fixed-target experiment. Secondary vertices arise in the following contexts:

Decays of unstable particles The decaying particle is call the "mother" particle, the decay products are called the "daughter" particles. Depending on the lifetime and the momentum of the mother particle, the secondary decay vertex is displaced by a certain distance from its production point, which can be either a primary or itself a secondary vertex. The properties of the mother particles have to be inferred from the decay products. Finding decays is important for many types of physics analyses as well as for momentum scale calibration, by comparing the reconstructed mass of the mother particle to the known one.

Interactions in the detector material Finding interaction vertices can be useful for mapping the amount and position of material in the detector. Two types of interactions are important in practice: photon conversions, i.e., pair production of electrons and positrons or muons in the electric field of a nucleus; and hadronic interactions of primary particles. A decay of a charged particles into a single charged daughter particle plus some neutral daughters looks just like a kink in the track. Kink finding is the subject of Sect. 6.4.3. The following subsections present some methods for finding decay and interaction vertices.

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_9

9.2 Decays of Short-Lived Particles

In the present context, short-lived particles are defined as particles that decay before they enter the first layer of the innermost tracking device. This includes B (beauty) and D (charmed) hadrons as well as τ leptons, which travel no more than a few millimeters before they decay.

The key to successful secondary vertex reconstruction is a precise knowledge of the primary vertex and the correct selection of the tracks of the decay products. In order to make sure that the primary vertex is not contaminated with secondary tracks, fake tracks or—at the LHC—pile-up tracks, the fit has to be made robust, either by removing outliers (Sect. 8.2.3), or by employing a robust fitting algorithm in the first place (Sects. 8.2.1 and 8.2.2). If there is prior information on the beam spot, it should be included into the primary vertex fit. The tracks removed from the primary vertex or down-weighted as outliers are by definition candidates for secondary tracks; see Sect. 7.3.3.

The selection of the secondary tracks depends on the physics goals of the analysis. If a specific decay channel is considered, the tracks can be selected according to kinematic criteria, for instance their type, charge, and momentum [1]. In addition, the distance of the track from the primary vertex in the transverse plane, called the transverse impact parameter d^0, can be used to verify that the selected tracks are not produced at the primary vertex. As the uncertainty of d^0 depends on the angle and the momentum of the track, the test is based on the standardized impact parameter $s^0 = d^0/\sigma[d^0]$, also called the significance of the impact parameter.

Secondary tracks can also be selected by their impact parameter alone. To this end, the impact parameter of a track is given a sign that is positive if the track intersects the trajectory of the decaying hadron or τ lepton downstream of the primary vertex [1–3]. As the trajectory is not known before the secondary vertex fit, it is approximated by the jet axis of the jet the decaying particle is embedded in. Only tracks with positive sign are candidates for secondary tracks.

The information contained in the impact parameter of a secondary track can be augmented by considering its functional relation with the azimuth angle of the track in the transverse projection, see Fig. 9.1. Let $\phi_1, \ldots, \phi_m$ be the azimuth angles and $d_1^0, \ldots, d_m^0$ the transverse impact parameters of m tracks emerging from the same secondary vertex. If the primary vertex is very close to the origin of the coordinate system, as it usually is, the following functional relation holds approximately:

$$d_i^0 \approx \ell \sin(\phi_i - \phi_\mathrm{h}) \approx \ell(\phi_i - \phi_\mathrm{h}), \quad i = 1, \ldots, m, \tag{9.1}$$

where ℓ is the decay length of the decaying particle and ϕ_h its azimuth angle. Thus in the (ϕ, d^0)-plane the points corresponding to secondary tracks lie on a straight line. This line can be found in various ways, for instance, by a Hough transform (Sect. 5.1.2), by histogramming the direction angles of segments connecting two points or by agglomerative clustering of the segments [3].

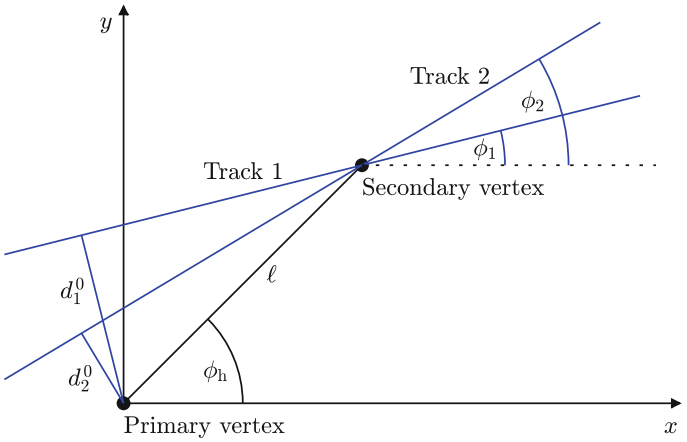

Fig. 9.1 The functional relation between ϕ and d^0 of secondary tracks

9.3 Decays of Long-Lived Particles

Long-lived particles such as neutral K-short mesons K_S^0 and Λ baryons decay in the tracker volume. Reconstruction of their decays serves as a powerful cross-check of the magnetic field map and the alignment, as their mass is very well known, and systematic errors in the field map as well as misalignment can lead to a bias in the momentum and invariant mass estimation. Candidates for the decay products can be identified by a large impact parameter and the fact that tracker hits are consistently missing up to a certain layer. In the search for 2-prong neutral decays, so-called V0s, pairs of tracks with opposite charge and small distance in space are combined to vertex candidates. Without particle identification, the Armenteros–Podolansky plot is a useful tool to decide between hypotheses about the mother particle from the kinematics of their decay products [4]. It is a scatter plot of the longitudinal momentum asymmetry versus the transverse momentum of the positive decay product, see Fig. 9.2. The figure shows that in some regions of the plot a unique decision may not be possible.

The resolution of eventual ambiguities requires a kinematic fit with mass constraints (see Sect. 8.3) based on particle identification of the decay products. For example, in the case of K_S^0 versus $\Lambda/\bar{\Lambda}$, charged pions have to be separated from protons.

If the decay vertices are allowed to have more than two charged outgoing particles, two-track vertices can serve as seed vertices [5]. A seed vertex is rejected if at least one of its tracks has hits between the seed vertex and the primary vertex. The seed vertices are then clustered to larger vertices by an agglomerative procedure.

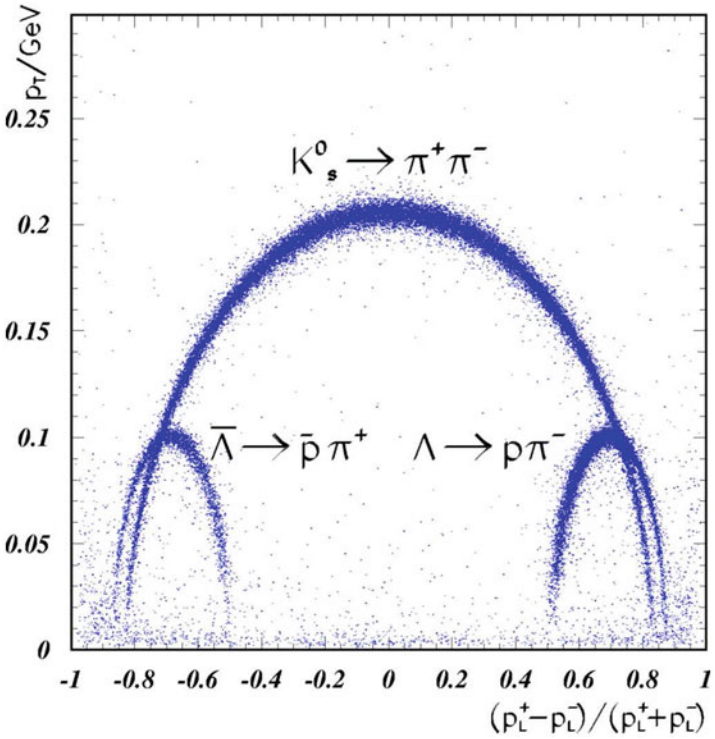

Fig. 9.2 Armenteros–Podolansky plot for K^0_S and $\Lambda/\bar{\Lambda}$. (From [4], with permission by the author)

9.4 Photon Conversions

The reconstruction of photon conversions is an integral part of photon reconstruction in general. While the energy of unconverted photons is measured directly in the electromagnetic calorimeter (ECAL), a converted photon is reconstructed from the charged lepton pair, mostly an e^+e^- pair, that is produced in the electric field of a nucleus. A characteristic feature of the conversion vertex is the fact that the two leptons are parallel to each other. As a consequence, the vertex position along the photon direction is not very well defined by the vertex fit alone. Two topical examples of the reconstruction of converted photons are presented in the following paragraphs.

In the ATLAS experiment (Sect. 1.6.1.2), the first step is the standard electron/positron reconstruction, briefly described in Sect. 10.2. Clusters in the ECAL are built and used to create regions of interest (ROIs). Inside an ROI track finding is modified such that up to 30% energy loss is allowed at each material layer. After loosely matching the tracks in the ROIs with the ECAL clusters, tracks with silicon hits are refitted with the Gaussian-sum filter (GSF, Sect. 6.2.3). Conversion finding

is run on the loosely matched tracks that have a high probability to be electrons or positrons according to the transition radiation detector. A conversion may have one or two outgoing tracks. Double-track conversions are created when two tracks form a vertex that is consistent with the hypothesis that they have been produced from a massless particle. Single-track conversions look like tracks without hits before a certain layer. If there are several conversions matched to a cluster, double-track conversions are preferred over single-track conversions.

In CMS (Sect. 1.6.1.3) the reconstruction of photon conversions uses the full tracking [6]; see also Sect. 10.3. Electrons and positrons in particular are reconstructed by associating a track in the silicon tracker with a cluster in the ECAL [7]. Besides the standard seeding of tracks in the pixel detector, additional seeds are computed by extrapolating the electron/positron from the cluster toward the interaction vertex, using both charge hypotheses. Starting from these seeds, track candidates are built by the combinatorial Kalman filter (Sect. 5.1.7) and sent to track fit with the GSF (Sect. 6.2.3). Electron/positron candidates are then created from the association of a GSF track with a cluster in the ECAL. Track pairs of opposite charge sign are selected and filtered by various constraints: the two tracks must have small angular separation; the trajectories (helices) projected to the transverse plane must not intersect; and the presumptive vertex must be inside the tracker volume. Track pairs that pass these filters are sent to the vertex fit with the constraint that the tracks are parallel at the vertex; see Sect. 8.3. The track pair is kept if the p-value of the chi-square statistic indicates a good fit.

9.5 Hadronic Interactions

Knowledge of the material in a tracking device is important for understanding and tuning the track reconstruction algorithms. Reconstruction of hadronic interactions in the tracker material gives a precise map of the amount and location of active and inactive parts of the device. The hadronic interactions result in secondary vertices that have to be found. Whereas, in photon conversions, the vertex position along the photon direction is ill defined, the vertex position of hadronic interactions can be precisely estimated in all directions.

Tracks from secondary interaction vertices have a large impact parameter and may not be found by the standard track finder(s). In order to have the largest possible sample of such tracks, a special track finding pass can be implemented. Tracks from the primary vertex and short-lived decays are suppressed by a lower bound on the transverse impact parameter. Further quality cuts can be applied to select a reliable sample of secondary tracks. The following paragraphs describe examples from the experiments at the LHC.

In [8], a vertex finder is described that simultaneously finds all secondary vertices in an event in the ATLAS detector (Sect. 1.6.1.2). It starts by finding all possible intersections of pairs of selected tracks, performs a vertex fit and applies a quality cut. Wrong combinations can be further suppressed by requiring that neither track

has hits in the layers with a smaller radius than the vertex. Finally, nearby vertices are merged, and incompatibility between vertices sharing tracks are resolved via an incompatibility graph; see Sect. 5.3. For results on reconstruction efficiency and vertex resolution, see [8]. The search for hadronic interaction vertices can be complemented by the reconstruction of photon conversions; see Sect. 9.4.

A precision measurement of the structure of the CMS pixel detector (Sect. 1.6.1.3), including sensors and support material, and the beam pipe using hadronic interactions is reported in [9]. The track selection requires a minimal transverse momentum of $0.2\,\mathrm{GeV}/c$. For all pairs of selected tracks, their distance of closest approach is computed; if the distance is below a preset threshold, the midpoint of the points of closest approach is kept as a vertex candidate. These two-track vertices are then merged to larger vertices by agglomerative clustering, (Sect. 3.3.1). These vertices are sent to the adaptive vertex fit (AVF, Sect. 8.2.2).

The tracks associated to a fitted vertex are classified into incoming, outgoing, and merged tracks based on their transverse impact parameter and the number of hits before and after the vertex. At least three tracks have to be present, but not more than one incoming or merged tracks. Additional filters using the properties of the outgoing tracks serve to further reduce the number of fake interaction vertices. For results on resolution, efficiency and purity of the vertex reconstruction, see [9].

The Vertex Locator (VELO) of the LHCb experiment, see Sect. 1.6.1.4, was mapped by the reconstruction of secondary interaction vertices of hadrons produced in beam-gas collisions [10]. The data were collected during special runs in which helium or neon was injected into the collision region. The tracks used for the reconstruction of secondary vertices come from a modified track finding procedure that makes no prior assumption about the vertex position. Only well-reconstructed tracks with at least three 2D hits are selected. The secondary vertices are reconstructed from at least three tracks and have to be of good quality. They also must not be compatible with the primary vertex region. Only events with a single secondary vertex are used in the analysis. Results and the verification of the procedure with photon conversions to two muons can be found in [10].

References

1. W. Erdmann, Nucl. Instrum. Meth. Phys. Res. A **560**, 89 (2006)
2. F. Abe, et al., Phys. Rev. D **50**, 2966 (1994)
3. G. Segneri, F. Palla, Lifetime Based b-tagging with CMS. Technical Report CMS-NOTE-2002-046, CERN, Geneva (2002). https://cds.cern.ch/record/687465
4. M. Schmelling, Review of Measurements with HERA-B (2008). https://tinyurl.com/Schmelling. Talk given at the LHCb Workshop Neckarzimmern, March 2008
5. G. Aad, et al., Phys. Lett. B **719**(4), 280 (2013)
6. V. Khachatryan, et al., J. Instrum. **10**(08), P08010 (2015). https://iopscience.iop.org/article/10.1088/1748-0221/10/08/P08010
7. V. Khachatryan, et al., J. Instrum. **10**(06), P06005 (2015). https://iopscience.iop.org/article/10.1088/1748-0221/10/06/P06005

8. M. Aaboud, et al., J. Instrum. **11**(11), P11020 (2016). https://iopscience.iop.org/article/10.1088/1748-0221/11/11/P11020
9. A. Sirunyan, et al., J. Instrum. **13**(10), P10034 (2018). https://iopscience.iop.org/article/10.1088/1748-0221/13/10/P10034
10. M. Alexander, et al., J. Instrum. **13**(06), P06008 (2018). https://iopscience.iop.org/article/10.1088/1748-0221/13/06/P06008

Part IV
Case Studies

Chapter 10
LHC Experiments

Abstract The chapter gives an overview of the track and vertex reconstruction methods of the LHC experiments that were used in production during Run 2 of the LHC, which ended in autumn of 2018.

10.1 ALICE

Track reconstruction in ALICE [1, 2] starts with cluster finding in the two principal tracking detectors, i.e., in the pad rows of the TPC and in the silicon sensors of the Inner Tracking System (ITS, Sect. 1.6.1.1). The clusters reconstructed in the two innermost pixel layers of the ITS are used for a preliminary reconstruction of the primary vertex. Track reconstruction takes place through three main passes.

The first pass starts out in the outer part of the TPC and proceeds inwards through the ITS to the vertex region. Track seeds for primary particles are formed from pairs of measurements in the outer pad rows of the TPC and the primary vertex. Beginning from the seeds, track finding proceeds in the inward direction using a combinatorial Kalman filter (see Sect. 5.1.7).

Each reconstructed TPC track is extrapolated to the outermost layers of the ITS and used as a seed for track finding in the ITS. Track following in the ITS uses the particle hypothesis computed from the energy loss in the TPC, if available. Seven hypotheses are considered: electrons, muons, pions, kaons, protons, deuterons and tritons. If the $\mathrm{d}E/\mathrm{d}x$ information is missing or inconclusive, the pion mass is used. An illustration of how a particle hypothesis can be made using a combination of $\mathrm{d}E/\mathrm{d}x$ information (from the TPC measurements) and momentum information (from track reconstruction) is shown in Fig. 10.1.

A difficult point in the first-pass inwards tracking is the transition between the TPC and the ITS, due to the quite long (approximately 0.5 m) propagation distance between the two tracking systems. Given the uncertainty of the position of the track candidate propagated from the TPC and the high density of clusters in the ITS, many measurements are potentially compatible with the extrapolation. In order to lower the combinatorial complexity, the position coordinates of ITS hits used in finding

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_10

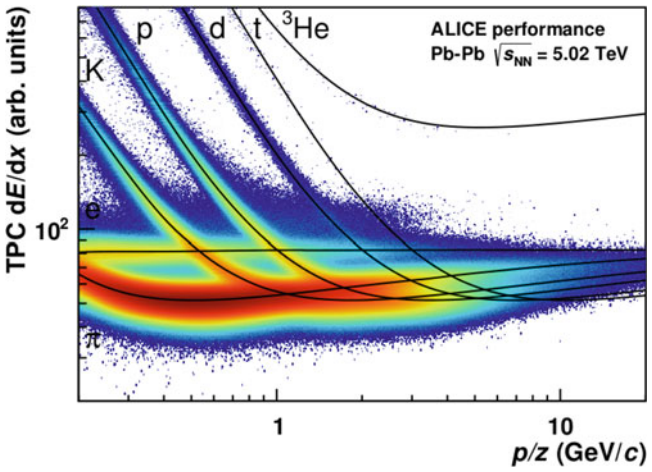

Fig. 10.1 Ionization energy loss as a function of momentum for a set of particles in the ALICE experiment. (From https://arxiv.org/abs/1701.04810. ©CERN for the benefit of the ALICE Collaboration. Reproduced under License CC-BY-4.0)

primary tracks are augmented by the two angles describing the direction from the measurement to the primary vertex, effectively making the hits four-dimensional. When all layers of the ITS are traversed, the best surviving candidate (in terms of a quality measure using for instance the total χ^2 of the track and the number of assigned clusters) of each combinatorial tree is selected. Additional track candidates are then found by a stand-alone search in the ITS, using hits not attached to TPC tracks.

During the outward propagation in the second-pass through ITS, TPC and transition radiation detector (TRD), the track length and seven time-of-flight hypotheses per track (corresponding to the seven particles mentioned above) are calculated. This information is subsequently used for particle identification in the time-of-flight detector (TOF). Whenever possible, tracks are matched with hits in the TOF and other detectors outside the TRD.

In the third pass towards the vertex region, measurements assigned to the surviving track candidates during the first pass are used in the refit. The primary vertex is again fitted, now using the full available information from the reconstructed tracks as well as the average position and spread of the beam-beam interaction region. For a review of the track reconstruction performance, see [2, 3].

The high-level trigger (HLT) of ALICE also performs GPU-accelerated track finding and fitting in the TPC [4]. A cellular automaton finds track seeds, which are then extended by a simplified Kalman filter. After track segments are merged to the final track candidates, a full Kalman filter track fit is performed. A detailed description of all stages of the HLT can be found in [5]. The current HLT tracking is the base for the future TPC tracking in Run 3, which will run in GPUs as well [6].

10.2 ATLAS

The main track reconstruction strategy in the ATLAS experiment (see Sect. 1.6.1.2) is to start out with track seeds in the innermost part of the Inner Detector and proceed outwards [7–9]. First clusters are assembled in the pixel and SCT detector subsystems by a connected component analysis (CCA) [10], and from these clusters 3D measurements (so-called space-points) are created. In dense environments such as the core of high-energy jets adjacent clusters from neighbouring particles can be so close that they overlap. Identifying such merged clusters correctly during the track reconstruction procedure is very important, and ATLAS has developed a method using a multi-layer, feed-forward neural network in order to solve this task efficiently [11].

Track seeds are generated from sets of three space-points. Seeds are processed iteratively by first considering sets of space-points from the SCT detector, then sets from the pixel detector, and finally sets originating from a combination of the two sub-detectors. Starting out from the seeds, track finding is carried out by a combinatorial Kalman filter (see Sect. 5.1.7). Each seed can potentially give rise to a number of track candidates, as long as the candidates pass a set of quality criteria. Efforts to speed up the track reconstruction are ongoing at the time of writing; first results based on improving the purity of the seed collection are given in [12]. More recent results are described in [13].

After iterating over all the seeds, the set of all track candidates are processed by an ambiguity solver. A flow diagram of the ATLAS ambiguity solver is shown in Fig. 10.2.

The track candidates are first ranked by a track score given to each candidate. The track score definition is intended to give high scores to candidates with a high probability of being a real track and therefore depends on, e.g., cluster resolutions,

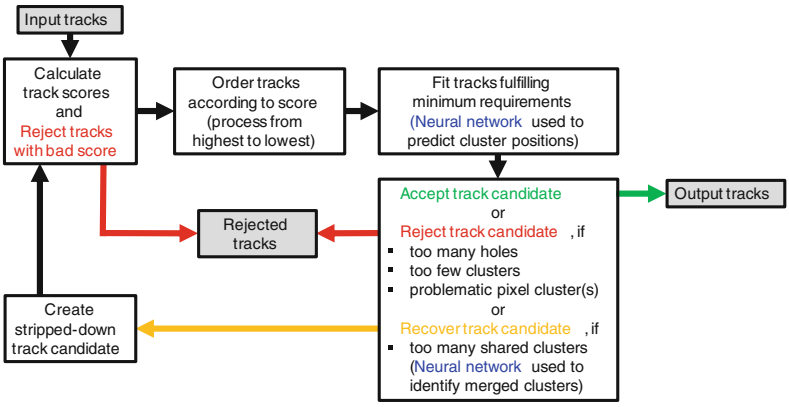

Fig. 10.2 Flow diagram of the ATLAS ambiguity solver. (From [9], reproduced under License CC-BY-4.0)

the number of holes on the track and the track χ^2. Track candidates passing a set of basic quality criteria are submitted to a full, high-resolution track fit using all available information relevant for an optimal estimation of the track parameters. After the track fit, track candidates can either be accepted, rejected or stripped down if the candidate contains too many clusters shared by other track candidates. The stripped-down candidates are scored again and re-submitted to the entire procedure of ambiguity resolving. It can be noted that the full track fit is invoked after the track scoring stage in order to save computational load.

The set of output track segments from the pixel and SCT detectors are submitted to the TRT track extension, which extrapolates the track segments through the TRT and searches for compatible segments in this drift tube detector. The search and subsequent resolution of drift-time ambiguities are done using a line fit in coordinate projections making the track model approximately linear [7]. At high levels of pile-up the occupancy of the TRT is so high that including drift-time hits not always increases track reconstruction performance. However, tube hits (i.e., using only the position of the centre straw) in all cases contribute to electron identification through information about transition radiation.

In order to find tracks originating, e.g., from secondary vertices or photon conversions, a secondary track reconstruction strategy starts out by finding track segments in the TRT using a Hough transform [7]. The TRT segments are then back-tracked into the SCT, which allows finding small track segments in the silicon detector that were not found during the first inside-out pass.

Electron reconstruction requires special attention [14] due to the potentially large amounts of emitted bremsstrahlung. In ATLAS, there is no separate iteration for electron track reconstruction. However, specific handling of bremsstrahlung is triggered by electromagnetic showers during the entire track reconstruction chain. The combinatorial Kalman filter allows for kinks in the track candidates if they are inside a region compatible with an electromagnetic shower. The ambiguity solver uses a global LS fit which allows for bremsstrahlung breakpoints in the track model. During electron identification, a full Gaussian-sum filter is invoked.

The primary vertex reconstruction in ATLAS is divided into the two classical categories vertex finding and vertex fitting [15]. The input is the set of reconstructed tracks in an event selected according to a set of quality criteria. This set is first used to obtain a seed position for the primary vertex. The seed position in the bending plane is the center of the beam spot, whereas the longitudinal coordinate is defined as the mode of the longitudinal position of the tracks at their respective points of closest approach to the beam spot. Subsequently, the set of tracks and the seed position is submitted to an iterative vertex finding procedure using the adaptive vertex fit (AVF) with annealing (Sect. 8.2.2). A typical distribution of track weights for a set of values of the temperature parameter T is shown in Fig. 10.3.

After the final iteration, the tracks with weights so small that they can be considered incompatible with the reconstructed vertex are removed from the vertex candidate and returned to the pool of unused tracks. The unused tracks are then submitted to a new iteration of the vertex finding procedure, which continues

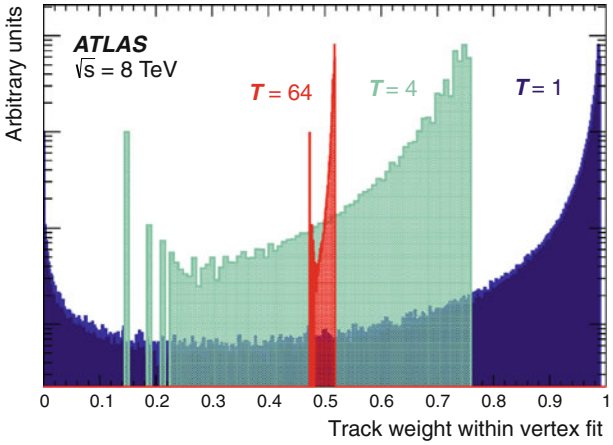

Fig. 10.3 Histogram of track weights in the adaptive vertex fit for a set of different temperatures. (From [15], reproduced under License CC-BY-4.0)

until all tracks have been used or no additional vertex can be found among the remaining tracks. For secondary vertex finding inside jets, see [16]. Stand-alone vertex reconstruction in the muon spectrometer of ATLAS is described in [17].

10.3 CMS

For a brief description of the CMS tracking system, see Sect. 1.6.1.3. Although the pixel detector and the silicon strip detector are mechanically independent sub-detectors with different sensor technology, they are considered as a single tracking detector as far as track finding and track fitting are concerned.

The CMS track reconstruction is based on an iterative approach [18–21]. The principal difference between iterations is the configuration of the seed generation and the target tracks. The iterative search starts with the tracks that are the least difficult to find, and proceeds to more difficult classes: low momentum tracks, tracks from short-lived decays, and tracks from long-lived decays. In each iteration, hits associated to high-quality tracks are masked, reducing the combinatorial overhead for the following iteration. Finally, special iterations that improve track reconstruction in high-density environments such as jets are executed, followed by iterations targeting muon tracks in combination with the muon chambers. For a recent comprehensive overview of the tracking performance, see [22].

Each iteration consists of four steps:

1. *Seed generation.* Initial track segments, called seeds, are found by a combinatorial search or by a cellular automaton [23]; see also Sect. 5.1.5. Using the

information of three or four hits pixel and/or strips, the trajectory parameters and the corresponding uncertainties of the seed are computed.
2. *Track finding.* Each seed is used as the starting segment of the combinatorial Kalman filter (Sect. 5.1.7). At most one missing hit is allowed in a track candidate.
3. *Track fitting.* A Kalman filter or, for electron candidates, a Gaussian-sum filter/smoother (Sect. 6.2.3) is performed to obtain the final estimate of the track parameters at the interaction point exploiting the full trajectory information. The Kalman filter is also available in a parallelized version [24–26].
4. *Track classification.* Tracks are divided into classes according to different track quality criteria; see also Sect. 6.4.

The twelve tracking iterations used after the pixel upgrade to four layers are listed in Table 10.1, showing the seed type and the targeted tracks [20]. Iteration 9 is special insofar as it improves track reconstruction in jets and hadronic τ lepton decays by re-clustering pixel hits, using the jet direction to predict the expected cluster shape and charge [27, 28]. Iterations 10 and 11 reconstruct global muon tracks. For the combined electron reconstruction with tracker and electromagnetic calorimeter, see [29] and Sect. 9.4.

Primary vertex reconstruction starts with selecting tracks that are produced promptly by setting a threshold on their transverse impact parameter [18]. The selected tracks are then clustered on the basis of their z-coordinates at their point of closest approach to the center of the beam spot, allowing for an arbitrary

Table 10.1 List of the tracking iterations in CMS used after the Pixel Tracker upgrade with the corresponding seeding configuration used and target tracks [20]

Iteration	Step Name	Seeding	Target Track
0	Initial	pixel quadruplets	prompt, high p_T
1	LowPtQuad	pixel quadruplets	prompt, low p_T
2	HighPtTriplet	pixel triplets	prompt, high p_T recovery
3	LowPtTriplet	pixel triplets	prompt, low p_T
4	DetachedQuad	pixel quadruplets	from B hadron decay, $r \leq 5$ cm
5	DetachedTriplet	pixel triplets	from B hadron decay, $r \leq 10$ cm
6	MixedTriplet	pixel+strip triplets	displaced, $r \leq 7$ cm
7	PixelLess	inner strip pairs	displaced, $r \leq 25$ cm
8	TobTec	outer strip pairs	displaced, $r \leq 60$ cm
9	JetCore	pixel pairs in jets	high-p_T jets
10	Muon inside-out	muon-tagged tracks	muons
11	Muon outside-in	standalone muons	muons

number of clusters. The algorithm of choice is clustering by deterministic annealing (Sect. 7.2.4). Vertex candidates with at least two tracks are fitted by an adaptive vertex fitter (Sect. 8.2.2). Finally the signal vertex is determined as the primary vertex with the highest weighted sum of the squared momenta of the jets and isolated tracks associated to the vertex.

Iterations 4 and 5 of the track finding target secondary tracks from short-lived decays, for instance B hadrons. Other candidate tracks for inclusion in a secondary vertex are tracks rejected by the adaptive vertex fit of the signal vertex. Finding secondary vertices is then basically a combinatorial search, followed by a vertex fit and possibly a kinematic fit for each candidate.

Tracks from long-lived decays, for instance K-short mesons and Λ baryons, are the targets of iterations 6–8, each for a different range of the radial distance r of the decay from the beam line. Reconstruction of these decay vertices is again a combinatorial search followed by a vertex fit and a kinematic fit.

CMS also has an independent reconstruction of tracks and primary vertices based purely on pixel hits. This is significantly faster than the standard track and vertex reconstruction chain and therefore a valuable tool for the high-level trigger [18, 30].

10.4 LHCb

Track reconstruction in the LHCb detector uses hits in the VELO, TT and T stations; see Sect. 1.6.1.4 and Fig. 10.4. Depending on which detectors are crossed by a particle, different track types are defined [31]:

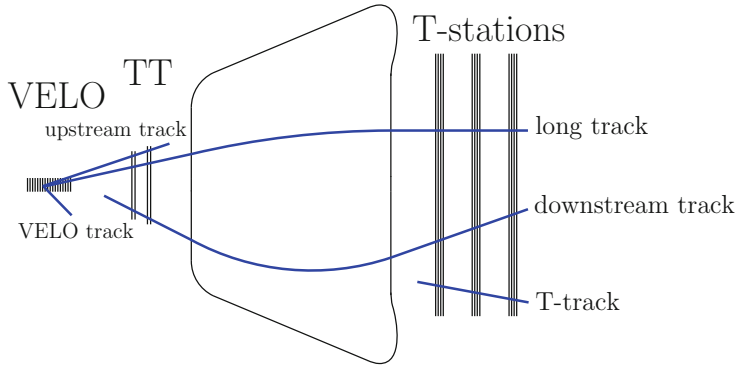

Fig. 10.4 Schematic diagram of the LHCb tracking system and the five track types. (From [31], reproduced under License CC-BY-4.0)

- *VELO tracks* have hits in the VELO. They are particularly important in the primary vertex reconstruction. They can be extended to upstream and long tracks during reconstruction.
- *Upstream tracks* have hits in the VELO and TT stations. In general their momentum is too low to traverse the magnet and reach the T stations. If they do, they can be extended to long tracks.
- *Long tracks* have hits in both the VELO and the T stations, and optionally in TT. They are the most important set of tracks for physics analyses.
- *Downstream tracks* have hits in the TT and T stations. They are important for the reconstruction of long-lived particles that decay outside the VELO acceptance.
- *T tracks* pass only through the T stations. Like the downstream tracks, they are useful for particle identification in the Ring Imaging Cherenkov detectors.

Track reconstruction is done in the two stages of the high-level trigger, HLT1 and HLT2 (Sect. 2.1.3). In fact, the full event reconstruction runs in the high-level trigger [31, 32].

The partial track reconstruction in HLT1 starts with finding straight lines in the VELO, which are fitted by a simplified Kalman filter. The fitted tracks are used for the reconstruction of the primary vertices. The primary vertex finding, see [33] and [34, Appendix A], proceeds in two steps: seeding and fitting. The 3D seeding algorithm starts with a base track and looks for tracks close to it. If at least four close tracks are found, a robust average of the points of closest approach of all track pairs is computed and stored as a seed. The tracks are marked as used and the next base track is processed. The fast seeder clusters tracks according to the z-coordinates of their points of closest approach to beam line, similar to the primary vertex finder of CMS (see Sect. 10.3).

Before fitting, the seeds are ordered in descending track multiplicity, so that primary tracks are fitted first, and the probability of incorrectly labeling a secondary vertex as primary is minimized. The vertex fit is a redescending M-estimator with Tukey's biweight function, see Table 6.1 in Sect. 6.2.1. HLT1 proceeds with forming upstream tracks by extrapolating VELO tracks to the TT. High-momentum tracks are extended to the T stations. The resulting long tracks are fitted with a full Kalman filter and sent to the fake track rejection by a neural network [35].

HLT2 performs the full track reconstruction, in the sequence shown in Fig. 10.5. VELO tracks are extended to long tracks, without a threshold on the transverse momentum. Next, a stand-alone track finding is done in the T stations [36]. The found tracks are combined with VELO to long tracks. Tracks produced in the decays of long-lived particles outside the VELO are reconstructed from track segments in the T stations that are extrapolated backwards and combined with hits in the TT. The final steps are a full track fit with the Kalman filter, fake track rejection by a neural network, and removal of duplicates.

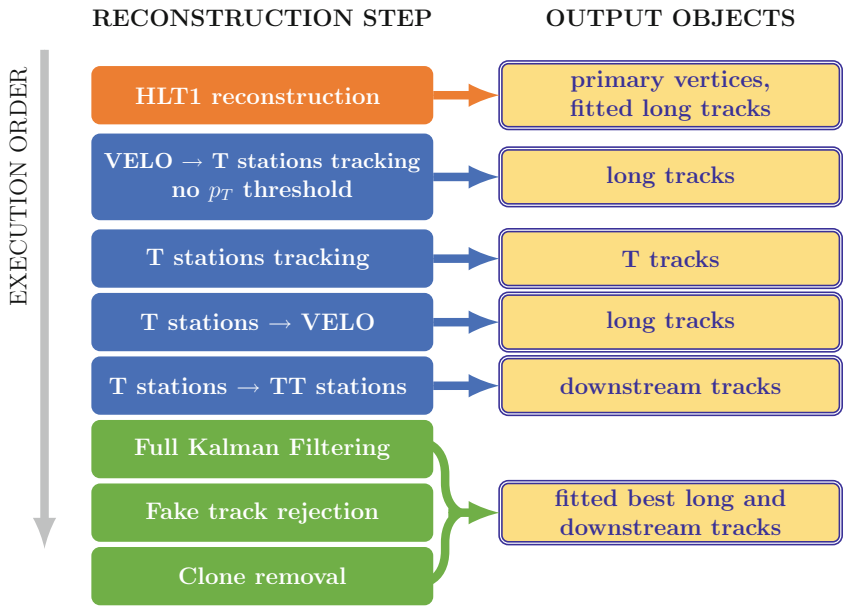

Fig. 10.5 Sequence of the full track reconstruction in LHCb. (From [31], reproduced under License CC-BY-4.0)

References

1. I. Belikov and ALICE Collaboration, EPJ Web Conf. **70**, 00029 (2014). https://doi.org/10.1051/epjconf/20147000029
2. B. Abelev, et al., Int. J. Mod. Phys. A **29**, 1430044 (2014). https://www.worldscientific.com/doi/pdf/10.1142/S0217751X14300440
3. D. Rohr, for the ALICE collaboration, in *Proceedings of the 5th Large Hadron Collider Physics Conference (LHCP 2017)* (2017). https://arxiv.org/pdf/1709.00618.pdf
4. D. Rohr, S. Gorbunov, V. Lindenstruth, ALICE Collaboration, J. Phys. Conf. Ser. **898**(3), 032030 (2017). https://iopscience.iop.org/article/10.1088/1742-6596/898/3/032030
5. S. Acharya, et al., Real-time data processing in the ALICE High Level Trigger at the LHC. Technical Report CERN-EP-2018-337, CERN, Geneva (2018). https://arxiv.org/pdf/1812.08036.pdf
6. P. Buncic, M. Krzewicki, P. Vande Vyvre, Technical Design Report for the Upgrade of the Online-Offline Computing System. Technical Report ALICE-TDR-019, CERN, Geneva (2015). https://cds.cern.ch/record/2011297
7. T. Cornelissen, et al., J. Phys. Conf. Ser. **119**(3), 032014 (2008)
8. A. Salzburger, J. Phys.: Conf. Ser. **664**(7), 072042 (2015). https://iopscience.iop.org/article/10.1088/1742-6596/664/7/072042
9. M. Aaboud and others, Eur. Phys. J. C **77**(10), 673 (2017)
10. A. Rosenfeld, J.L. Pfaltz, J. ACM **13**(4), 471 (1966)
11. G. Aad, et al., J. Instrum. **9**(09), P09009 (2014)
12. The ATLAS Collaboration, Fast Track Reconstruction for HL-LHC. Technical Report ATL-PHYS-PUB-2019-041, CERN, Geneva (2019). http://cds.cern.ch/record/2693670/files/ATL-PHYS-PUB-2019-041.pdf

13. F. Klimpel (On behalf of the ATLAS collaboration), Fast tracking for the HL-LHC ATLAS detector. Technical Report ATL-PHYS-PROC-2020-044, CERN, Geneva (2020). https://cds.cern.ch/record/2721751
14. ATLAS Collaboration, Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton-proton collision data at $\sqrt{s} = 13$ TeV. Technical Report CERN-EP-2018-273, CERN, Geneva (2019). https://cds.cern.ch/record/2657964
15. M. Aaboud and others, Eur. Phys. J. C **77**(5), 332 (2017)
16. S. Heer, ATLAS Collaboration, in *Proceedings of the European Physical Society Conference on High Energy Physics* (2017), p. 762. https://pos.sissa.it/314/762
17. The ATLAS collaboration, J. Instrum. **9**(02), P02001 (2014). https://iopscience.iop.org/article/10.1088/1748-0221/9/02/P02001
18. S. Chatrchyan, et al., J. Instrum. **9**(10), P10009 (2014). https://iopscience.iop.org/article/10.1088/1748-0221/9/10/P10009/meta
19. M. Rovere, on behalf of the CMS Collaboration, J. Phys. Conf. Ser. **664**(7), 072040 (2015). https://iopscience.iop.org/article/10.1088/1742-6596/664/7/072040
20. CMS Collaboration, 2017 tracking performance plots. Technical Report CMS-DP-2017-015, CERN, Geneva (2017). http://cds.cern.ch/record/2290524
21. A.M. Sirunyan, et al., J. Instrum. **12**(10), P10003 (2017). https://iopscience.iop.org/article/10.1088/1748-0221/12/10/P10003
22. E. Brondolin, Track reconstruction in the CMS experiment for the High Luminosity LHC. Ph.D. thesis, Technische Universität Wien (2018). http://cds.cern.ch/record/2308020
23. F. Pantaleo, New Track Seeding Techniques for the CMS Experiment. Ph.D. thesis, Universität Hamburg (2017). https://cds.cern.ch/record/2293435
24. G. Cerati, et al., J. Phys. Conf. Ser. **1085**(4), 042016 (2018). https://iopscience.iop.org/article/10.1088/1742-6596/1085/4/042016
25. G. Cerati, et al., in *Proceedings of the CTD/WIT 2019* (2019). https://arxiv.org/pdf/1906.11744
26. G. Cerati, et al., Submitted to J. Instrum. (2020). https://arxiv.org/pdf/2006.00071
27. A. Rizzi, High pt jets tracking. CMSPublic Web. https://twiki.cern.ch/twiki/bin/view/CMSPublic/HighPtTrackingDP
28. G. Cerati, PoS **Vertex2014**, 037 (2015). https://pos.sissa.it/227/037
29. V. Khachatryan, et al., J. Instrum. **10**(06), P06005 (2015). https://iopscience.iop.org/article/10.1088/1748-0221/10/06/P06005
30. S. Cucciarelli, et al., Track reconstruction, primary vertex finding and seed generation with the Pixel Detector. Technical Report CMS-NOTE-2006-026, CERN, Geneva (2006). https://cds.cern.ch/record/927384
31. R. Aaij, et al., Performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC. Technical Report CERN-LHCb-DP-2019-001, CERN, Geneva (2018). https://cds.cern.ch/record/2652801
32. R. Aaij, et al., A comprehensive real-time analysis model at the LHCb experiment. Technical Report CERN-LHCb-DP-2019-002, CERN, Geneva (2019). https://cds.cern.ch/record/2665946
33. M. Kucharczyk, P. Morawski, M. Witek, Primary Vertex Reconstruction at LHCb. Technical Report CERN-LHCb-PUB-2014-044, CERN, Geneva (2014). https://cds.cern.ch/record/1756296
34. A. Dziurda, Studies of time-dependent CP violation in charm decays of B_s^0 mesons. Ph.D. thesis, Polish Academy of Sciences (2015). https://cds.cern.ch/record/2115353
35. M. De Cian, et al., Fast neural-net based fake track rejection in the LHCb reconstruction. Technical Report CERN-LHCb-PUB-2017-011, CERN, Geneva (2017). https://cds.cern.ch/record/2255039
36. O. Callot, M. Schiller, PatSeeding: A Standalone Track Reconstruction Algorithm. Technical Report CERN-LHCb-2008-042, CERN, Geneva (2008). https://cds.cern.ch/record/1119095

Chapter 11
Belle II and CBM

Abstract The chapter gives an overview of the tracking and vertexing algorithms of two experiments not at the LHC, Belle II at SuperKEKB in Japan and CBM at FAIR in Germany.

11.1 Belle II

The Belle II experiment [1] started operation in late 2018. The following overview of the reconstruction methods reflects the status after a couple of months of running; further developments and adaptations that have been implemented in parallel to the rising luminosity of the SuperKEKB collider are documented in [2].

The tracking system of Belle II has two principal components, the vertex detector (VXD) and the central drift chamber (CDC); see Sect. 1.6.2.1. The VXD consists of two parts: the PXD with two layers of DEPFET pixel sensors, and the SVD with four layers of double-sided silicon strip sensors. The overall design of the track reconstruction is shown in Fig. 11.1.

Track finding in the CDC starts with a filter, implemented as a boosted decision tree, that rejects background hits. This is followed by two track finders. The global track finder employs a Legendre transform (Sect. 5.1.4), to find complete tracks originating from the interaction region. Peak finding in the Legendre space is done by a fast iterative quadtree algorithm [3]. The local track finder builds segments from the hits in a superlayer. This is followed by a multi-stage combination and filter algorithm that connects valid segments and tracks, rejects fake segments, and outputs the final CDC track sample [4].

Stand-alone track finding in the SVD [5, 6] is based on a cellular automaton (CA) complemented by a sector map (Sect. 5.1.5). The sector map stores prior information about which triplets of hits can be part of a track. The vertex position can be used as a virtual hit. Track segments that pass all cuts form the cells of the CA. Figure 11.2 shows the final state of the CA with a toy event, without background.

When the CA has finished, each track candidate is assigned a quality index, which is the p-value of a triplet fit (Sect. 6.1.5), a circle fit, or a Riemann helix

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0_11

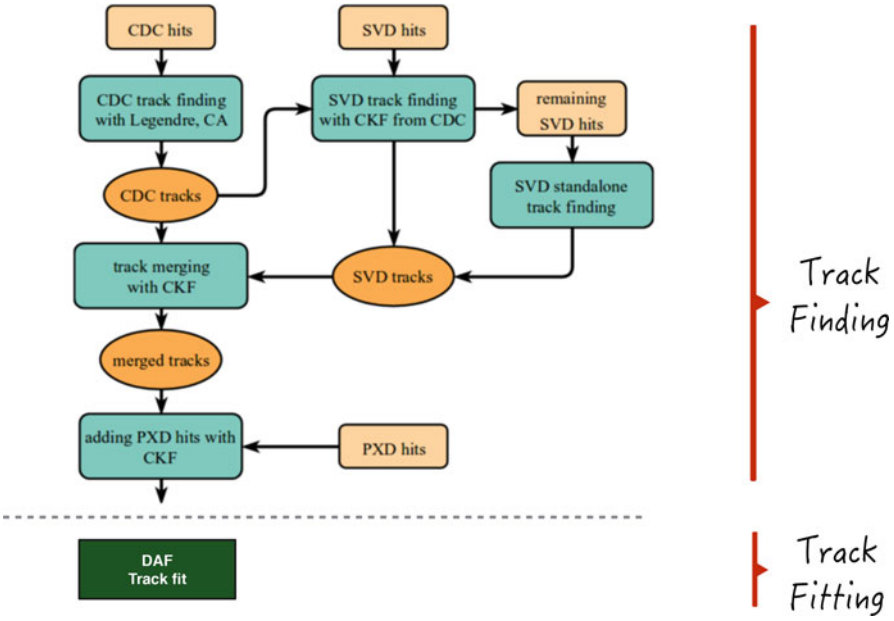

Fig. 11.1 Scheme of the track reconstruction in Belle II. (Courtesy of B. Scavino, Mainz)

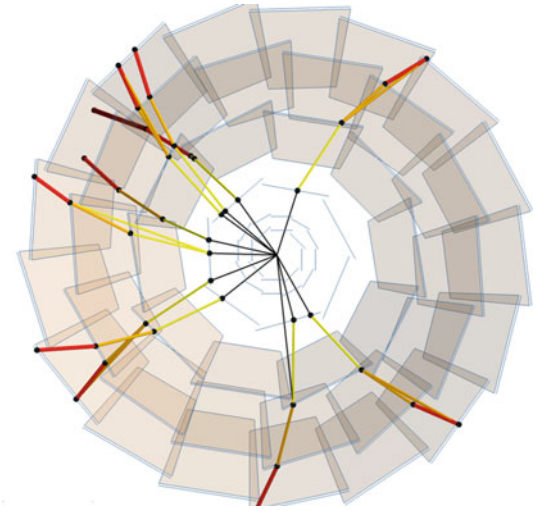

Fig. 11.2 Final state of the CA with a toy event in the VXD. The colors of the cells (track segments) represent their states. (Courtesy of J. Lettenbichler, Vienna)

fit (Sect. 6.3). Finally, non-overlapping candidates are selected, either with a greedy algorithm or a Hopfield network (Sect. 5.3). At the time of writing, the triplet fit and the greedy algorithm are the defaults.

The merging of CDC and VXD tracks is based on a boosted decision tree that is trained on valid combinations of CDC and VXD tracks. It takes the track parameters

as determined by a Deterministic Annealing Filter (DAF, Sect. 6.2.2) as input and returns a score in the interval (0, 1). Combinations above a threshold are accepted. CDC tracks, SVD tracks and combined tracks are extrapolated into the PXD to pick up hits by a combinatorial Kalman filter [7], and then passed to the DAF. The implementation of the DAF is the one in GENFIT [8–10].

The B mesons produced by the decay of the $\Upsilon(4S)$ resonance are reconstructed hierarchically [11], and their decay chains are fitted by a tree fitter based on [12]. V0 decays are reconstructed by pairing positive and negative tracks and performing a vertex fit [13].

11.2 CBM

CBM is a heavy-ion experiment at FAIR [14], currently in the preparation phase. The first beam is expected in 2024. The interaction rate will be up to 10 MHz, with up to 1000 charged particles produced in a central collision. Full online event reconstruction and selection will be required [15]. This is the task of the First Level Event Selection Package (FLES, [16, 17]). The algorithms in FLES exploit vector instructions for intra-processor parallelism and parallelism between cores on the event level.

The FLES package consists of several modules: track finding, track fitting, short-lived particles finding, event building and event selection [18], see Fig. 11.3. CBM runs without a trigger; time-stamped data will be put into a buffer in time slices of a certain length instead. The association of tracker hits with events is performed by software [18], using the hit time information in the Silicon Tracking System (STS, see Sect. 1.6.2.2). The resolution of the hit time is 5 ns.

Track finding in the STS (Sect. 1.6.2.2) is done by a 4D Cellular Automaton, where the fourth dimension is time [19, 20]. The cells of the CA are triplets of hits correlating in space and in time. The principle of the CA track finder is illustrated in Fig. 5.5 in Sect. 5.1.5. For track finding efficiencies, see [18].

The track candidates from the CA are sent to the Kalman Filter Track Fitter. The Kalman Filter is "SIMDized", i.e., uses the Single-Instruction Multiple-Data features of the processors employed [21]. The extrapolation in the inhomogeneous magnetic field uses the approximate analytical method described in Sect. 4.3.2.2.

After the primary vertex fit with the Kalman Filter, tracks are sorted into primary and secondary tracks. The reconstruction of short-lived particles is done by the KFParticle package [22], also based on the Kalman Filter and designed to fit decay chains. The reconstruction of a short-lived particle is implemented iteratively. The KFParticle package starts with an initial approximation of the secondary vertex, adds particles one after another, refines the state vector and gives the optimal values after the last particle [16, 17]. Besides the geometrical constraints, mass constraints and topological constraints can be imposed on the secondary vertex.

Track reconstruction in the three stations of the transition radiation detector is done by a cellular automaton as well [23].

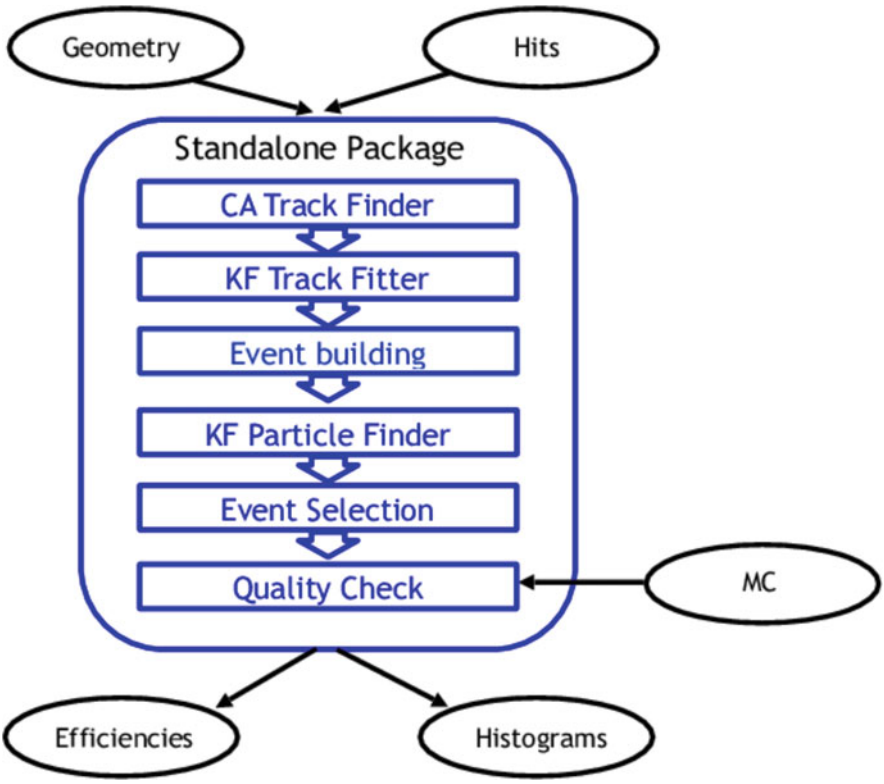

Fig. 11.3 Flow diagram of the First Level Event Selection Package in CBM. (From [17], reproduced under License CC-BY-4.0)

References

1. J. Kahn, in *CERN-BINP Workshop for Young Scientists in e^+e^- Colliders*, ed. by V. Brancolini, L. Linssen (2017). http://cds.cern.ch/record/2301997
2. V. Bertacchi, et al. Track Finding at Belle II (2020). https://arxiv.org/pdf/2003.12466.pdf. To appear in Computer Physics Communications
3. V. Trusov, Development of Pattern Recognition Algorithms for the Central Drift Chamber of the Belle II Detector. Ph.D. thesis, KIT, Karlsruhe (2016). https://publikationen.bibliothek.kit.edu/1000063754
4. T. Hauth, for the Belle II Collaboration. Pattern Recognition at Belle II (2016). https://docs.belle2.org/record/555. Talk given at the International Workshop on Future Linear Colliders, Morioka
5. J. Lettenbichler, Real-time pattern recognition in the central tracking detector of the Belle II Experiment. Ph.D. thesis, TU Wien (2016). http://repositum.tuwien.ac.at/obvutwhs/download/pdf/1656041
6. T. Bilka, et al., EPJ Web Conf. **150**, 00007 (2017). https://doi.org/10.1051/epjconf/201715000007

7. N.L. Braun, Combinatorial Kalman Filter and High Level Trigger Reconstruction for the Belle II Experiment. Ph.D. thesis, KIT, Karlsruhe (2019). https://publikationen.bibliothek.kit.edu/1000089317
8. J. Rauch, T. Schlüter, GENFIT—a Generic Track-Fitting Toolkit (2016). https://arxiv.org/abs/1410.3698v2
9. GENFIT—a generic toolkit for track reconstruction for experiments in particle and nuclear physics. http://genfit.sourceforge.net
10. T. Bilka, et al., Submitted to Nucl. Instrum. Meth. Phys. Res. A (2019). https://arxiv.org/pdf/1902.04405.pdf
11. T. Keck, Machine learning algorithms for the Belle II experiment and their validation on Belle data. Ph.D. thesis, KIT, Karlsruhe (2017). https://publikationen.bibliothek.kit.edu/1000078149
12. W.D. Hulsbergen, Nucl. Instrum. Meth. Phys. Res. A **552**(3), 566 (2005)
13. T. Schlüter, PoS **Vertex2014**, 038 (2015). https://pos.sissa.it/227/038
14. P. Senger, the CBM Collaboration, J. Phys. Conf. Ser. **50**, 357 (2006). https://iopscience.iop.org/article/10.1088/1742-6596/50/1/048
15. I. Kisel, Nucl. Instrum. Meth. Phys. Res. A **566**(1), 85 (2006)
16. I. Kisel, I. Kulakov, M. Zyzak, IEEE Trans. Nucl. Sci. **60**(5), 3703 (2013)
17. V. Akishina, I. Kisel, EPJ Web Conf. **173**, 01002 (2018). https://doi.org/10.1051/epjconf/201817301002
18. V. Akishina, I. Kisel, EPJ Web Conf. **127**, 00003 (2016). https://doi.org/10.1051/epjconf/201612700003
19. V. Akishina, I. Kisel, J. Phys. Conf. Ser. **599**(1), 012024 (2015). https://iopscience.iop.org/article/10.1088/1742-6596/599/1/012024
20. V. Akishina, I. Kisel, IEEE Trans. Nucl. Sci. **62**(6), 3172 (2015)
21. S. Gorbunov, et al., Comput. Phys. Commun. **178**(5), 374 (2008)
22. S.Gorbunov, I. Kisel, Reconstruction of decayed particles based on the Kalman filter. Technical Report CBM-SOFT-note-2007-003, STAR experiment, Brookhaven National Laboratory (2007). https://tinyurl.com/Gorbunov-KFParticle
23. A. Bubak, et al., Acta Phys. Pol. B **41**, 3 (2010). https://www.actaphys.uj.edu.pl/fulltext?series=Reg&vol=41&page=3

Appendix A
Jacobians of the Parameter Transformations

Transformation from One Curvilinear Frame to Another

The non-zero terms of the Jacobian matrix are:

$$\psi: \qquad \partial\psi/\partial\psi_0 = 1,$$

$$\begin{aligned}
\phi: \qquad \partial\phi/\partial\psi_0 &= -\frac{\alpha K}{\cos\lambda}\cdot\frac{1}{\psi}\cdot(\boldsymbol{n}\cdot\boldsymbol{u})\cdot[\boldsymbol{t}\cdot(\boldsymbol{r}_0-\boldsymbol{r})]\,,\\
\partial\phi/\partial\phi_0 &= \frac{\cos\lambda_0}{\cos\lambda}\Big\{\cos\theta\cdot(\boldsymbol{u}_0\cdot\boldsymbol{u})+\sin\theta\cdot((\boldsymbol{h}\times\boldsymbol{u}_0)\cdot\boldsymbol{u})\\
&\quad+\ (1-\cos\theta)\cdot(\boldsymbol{h}\cdot\boldsymbol{u}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{u})\\
&\quad+\ \alpha\,(\boldsymbol{n}\cdot\boldsymbol{u})\,[-\sin\theta\,(\boldsymbol{u}_0\cdot\boldsymbol{t})+\alpha\,(1-\cos\theta)\,(\boldsymbol{u}_0\cdot\boldsymbol{n})\\
&\quad-\ (\theta-\sin\theta)\,(\boldsymbol{h}\cdot\boldsymbol{t})\,(\boldsymbol{h}\cdot\boldsymbol{u}_0)]\Big\},\\
\partial\phi/\partial\lambda_0 &= \frac{1}{\cos\lambda}\Big\{\cos\theta\cdot(\boldsymbol{v}_0\cdot\boldsymbol{u})+\sin\theta\cdot((\boldsymbol{h}\times\boldsymbol{v}_0)\cdot\boldsymbol{u})\\
&\quad+\ (1-\cos\theta)\cdot(\boldsymbol{h}\cdot\boldsymbol{v}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{u})\\
&\quad+\ \alpha\,(\boldsymbol{n}\cdot\boldsymbol{u})\,[-\sin\theta\,(\boldsymbol{v}_0\cdot\boldsymbol{t})+\alpha\,(1-\cos\theta)\,(\boldsymbol{v}_0\cdot\boldsymbol{n})\\
&\quad-\ (\theta-\sin\theta)\,(\boldsymbol{h}\cdot\boldsymbol{t})\,(\boldsymbol{h}\cdot\boldsymbol{v}_0)]\Big\},
\end{aligned}$$

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0

$$\partial\phi/\partial x_{\perp 0} = -\frac{\alpha K}{\cos\lambda}\,(\boldsymbol{n}\cdot\boldsymbol{u})\,(\boldsymbol{u}_0\cdot\boldsymbol{t})\,,$$
$$\partial\phi/\partial y_{\perp 0} = -\frac{\alpha K}{\cos\lambda}\,(\boldsymbol{n}\cdot\boldsymbol{u})\,(\boldsymbol{v}_0\cdot\boldsymbol{t})\,,$$

λ :
$$\partial\lambda/\partial\psi_0 = -\alpha K\cdot\frac{1}{\psi}\cdot(\boldsymbol{n}\cdot\boldsymbol{v})\cdot[\boldsymbol{t}\cdot(\boldsymbol{r}_0-\boldsymbol{r})]\,,$$
$$\begin{aligned}\partial\lambda/\partial\phi_0 = \cos\lambda_0\{&\cos\theta\cdot(\boldsymbol{u}_0\cdot\boldsymbol{v}) + \sin\theta\cdot((\boldsymbol{h}\times\boldsymbol{u}_0)\cdot\boldsymbol{v})\\ &+\ (1-\cos\theta)\cdot(\boldsymbol{h}\cdot\boldsymbol{u}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{v})\\ &+\ \alpha\,(\boldsymbol{n}\cdot\boldsymbol{v})\,[-\sin\theta\,(\boldsymbol{u}_0\cdot\boldsymbol{t}) + \alpha\,(1-\cos\theta)\,(\boldsymbol{u}_0\cdot\boldsymbol{n})\\ &-\ (\theta-\sin\theta)\,(\boldsymbol{h}\cdot\boldsymbol{t})\,(\boldsymbol{h}\cdot\boldsymbol{u}_0)]\},\end{aligned}$$
$$\begin{aligned}\partial\lambda/\partial\lambda_0 = \cos\theta\cdot(\boldsymbol{v}_0\cdot\boldsymbol{v}) &+ \sin\theta\cdot((\boldsymbol{h}\times\boldsymbol{v}_0)\cdot\boldsymbol{v})\\ &+\ (1-\cos\theta)\cdot(\boldsymbol{h}\cdot\boldsymbol{v}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{v})\\ &+\ \alpha\,(\boldsymbol{n}\cdot\boldsymbol{v})\,[-\sin\theta\,(\boldsymbol{v}_0\cdot\boldsymbol{t}) + \alpha\,(1-\cos\theta)\,(\boldsymbol{v}_0\cdot\boldsymbol{n})\\ &-\ (\theta-\sin\theta)\,(\boldsymbol{h}\cdot\boldsymbol{t})\,(\boldsymbol{h}\cdot\boldsymbol{v}_0)],\end{aligned}$$
$$\partial\lambda/\partial x_{\perp 0} = -\alpha K\,(\boldsymbol{n}\cdot\boldsymbol{v})\,(\boldsymbol{u}_0\cdot\boldsymbol{t})\,,$$
$$\partial\lambda/\partial y_{\perp 0} = -\alpha K\,(\boldsymbol{n}\cdot\boldsymbol{v})\,(\boldsymbol{v}_0\cdot\boldsymbol{t})\,,$$

$x_\perp$:
$$\partial x_\perp/\partial\psi_0 = \frac{1}{\psi}\cdot[\boldsymbol{u}\cdot(\boldsymbol{r}_0-\boldsymbol{r})]\,,$$
$$\begin{aligned}\partial x_\perp/\partial\phi_0 = \cos\lambda_0\Bigg\{&\frac{\sin\theta}{K}\,(\boldsymbol{u}_0\cdot\boldsymbol{u}) + \frac{1-\cos\theta}{K}\,((\boldsymbol{h}\times\boldsymbol{u}_0)\cdot\boldsymbol{u})\\ &+\ \frac{\theta-\sin\theta}{K}\,(\boldsymbol{h}\cdot\boldsymbol{u}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{u})\Bigg\}\,,\end{aligned}$$
$$\begin{aligned}\partial x_\perp/\partial\lambda_0 = &\frac{\sin\theta}{K}\,(\boldsymbol{v}_0\cdot\boldsymbol{u}) + \frac{1-\cos\theta}{K}\,((\boldsymbol{h}\times\boldsymbol{v}_0)\cdot\boldsymbol{u})\\ &+\ \frac{\theta-\sin\theta}{K}\,(\boldsymbol{h}\cdot\boldsymbol{v}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{u})\,,\end{aligned}$$
$$\partial x_\perp/\partial x_{\perp 0} = \boldsymbol{u}_0\cdot\boldsymbol{u},$$
$$\partial x_\perp/\partial y_{\perp 0} = \boldsymbol{v}_0\cdot\boldsymbol{u},$$

$$
\begin{aligned}
y_\perp: \qquad \partial y_\perp/\partial\psi_0 &= \frac{1}{\psi}\cdot[\boldsymbol{v}\cdot(\boldsymbol{r}_0-\boldsymbol{r})]\,,\\
\partial y_\perp/\partial\phi_0 &= \cos\lambda_0\left\{\frac{\sin\theta}{K}(\boldsymbol{u}_0\cdot\boldsymbol{v})+\frac{1-\cos\theta}{K}((\boldsymbol{h}\times\boldsymbol{u}_0)\cdot\boldsymbol{v})\right.\\
&\quad\left.+\frac{\theta-\sin\theta}{K}(\boldsymbol{h}\cdot\boldsymbol{u}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{v})\right\},\\
\partial y_\perp/\partial\lambda_0 &= \frac{\sin\theta}{K}(\boldsymbol{v}_0\cdot\boldsymbol{v})+\frac{1-\cos\theta}{K}((\boldsymbol{h}\times\boldsymbol{v}_0)\cdot\boldsymbol{v})\\
&\quad+\frac{\theta-\sin\theta}{K}(\boldsymbol{h}\cdot\boldsymbol{v}_0)\cdot(\boldsymbol{h}\cdot\boldsymbol{v})\,,\\
\partial y_\perp/\partial x_{\perp 0} &= \boldsymbol{u}_0\cdot\boldsymbol{v},\\
\partial y_\perp/\partial y_{\perp 0} &= \boldsymbol{v}_0\cdot\boldsymbol{v}.
\end{aligned}
$$

Transformations Between a Local Frame and the Curvilinear Frame

$$
\partial\,(\psi,\phi,\lambda,x_\perp,y_\perp)/\partial\left(\psi,v',w',v,w\right)=
$$

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0\\
0 & \dfrac{(\boldsymbol{t}\cdot\boldsymbol{i})\,(\boldsymbol{u}\cdot\boldsymbol{j})}{\cos\lambda} & \dfrac{(\boldsymbol{t}\cdot\boldsymbol{i})\,(\boldsymbol{u}\cdot\boldsymbol{k})}{\cos\lambda} & -\dfrac{\alpha K\,(\boldsymbol{t}\cdot\boldsymbol{j})\,(\boldsymbol{u}\cdot\boldsymbol{n})}{\cos\lambda} & -\dfrac{\alpha K\,(\boldsymbol{t}\cdot\boldsymbol{k})\,(\boldsymbol{u}\cdot\boldsymbol{n})}{\cos\lambda}\\
0 & (\boldsymbol{t}\cdot\boldsymbol{i})\,(\boldsymbol{v}\cdot\boldsymbol{j}) & (\boldsymbol{t}\cdot\boldsymbol{i})\,(\boldsymbol{v}\cdot\boldsymbol{k}) & -\alpha K\,(\boldsymbol{t}\cdot\boldsymbol{j})\,(\boldsymbol{v}\cdot\boldsymbol{n}) & -\alpha K\,(\boldsymbol{t}\cdot\boldsymbol{k})\,(\boldsymbol{v}\cdot\boldsymbol{n})\\
0 & 0 & 0 & (\boldsymbol{u}\cdot\boldsymbol{j}) & (\boldsymbol{u}\cdot\boldsymbol{k})\\
0 & 0 & 0 & (\boldsymbol{v}\cdot\boldsymbol{j}) & (\boldsymbol{v}\cdot\boldsymbol{k})
\end{pmatrix}.
$$

$$
\partial\left(\psi,v',w',v,w\right)/\partial\,(\psi,\phi,\lambda,x_\perp,y_\perp)=
$$

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0\\
0 & \dfrac{\cos\lambda\,(\boldsymbol{v}\cdot\boldsymbol{k})}{(\boldsymbol{t}\cdot\boldsymbol{i})^2} & -\dfrac{(\boldsymbol{u}\cdot\boldsymbol{k})}{(\boldsymbol{t}\cdot\boldsymbol{i})^2} & -\dfrac{\alpha K\,(\boldsymbol{u}\cdot\boldsymbol{i})\,(\boldsymbol{m}\cdot\boldsymbol{k})}{(\boldsymbol{t}\cdot\boldsymbol{i})^3} & -\dfrac{\alpha K\,(\boldsymbol{v}\cdot\boldsymbol{i})\,(\boldsymbol{m}\cdot\boldsymbol{k})}{(\boldsymbol{t}\cdot\boldsymbol{i})^3}\\
0 & -\dfrac{\cos\lambda\,(\boldsymbol{v}\cdot\boldsymbol{j})}{(\boldsymbol{t}\cdot\boldsymbol{i})^2} & \dfrac{(\boldsymbol{u}\cdot\boldsymbol{j})}{(\boldsymbol{t}\cdot\boldsymbol{i})^2} & \dfrac{\alpha K\,(\boldsymbol{u}\cdot\boldsymbol{i})\,(\boldsymbol{m}\cdot\boldsymbol{j})}{(\boldsymbol{t}\cdot\boldsymbol{i})^3} & \dfrac{\alpha K\,(\boldsymbol{v}\cdot\boldsymbol{i})\,(\boldsymbol{m}\cdot\boldsymbol{j})}{(\boldsymbol{t}\cdot\boldsymbol{i})^3}\\
0 & 0 & 0 & \dfrac{(\boldsymbol{v}\cdot\boldsymbol{k})}{(\boldsymbol{t}\cdot\boldsymbol{i})} & -\dfrac{(\boldsymbol{u}\cdot\boldsymbol{k})}{(\boldsymbol{t}\cdot\boldsymbol{i})}\\
0 & 0 & 0 & -\dfrac{(\boldsymbol{v}\cdot\boldsymbol{j})}{(\boldsymbol{t}\cdot\boldsymbol{i})} & \dfrac{(\boldsymbol{u}\cdot\boldsymbol{j})}{(\boldsymbol{t}\cdot\boldsymbol{i})}
\end{pmatrix},
$$

where $\boldsymbol{m} = \boldsymbol{t} \times \boldsymbol{n}$ and

$$(\boldsymbol{m} \cdot \boldsymbol{k}) = (\boldsymbol{v} \cdot \boldsymbol{k})\,(\boldsymbol{n} \cdot \boldsymbol{u}) - (\boldsymbol{u} \cdot \boldsymbol{k})\,(\boldsymbol{n} \cdot \boldsymbol{v})\,,$$
$$(\boldsymbol{m} \cdot \boldsymbol{j}) = (\boldsymbol{v} \cdot \boldsymbol{j})\,(\boldsymbol{n} \cdot \boldsymbol{u}) - (\boldsymbol{u} \cdot \boldsymbol{j})\,(\boldsymbol{n} \cdot \boldsymbol{v})\,.$$

Transformations Between the Intermediate Cartesian Frame and the Local Frame

$$\partial\left(\psi, v', w', v, w\right)/\partial\left(p'_x, p'_y, p'_z, x', y', z'\right) =$$

$$\begin{pmatrix} -qp'_x/g & -qp'_y/g & -qp'_z/g & 0 & 0 & 0 \\ 1/p'_z & 0 & -p'_x/{p'_z}^2 & 0 & 0 & 0 \\ 0 & 1/p'_z & -p'_y/{p'_z}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

with

$$g = \left({p'_x}^2 + {p'_y}^2 + {p'_z}^2\right)^{3/2}.$$

$$\partial\left(p'_x, p'_y, p'_z, x', y', z'\right)/\partial\left(\psi, v', w', v, w\right) =$$

$$\begin{pmatrix} -\dfrac{q \cdot s_z}{\psi^2} \cdot \dfrac{v'}{h} & \dfrac{q \cdot s_z}{\psi} \cdot \dfrac{1 + w'^2}{h^3} & -\dfrac{q \cdot s_z}{\psi} \cdot \dfrac{v'w'}{h^3} & 0 & 0 \\ -\dfrac{q \cdot s_z}{\psi^2} \cdot \dfrac{w'}{h} & -\dfrac{q \cdot s_z}{\psi} \cdot \dfrac{v'w'}{h^3} & \dfrac{q \cdot s_z}{\psi} \cdot \dfrac{1 + v'^2}{h^3} & 0 & 0 \\ -\dfrac{q \cdot s_z}{\psi^2} \cdot \dfrac{1}{h} & -\dfrac{q \cdot s_z}{\psi} \cdot \dfrac{v'}{h^3} & -\dfrac{q \cdot s_z}{\psi} \cdot \dfrac{w'}{h^3} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

with

$$s_z = \operatorname{sign}(p_z),\ h = \sqrt{1 + v'^2 + w'^2}.$$

Appendix B
Regularization of the Kinematic Fit

If $\boldsymbol{V}$ in Eq. (8.61) is not of full rank (singular), it can be regularized by adding a positive multiple of the identity matrix:

$$\boldsymbol{V}^{\delta} = \boldsymbol{V} + \delta \boldsymbol{I} \quad \text{with} \quad \delta > 0.$$

The eigenvalues of $\boldsymbol{V}^{\delta}$ are equal to the non-negative eigenvalues of $\boldsymbol{V}$ plus δ. Therefore, they are all positive and $\boldsymbol{V}^{\delta}$ is regular. The corresponding estimate $\hat{\boldsymbol{\alpha}}^{\delta}$ (see Eq. (8.68)) is equal to:

$$\hat{\boldsymbol{\alpha}}^{\delta} = \boldsymbol{y} - \boldsymbol{V}^{\delta}\boldsymbol{H}^{\mathsf{T}}\boldsymbol{G}_H^{\delta}(\boldsymbol{H}\boldsymbol{y} + \boldsymbol{d}),$$

with

$$\begin{aligned}\boldsymbol{G}_H^{\delta} &= (\boldsymbol{H}\boldsymbol{V}^{\delta}\boldsymbol{H}^{\mathsf{T}})^{-1} = [\boldsymbol{H}(\boldsymbol{V} + \delta\boldsymbol{I})\boldsymbol{H}^{\mathsf{T}}]^{-1} = \\ &= [\boldsymbol{H}\boldsymbol{V}\boldsymbol{H}^{\mathsf{T}} + \delta\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}}]^{-1} = [\boldsymbol{G}_H{}^{-1} + \delta\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}}]^{-1}.\end{aligned}$$

Application of the Woodbury identity [1, Theorem 18.2.8] yields:

$$\begin{aligned}\boldsymbol{G}_H^{\delta} &= \boldsymbol{G}_H - \boldsymbol{G}_H[1/\delta \cdot (\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}})^{-1} + \boldsymbol{G}_H]^{-1}\boldsymbol{G}_H = \\ &= \boldsymbol{G}_H - \delta \cdot \boldsymbol{G}_H[(\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}})^{-1} + \delta\boldsymbol{G}_H]^{-1}\boldsymbol{G}_H.\end{aligned}$$

It follows that

$$\lim_{\delta \to 0^+} \boldsymbol{G}_H^{\delta} = \boldsymbol{G}_H \quad \text{and} \quad \lim_{\delta \to 0^+} \hat{\boldsymbol{\alpha}}^{\delta} = \hat{\boldsymbol{\alpha}}.$$

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0

Reference

1. D. Harville, *Matrix Algebra From a Statistician's Perspective* (Springer Science & Business Media, New York, 2008)

Appendix C
Software

This appendix presents some software platforms and packages that are not tied to a specific experiment and are of particular interest to algorithm developers and users.

FairRoot

FairRoot [1] is an object-oriented framework for simulation, reconstruction and analysis. It contains base classes for easy definition of the user's experimental setup. Using the concept of Virtual Monte Carlo [2] different simulation programs can be deployed without changing the user code or the geometry description. The framework delivers classes for detector handling and magnetic field definition, so that the user can perform simulations with different detector setups and test different reconstruction algorithms. The framework is used by several experiments at FAIR, as well as by some groups outside.

ACTS: A Common Tracking Software

ACTS [3, 4] encapsulates track reconstruction algorithms in a generic, experiment-independent framework. It provides data structures and algorithms for simulation, track finding and track fitting, including track parameter propagation. Special emphasis is put on parallelization and vectorization of the software. The implementation is independent of particular assumptions on the detector technology, the detector configuration and the magnetic field. The further development of the project depends on active input from the tracking community. ACTS is thus an excellent testbed for new ideas on track finding and track fitting.

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0

GBL: General Broken Lines

The general broken lines (GBL) algorithm [5, 6] is a fast track refit based on the concept of linearized regression with breakpoints (Sects. 6.1.3 and 6.1.4). The track model is implemented in such a way that the model matrix has a very special structure and can be inverted in linear time with respect to the number of measurements. The GBL fit computes the full covariance matrix of the corrections to the track parameters along the track and gives the necessary input to track-based alignment with global methods, in particular Millepede II [7].

GENFIT

GENFIT [8–11] is a generic toolkit for track fitting. It provides classes for measurements, track representations and fitting algorithms. There are predefined measurement classes for various detector types, including planar detectors, drift chambers, and time projection chambers. Track representations include code for track parametrization and track extrapolation by Runge–Kutta. The available fitters are the Kalman filter (with or without reference track), the Deterministic Annealing Filter, and the General Broken Lines. There is an interface to the RAVE vertex reconstruction toolbox, see below.

A new and extended version, called GENFIT2, is described in [12]. Studies of its performance in the software frameworks of the experiments Belle II at SuperKEKB and PANDA at FAIR (Chap. 11) can be found in [12].

RAVE

RAVE [13] is an experiment-independent toolkit for vertex reconstruction. It comes with a stand-alone framework VERTIGO for debugging, testing and analyzing. The vertex fitters are the extended Kalman filter (Sect. 8.1.2.2) and the adaptive vertex fitter (Sect. 8.2.2). Secondary vertex finding by an iterated adaptive fitter (AVR, Sect. 7.3.3) is implemented as well. The toolkit can be embedded in the software framework of a collider experiment with minimal effort.

References

1. M. Al-Turany, et al., J. Phys. Conf. Ser. **396**(2), 022001 (2012). https://iopscience.iop.org/article/10.1088/1742-6596/396/2/022001
2. I. Hrivnacova, et al., CoRR **cs.SE/0306005** (2003). http://arxiv.org/abs/cs.SE/0306005

3. C. Gumpert, et al., J. Phys. Conf. Ser. **898**, 042011 (2017). https://iopscience.iop.org/article/10.1088/1742-6596/898/4/042011
4. A. Salzburger, et al. A Common Tracking Software. https://acts.web.cern.ch/ACTS
5. C. Kleinwort, Nucl. Instrum. Meth. Phys. Res. A **673**, 107 (2012)
6. C. Kleinwort, General Broken Lines. http://www.terascale.de/wiki/generalbrokenlines
7. V. Blobel, C. Kleinwort, F. Meier, Comput. Phys. Commun. **182**(9), 1760 (2011)
8. C. Höppner, et al., Nucl. Instrum. Meth. Phys. Res. A **620**(2), 518 (2010)
9. J. Rauch, T. Schlüter, J. Phys. Conf. Ser. **608**(1), 012042 (2015). https://iopscience.iop.org/article/10.1088/1742-6596/608/1/012042
10. J. Rauch, T. Schlüter, GENFIT—a Generic Track-Fitting Toolkit (2016). https://arxiv.org/abs/1410.3698v2
11. GENFIT—a generic toolkit for track reconstruction for experiments in particle and nuclear physics. http://genfit.sourceforge.net
12. T. Bilka, et al., Submitted to Nucl. Instrum. Meth. Phys. Res. A (2019). https://arxiv.org/pdf/1902.04405.pdf
13. W. Waltenberger, et al., J. Phys. Conf. Ser. **119**, 032037 (2008). https://iopscience.iop.org/article/10.1088/1742-6596/119/3/032037

Glossary and Abbreviations

ACTS	A Common Tracking Software
ALICE	A Large Ion Collider Experiment, an experiment at the LHC
ATLAS	A Toroidal LHC Apparatus, an experiment at the LHC
AVF	Adaptive Vertex Fit
Belle II	An experiment at the SuperKEKB collider
CBM	Compressed Baryon Matter, a future experiment at FAIR
CA	Cellular Automaton
CDF	Collider Detector at Fermilab, an experiment at the Tevatron
CDF	Cumulative Distribution Function
CERN	European Laboratory for Particle Physics in Geneva
CKF	Combinatorial Kalman Filter
CMS	Compact Muon Solenoid, an experiment at the LHC
DAF	Deterministic Annealing Filter
FAIR	Facility for Antiproton and Ion Research, an accelerator complex at GSI
GBL	General Broken Lines
GNN	Graph Neural Network
GSF	Gaussian-Sum Filter
GSI	German federal research institute for heavy ion research in Darmstadt
HL-HLC	High-luminosity stage of the LHC, starting in 2026
HLT	High-level trigger
KEK	The Japanese National Laboratory for High-Energy Physics
LHC	Large Hadron Collider, the large proton-proton collider at CERN
LHCb	Large Hadron Collider beauty, an experiment at the LHC
LS	Least Squares

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0

LSTM	Long Short-Term memory
PDF	Probability Density Function
RNN	Recurrent Neural Network
SuperKEKB	Electron-positron collider (B-factory) at KEK

Index

R. Frühwirth, A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection,
https://doi.org/10.1007/978-3-030-65771-0

V