General Broken Lines as advanced track model

Claus Kleinwort, DESY, Germany

November 6, 2013

Abstract

In HEP experiments the description of the trajectory of a charged particle is obtained from a fit to measurements in tracking detectors. The parametrization of the trajectory has to account for bending in the magnetic field, energy loss and multiple scattering in the detector material. General broken lines define a track model with proper description of multiple scattering leading to linear equations with a special structure of the corresponding matrix allowing for a fast solution with the computing time depending linearly on the number of measurements. The calculation of the full covariance matrix along the trajectory enables the application to track based alignment and calibration of large detectors with global methods.

Contents

1 Introduction

The trajectory of a charged particle in a homogeneous magnetic field neglecting the interactions with the detector material is described by a helix. In a global coordinate system (x, y, z) with the magnetic field in z-direction it can be parametrized by the inverse momentum (times charge) q/p, an angle ϕ_0 at and the distance d_0 to the point of closest approach in the (x, y)-plane, the dip angle λ to that plane and the offset z_0 at the point of closest approach. Energy loss in the detector material due to ionization or radiation (for electrons) leads to a reduction of the momentum. *Multiple scattering*, mainly due to Coulomb interaction with the electrons in the atoms, results in random changes in direction and spatial position with expectation values of zero and variance depending on the traversed material and the particle momentum. In addition the magnetic field is usually homogeneous only in an approximation. Therefore more advanced track models are required.

The effect of multiple scattering can be taken into account in different ways [1]. For global methods it can be added to the measurement errors leading to a non diagonal covariance matrix or explicitly fitted by scattering angles as additional track parameters. In both cases this requires the inversion of a large matrix of order n (number of measurements or number of scatterers) with computing time $\mathcal{O}(n^3)$. The progressive Kálmán filter [2] updating the track parameters for each additional measurement and scatterer with a computing time $\mathcal{O}(n)$ has become the standard method for track fitting.

The broken lines model is a fast track refit adding the description of multiple scattering to an initial trajectory and able to determine the covariance matrix of all track parameters. This allows the usage as track model for track based alignment and calibration with the global Millepede-II [3] algorithm. Corrections and covariance matrices for the local track parameters can be determined with a computing time $\mathcal{O}(n)$.

The original formulation [4] describes the case of a tracking detector with a solenoidal magnetic field and independent two-dimensional tracking in the bending and perpendicular to the bending plane. It constructs the planar trajectories from the measurements including the material around those as thick scatterers. In the presence of measurements with components in both planes a single trajectory in three dimensions is required. In the following this generalization and that for an arbitrary magnetic field is described for a track with n_{meas} one- or two- dimensional independent measurements in n_{plane} planes.

The application of a three-dimensional broken lines trajectory for the track-based alignment of the CMS silicon tracker with Millepede-II [3] is described in [5].

2 General broken lines

2.1 Seeding

The general broken lines are seeded by an initial trajectory. This can be the result of a fit of the measurements (*internal seeding*) or a prediction from another part of the detector (*external seeding*). The seeding track parameters at some reference point are used for the propagation along the trajectory according to the magnetic field (and average energy loss) to calculate residuals for the measurements and parameter transformation matrices. If the track fit has to be iterated to account for nonlinear effects the initial trajectory could be based on general broken lines itself.

2.2 Construction

At each measurement plane and each scatterer a local (orthonormal) coordinate system (u_1, u_2, w) is defined.[1] A local system moving with the track is the *curvilinear* frame $(x_\perp, y_\perp, z_\perp)$ with $z_\perp$ in track direction and $x_\perp$ in the global (x, y)-plane. The material between adjacent measurements planes is represented by up to two thin scatterers (zero thickness) with the same mean and variance of the material distribution. A single thin scatterer produces no direct spatial shifts but only a two-dimensional scattering angle with variance $\theta_0^2 \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$ [6]. The trajectory is constructed now from the thin scatterers adding the measurements by interpolation of the enclosing scatterers. At each thin scatterer a two-dimensional offset $\mathbf{u} = (u_1, u_2)$ in the local frame is defined as fit parameter of the track together with a common correction of the inverse momentum $\Delta q/p$. To define the start and end of the trajectory the first and last measurement planes are added to the sequence of scatterers. The number of scatterers n_{scat} is between two (no material at all) and $2\,n_{\text{plane}}$ (thick scatterers between all planes) and the number of track parameters to be fitted is $n_{\text{par}} = 2\,n_{\text{scat}} + 1$. Figure 1 illustrates this construction.

In the local frame a track can be described by the offset $\mathbf{u}$, the slope $\mathbf{u}' = \frac{\partial \mathbf{u}}{\partial w}$ and the inverse momentum q/p. With small distortions $(\Delta q/p, \Delta\mathbf{u}', \Delta\mathbf{u})$ of the local track parameters the offsets propagate like:

$$\Delta\mathbf{u}_{i+1} = \frac{\partial \mathbf{u}_{i+1}}{\partial \mathbf{u}_i}\Delta\mathbf{u}_i + \frac{\partial \mathbf{u}_{i+1}}{\partial \mathbf{u}'_i}\Delta\mathbf{u}'_i + \frac{\partial \mathbf{u}_{i+1}}{\partial q/p}\Delta q/p = \mathbf{J}_i\Delta\mathbf{u}_i + \mathbf{S}_i\Delta\mathbf{u}'_i + \mathbf{d}_i\Delta q/p \quad (1)$$

The derivatives $\mathbf{J}$, $\mathbf{S}$ and $\mathbf{d}$ can be obtained from the curvilinear Jacobian and the transformation of track parameters between the local and the curvilinear frame:

$$\frac{\partial(q/p, \mathbf{u}', \mathbf{u})_{i+1}}{\partial(q/p, \mathbf{u}', \mathbf{u})_i} = \frac{\partial(q/p, \mathbf{u}', \mathbf{u})_{i+1}}{\partial(q/p, \lambda, \phi, x_\perp, y_\perp)_{i+1}} \frac{\partial(q/p, \lambda, \phi, x_\perp, y_\perp)_{i+1}}{\partial(q/p, \lambda, \phi, x_\perp, y_\perp)_i} \frac{\partial(q/p, \lambda, \phi, x_\perp, y_\perp)_i}{\partial(q/p, \mathbf{u}', \mathbf{u})_i} \quad (2)$$

[1] A natural choice for the w-direction is perpendicular to the measurement plane for a measurement and parallel to the track direction for a scatterer.

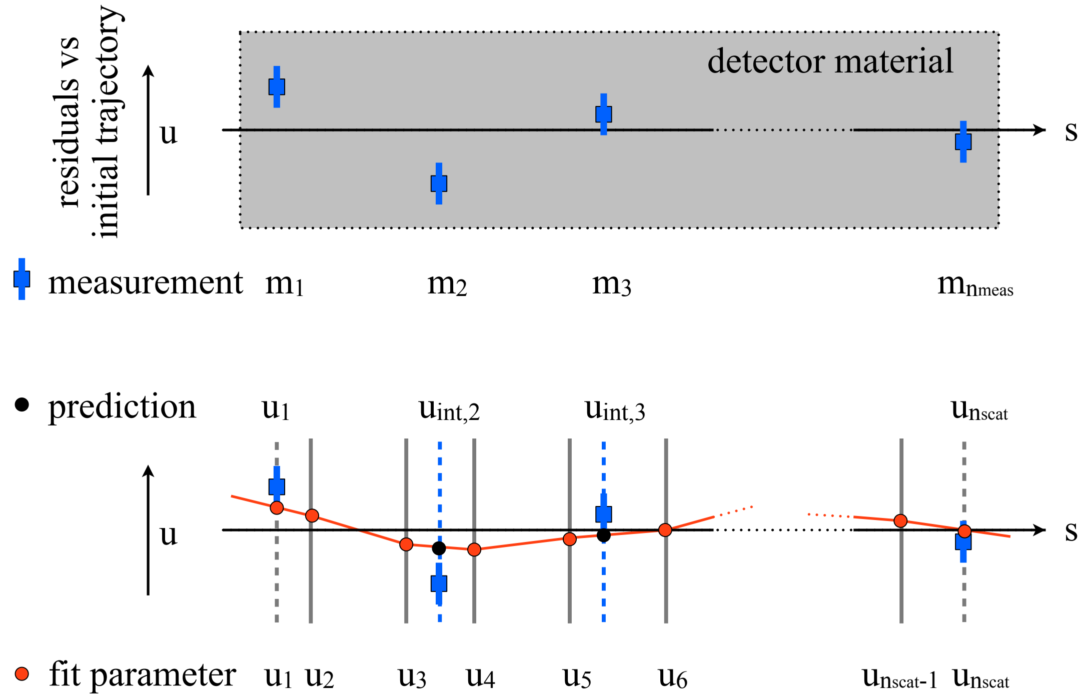

Figure 1: Simple example in one plane, no magnetic field, measurements m_i in planes perpendicular to the track direction and homogeneous material distribution. *Top:* Residuals versus initial trajectory along arc-length s. *Bottom:* Broken lines trajectory based on thin scatterers with offsets u. The material between measurement i and $i+1$ is described with two thin scatterers at $s = (s_{i+1}+s_i)/2 \pm (s_{i+1}-s_i)/\sqrt{12}$. First and last measurement define additional offsets. The fit prediction $u_{\text{int},i}$ for the measurement m_i is obtained by interpolation between the enclosing scatterers: $u_{\text{int},i} = f(u_{2i-1}, u_{2i})$, $(u_{\text{int},1} = u_1,\ u_{\text{int},n_{\text{meas}}} = u_{n_{\text{scat}}})$.

The case of a constant magnetic field and momentum is described in [7]. Solving for the slope correction yields:

$$\Delta \mathbf{u}'_i = \mathbf{S}_i^{-1}(\Delta \mathbf{u}_{i+1} - \mathbf{J}_i \Delta \mathbf{u}_i - \mathbf{d}_i \Delta q/p) \tag{3}$$

With a triplet of three adjacent offsets $(\mathbf{u}_-, \mathbf{u}_0, \mathbf{u}_+)$ two slopes can be determined at $\mathbf{u}_0$:

$$\mathbf{u}'_{0(+)} = \mathbf{W}_+(\mathbf{u}_+ - \mathbf{J}_+\mathbf{u}_0 - \mathbf{d}_+\Delta q/p), \mathbf{W}_+ = \mathbf{S}_+^{-1} \tag{4a}$$

$$\mathbf{u}'_{0(-)} = \mathbf{W}_-(\mathbf{J}_-\mathbf{u}_0 - \mathbf{u}_- + \mathbf{d}_-\Delta q/p), \mathbf{W}_- = -\mathbf{S}_-^{-1} \tag{4b}$$

The kink $\mathbf{k} = \mathbf{u}'_{0(+)} - \mathbf{u}'_{0(-)}$ measures the change of slope due to multiple scattering at the central scatterer $(\mathbf{u}_0)$:

$$\mathbf{k} = \mathbf{W}_+\mathbf{u}_+ - (\mathbf{W}_+\mathbf{J}_+ + \mathbf{W}_-\mathbf{J}_-)\mathbf{u}_0 + \mathbf{W}_-\mathbf{u}_- - (\mathbf{W}_+\mathbf{d}_+ + \mathbf{W}_-\mathbf{d}_-)\Delta q/p \tag{5}$$

In case of a measurement instead of a scatterer at $\mathbf{u}_0$ there is no scattering and solving the previous equation for $\mathbf{u}_{\text{int}} = \mathbf{u}_0$ with $\mathbf{k} \equiv \mathbf{0}$ using $\mathbf{N} = (\mathbf{W}_+\mathbf{J}_+ + \mathbf{W}_-\mathbf{J}_-)^{-1}$ results in the interpolation equation:

$$\mathbf{u}_{\text{int}} = \mathbf{N}(\mathbf{W}_+\mathbf{u}_+ + \mathbf{W}_-\mathbf{u}_-) - \mathbf{N}(\mathbf{W}_+\mathbf{d}_+ + \mathbf{W}_-\mathbf{d}_-)\Delta q/p \tag{6}$$

Inserting $\mathbf{u}_{\text{int}}$ for $\mathbf{u}_0$ into equation (4) yields:

$$\begin{aligned}\mathbf{u}'_{\text{int}} = {} & \mathbf{W}_-\mathbf{J}_-\mathbf{N}\mathbf{W}_+\mathbf{u}_+ + \mathbf{W}_+\mathbf{J}_+\mathbf{N}\mathbf{W}_-\mathbf{u}_- \\ & - (\mathbf{W}_-\mathbf{J}_-\mathbf{N}\mathbf{W}_+\mathbf{d}_+ + \mathbf{W}_+\mathbf{J}_+\mathbf{N}\mathbf{W}_-\mathbf{d}_-)\Delta q/p\end{aligned} \tag{7}$$

The track parameter variations in the local frame are $\Delta\mathbf{p}_{\text{loc}} = (\Delta q/p, \mathbf{u}', \mathbf{u})$ for a scatterer or $\Delta\mathbf{p}_{\text{loc}} = (\Delta q/p, \mathbf{u}'_{\text{int}}, \mathbf{u}_{\text{int}})$ for a measurement.

2.3 Parameter estimation

The fit parameters $\mathbf{x} = (\Delta q/p, \mathbf{u}_1, .., \mathbf{u}_{n_{scat}})$ have to describe measurements, kinks from multiple scattering and in case of external seeding the track parameters at the reference point (scatterer or measurement). Each of those have an expectation, a variance and a prediction depending linear on the fit parameters.

Measurement Using $\mathbf{P}\mathbf{u}_{\text{int}} = \mathbf{H}_m\mathbf{x}$ the expectation is the residuum $\mathbf{m}$ to the initial trajectory: $\langle \mathbf{H}_m\mathbf{x} \rangle = \mathbf{m}$. The projection matrix into the measurement system is $\mathbf{P} = \frac{\partial \mathbf{m}}{\partial \mathbf{u}}$, where the covariance matrix $\mathbf{V}_m$ is diagonal.

Kink In the case of an initial trajectory based on general broken lines there can be an initial non-zero kink $\mathbf{k}_0$ and with $\mathbf{k} = \mathbf{H}_k\mathbf{x}$ the expectation is: $\langle \mathbf{H}_k\mathbf{x} + \mathbf{k}_0 \rangle = \mathbf{0}$. For small kinks the variance $\mathbf{V}_k$ can be calculated by parameter transformation [7] from the curvilinear to the local slopes which depends on the projections $c_i = \mathbf{e}_{\text{track}} \cdot \mathbf{e}_{u_i}$ of the offset directions $\mathbf{e}_{u_i}$ of the local frame onto the track direction $\mathbf{e}_{\text{track}}$ and the variance of the multiple scattering angle:

$$\mathbf{V}_k = \frac{\partial(u'_1, u'_2)}{\partial(x'_\perp, y'_\perp)} \begin{pmatrix} \theta_0^2 & 0 \\ 0 & \theta_0^2 \end{pmatrix} \left[\frac{\partial(u'_1, u'_2)}{\partial(x'_\perp, y'_\perp)}\right]^T = \frac{\theta_0^2}{(1 - c_1^2 - c_2^2)^2} \begin{pmatrix} 1 - c_2^2 & c_1 c_2 \\ c_1 c_2 & 1 - c_1^2 \end{pmatrix} \tag{8}$$

With a least one offset defined perpendicular to the track direction this is a diagonal matrix.

External seed The trajectory used is based on external track parameters with variance $\mathbf{V}_s$ in the local frame at the reference point. On average no change of the track parameters at that point is expected: $\langle \Delta\mathbf{p}_{\text{loc}} \rangle = \langle \mathbf{H}_s\mathbf{x} \rangle = \mathbf{0}$.

The fit parameters are determined by minimizing:

$$\begin{aligned}\chi^2(\mathbf{x}) = \sum_{i=1}^{n_{\text{meas}}} (\mathbf{H}_{m,i}\mathbf{x} - \mathbf{m}_i)^T \mathbf{V}_{m,i}^{-1} (\mathbf{H}_{m,i}\mathbf{x} - \mathbf{m}_i) \quad &\text{(from measurements)} \\ + \sum_{i=2}^{n_{\text{scat}}-1} (\mathbf{H}_{k,i}\mathbf{x} + \mathbf{k}_{0,i})^T \mathbf{V}_{k,i}^{-1} (\mathbf{H}_{k,i}\mathbf{x} + \mathbf{k}_{0,i}) \quad &\text{(from kinks)} \\ + (\mathbf{H}_s\mathbf{x})^T \mathbf{V}_{\mathbf{s}}^{\mathbf{-1}} (\mathbf{H}_s\mathbf{x}) \quad &\text{(from external seed)}\end{aligned} \tag{9}$$

The minimization leads to a linear equation system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{x}$ of size $n = n_{\text{par}}$:

$$\mathbf{A} = \sum_{i=1}^{n_{\text{meas}}} \mathbf{H}_{m,i}^T \mathbf{V}_{m,i}^{-1} \mathbf{H}_{m,i} + \sum_{i=2}^{n_{\text{scat}}-1} \mathbf{H}_{k,i}^T \mathbf{V}_{k,i}^{-1} \mathbf{H}_{k,i} + \mathbf{H}_s^T \mathbf{V}_s^{-1} \mathbf{H}_s \tag{10a}$$

$$\mathbf{b} = \sum_{i=1}^{n_{\text{meas}}} \mathbf{H}_{m,i}^T \mathbf{V}_{m,i}^{-1} \mathbf{m}_i - \sum_{i=2}^{n_{\text{scat}}-1} \mathbf{H}_{k,i}^T \mathbf{V}_{k,i}^{-1} \mathbf{k}_{0,i} \tag{10b}$$

As all $\mathbf{u}_{\text{int},i}$ and $\Delta\mathbf{p}_{\text{loc}}$ depend only on two and all $\mathbf{k}_i$ on three adjacent offsets (and all on $\Delta q/p$), the offset part of all contributions ($\mathbf{H}^T\mathbf{V}^{-1}\mathbf{H}$) to $\mathbf{A}$ are block diagonal matrices with a single non-zero quadratic block of size $n_b \leq 3 \cdot \text{dimension}(\mathbf{u}) = 6$ and the sum $\mathbf{A}$ is a bordered band matrix with border size $b = 1$ (q/p part) and band width $m = \max(n_b) - 1 = 5$ (offset part): $\mathbf{A}_{kl} = 0$ for $\min(k, l) > b$ and $\text{abs}(k - l) > m$. This special structure of the matrix allows the usage of the root-free Cholesky decomposition $\mathbf{A}_u = \mathbf{L}\mathbf{D}\mathbf{L}^T$ of the band part of the matrix into a diagonal matrix $\mathbf{D}$ and a left unit triangular band matrix $\mathbf{L}$. The solution $\mathbf{x}$ is calculated with $\mathcal{O}(n(m + b)^2)$ and the full covariance matrix $\mathbf{V}_x = \mathbf{A}^{-1}$ with $\mathcal{O}(n^2(m+b))$ operations in comparison with simple inversion requiring $\mathcal{O}(n^3)$. The matrix $\mathbf{V}_x$ is required by the global alignment and calibration algorithm Millepede-II [3]. For a sparse matrix the elements of the inverse matrix inside the sparsity pattern can by calculated by special methods without those outside that pattern [8]. This allows to obtain the bordered band part of $\mathbf{V}_x$ with $\mathcal{O}(n(m+b)^2)$ operations. The elements of that part are sufficient to calculate at each measurement plane or scatterer the covariance matrix of the local track parameters $\Delta\mathbf{p}_{\text{loc}}$ and the error of the prediction $(\mathbf{Hx})$ of the track model enabling the determination of the *pulls* (residuals normalized to error from measurement and prediction) for the measurements and kinks.

In case the resulting track parameters are not small corrections as assumed in equation (1) the trajectory has to be iterated.

2.4 Comparision with Kálmán filter

Track fitting with the Kálmán filter algorithm [2] is a sequential procedure adding measurements and scatterers (process noise) to the trajectory one at the time. The addition of each measurement requires the inversion of several 5×5 matrices ($\mathbf{C}$ in equation (12)). For n measurements about $\mathcal{O}(n) \cdot 5^3$ operations are performed. The general broken lines can add several (n, between one and all) measurements with one fit using $\mathcal{O}(n_{par}) \cdot (5+1)^2$ operations with $n_{par} = 5..(4 \cdot n + 1)$ depending on the number of thin scatterers. In a toy setup the performance for track refitting has been compared. Tracks in a detector consisting of N planes with two independent measurements and thin scatterers coinciding with the measurement planes (simple model for a silicon tracker) or one or two thin scatterers between adjacent planes have been simulated. In all cases tested ($N = 10, 30, 50$) the refitting with general broken lines is about a factor 3–4 faster than with a Kálmán filter (filtering and smoothing).

In general only the covariance matrices for the track parameters at single points are calculated by the Kálmán filter. For the global covariance matrix from all track parameters of the trajectory required by global alignment and calibrations methods an extension of the Kálmán filter is described in [9].

An externally seeded general broken lines trajectory with one measurement has only one offset defined and the slope has to be used directly as fit parameter: $\mathbf{x} = \Delta\mathbf{p}_{\text{loc},1} = (\Delta q/p, \mathbf{u}'_1, \mathbf{u}_1)$. The solution of the linear equation system is:

$$\mathbf{x} = \mathbf{A}^{-1}\left[\mathbf{H}_{m,1}^T \mathbf{V}_{m,1}^{-1} \mathbf{m}_1\right], \; \mathbf{A}^{-1} = \left[\mathbf{V}_s^{-1} + \mathbf{H}_{m,1}^T \mathbf{V}_{m,1}^{-1} \mathbf{H}_{m,1}\right]^{-1} \qquad (11)$$

This is equivalent to the filtering step of a Kálmán filter in the weighted mean formalism (equation (8b) in [2]):

$$x_k = \mathbf{C}_k \left[\left(\mathbf{C}_k^{k-1}\right)^{-1} x_k^{k-1} + \mathbf{H}_k^t \mathbf{V}_k^{-1} m_k\right], \; \mathbf{C}_k = \left[\left(\mathbf{C}_k^{k-1}\right)^{-1} + \mathbf{H}_k^t \mathbf{V}_k^{-1} \mathbf{H}_k\right]^{-1} \qquad (12)$$

As the initial trajectory has been based on the track parameters from the external seed the prediction x_k^{k-1} is zero, $\mathbf{C}_k$ corresponds to $\mathbf{A}^{-1}$, $\mathbf{C}_k^{k-1}$ to $\mathbf{V}_s$ and m_k to $\mathbf{m}_1$.

3 Manual

This section describes version V01-16-01 of the C++, Python and FORTRAN implementations of general broken lines provided by the statistics tools group of the Helmholtz Alliance 'Physics at the Terascale' [10]. Detailed descriptions are available in the corresponding *doxygen* documentation.

3.1 Overview

The general broken lines are constructed from a sequence of points ordered by arc-length on an initial trajectory provided for example by an external seed, a fit with a simplified track model or the track finding. Each point may contain a measurement or (thin) scatterer or both and can be queried after the fit for the local track parameter corrections and covariance matrix. One point can be the reference point for an external track seed. The user has to define at each point a local coordinate system and to provide information about the measurements and scatterers in and the jacobians (transformation matrices for track parameter variations) between those systems.

3.2 Special features

Additional parameters. *Local* ($\mathbf{x}_l$) or *global* ($\mathbf{x}_g$) parameters can be defined for the track fit in addition to the usual track parameters. The local parameters are fitted for each track, the global ones can be used to implement calibration or alignment parameters to be determined by Millepede-II.

For example in a wire chamber the drift distance $d \approx (t_d - t_0) \cdot v_d$ is a function of the measured drift time t_d, the time t_0 the particle passed through the detector and the drift velocity v_d. For each track a Δt_0 could be fitted as additional local track parameter to check for example the consistency with the expected value (from the event or the accelerator). A calibration parameter correction Δv_d could be determined as global parameter from a larger set of tracks.

Diagonalization (C++, Python). Non-diagonal covariance matrices for measurements $\mathbf{V}_m$ and kinks $\mathbf{V}_k$ are diagonalized internally.

Interface to Millepede-II [3]. The required binary data files can be produced directly from the trajectories. As Millepede-II allows only for independent scalar measurements the covariance matrices for measurements $\mathbf{V}_m$ and kinks $\mathbf{V}_k$ must be diagonal (FORTRAN).

Robust estimators. For outlier down-weighting the method of M-estimators can be used. In this maximum likelihood ansatz not the (sum of) *squares* of normalized residuals z are minimized, but iteratively a function $\rho(z)$ giving less weight $\omega(z) = \frac{\partial\rho(z)}{\partial z}/z$ to large values of the residuals. Implemented are the Cauchy, Huber and Tukey ('bisquare') function (with asymptotic effciency[2] of 95% on a unit normal distribution $N(z)$) [11]:

	χ^2 ($\rho(z) = \frac{1}{2}z^2$)	Cauchy	Huber	Tukey
$\omega(z)$ for $\lvert z\rvert \le c$	1	$[1+(z/c)^2]^{-1}$	1	$[1-(z/c)^2]^2$
$\omega(z)$ for $\lvert z\rvert > c$	1	$[1+(z/c)^2]^{-1}$	$c/\lvert z\rvert$	0
$2\int_{-\infty}^{+\infty}\rho(z)N(z)dz$	1	0.8228	0.9326	0.8737
c		2.3849	1.3450	4.6851

3.3 C++ implementation

Matrices are implemented with ROOT [12]. User input or output is in the form of TMatrices. Internally SMatrices are used for fixes sized and simple matrices based on std::vector<> for variable sized matrices. Besides simple trajectories describing the path of a single particle composed trajectories are supported. These are constructed from the trajectories of multiple particles and some external parameters (like those describing a decay) and the transformations at the first points from the external to the local track parameters.

Listed below is the calling sequence, calls to '()' are mandatory, to '[]' optionally.

(1) Create list of points on initial trajectory:

```
std::vector<GblPoint> list
```

(2) For all points on initial trajectory:

(2.1) Create point and add appropriate attributes:

(2.1.1) Create point (with jacobian from previous point):

```
point = gbl::GblPoint(jacobian)
```

[2.1.2] Add measurement:

```
point.addMeasurement(..)
```

[2.1.2.1] Add additional local parameters:

```
point.addLocals(..)
```

[2.1.2.1] Add additional global parameters:

```
point.addGlobals(..)
```

[2] Minimum possible variance for an unbiased estimator divided by its actual variance.

[2.1.3] Add scatterer:

```
point.addScatterer(..)
```

(2.2) Add point (ordered by arc length) to list:

```
list.push_back(point)
```

(3) Create (simple) trajectory from list of points:

```
traj = gbl::GblTrajectory(list)
```

Optionally with external seed:

```
traj = gbl::GblTrajectory(list,seed)
```

[4] Check validity of trajectory:

```
if (not traj.isValid()) .. //abort
```

(5) Fit trajectory, return error code, get Chi2, Ndf (and weight lost by M-estimators):

```
ierr = traj.fit(..)
```

[6] For any point on initial trajectory:

[6.1] Get corrections and covariance matrix for track parameters:

```
[..] = traj.getResults(label)
```

[6.2] Get residuals for measurement:

```
[..] = traj.getMeasResults(label)
```

[6.3] Get (kink) residuals for scatterer.:

```
[..] = traj.getScatResults(label)
```

[7] Write trajectory to MP binary file:

```
traj.milleOut(..)
```

3.4 Python implementation

Requires at least Python version 2.5 and the 'numpy' module. Implements a module 'gblpy'.

Listed below is the calling sequence, calls to '()' are mandatory, to '[]' optionally.

(1) Create trajectory:

```
traj = GblTrajectory()
```

(2) For all points on initial trajectory:

(2.1) Create point and add appropriate attributes:

(2.1.1) Create point (with jacobian from previous point):

```
point = GblPoint(jacobian)
```

[2.1.2] Add measurement:

```
point.addMeasurement(..)
```

[2.1.2.1] Add additional local parameters:

```
point.addLocals(..)
```

[2.1.2.1] Add additional global parameters:

```
point.addGlobals(..)
```

[2.1.3] Add scatterer:

```
point.addScatterer(..)
```

(2.2) Add point (ordered by arc length) to trajectory, get label of point:

```
label = traj.addPoint(point)
```

[3] Add external seed:

```
traj.addExternalSeed(..)
```

(4) Fit trajectory, return error code, get Chi2, Ndf (and weight lost by M-estimators):

```
[..] = traj.fit()
```

[5] For any point on initial trajectory:

[5.1] Get corrections and covariance matrix for track parameters:

```
[..] = traj.getResults(label)
```

[6] Write trajectory to MP binary file:

```
traj.milleOut(..)
```

3.5 Fortran implementation

With n_{add} additional local parameters at each point $\texttt{NP} = 5 + n_{add}$ track parameter corrections $(\Delta\mathbf{p}_{\text{loc}}, \Delta\mathbf{x}_l)$ are determined. The corresponding covariance matrix is used in symmetric storage mode. Skipping the lower triangular part of the symmetric matrix it is stored in a vector of size $\texttt{NP2} = (\texttt{NP} + 1) \cdot \texttt{NP}/2$ with element (i, j) $(i \geq j)$ at position $i \cdot (i - 1)/2 + j$.

Listed below is the calling sequence, calls to ’()’ are mandatory, to ’[]’ optionally.

(1) Initialize fit.

```
CALL GBLINI(LPRINT)
```

`LPRINT`: Print level, 0 = none, 1 = minimal, 2 = more

In case the two offset directions (u_1, u_2) are not correlated by any measurement or jacobian at any point the fit can be reduced to a projection into one of the (u_i, w) planes:

```
CALL GBLINP(LPRINT,ICOORD)
```

`LPRINT`: Print level, 0 = none, 1 = minimal, 2 = more

`ICOORD`: Coordinate for projection (1 or 2)

(2) Add point. Points have to be added in correct order (increasing or decreasing arc-length).

```
CALL GBLADP(IRET)
```

`IRET`: Returned. Label (=number) of point added (>0) or error (=0).

[3] Add measurement to current point.

```
CALL GBLADM(PROJ,RES,PREC)
```

`PROJ(2,2)` : Projection matrix $\mathbf{P} = \frac{\partial \mathbf{m}}{\partial \mathbf{u}}$ of measurement directions into local system (double precision)

`RES(2)` : Residuals $\mathbf{m}$ to initial trajectory

`PREC(2)` : Diagonal of inverse covariance matrix $\mathbf{V}_m^{-1}$

[3a] Add local derivatives to current measurement.

```
CALL GBLADL(NDER,DER,IRET)
```

`NDER` : Number of local derivatives

`DER(2,*)` : Local derivatives $\partial \mathbf{m}/\partial \mathbf{x}_l$

`IRET` : Returned. Number of non zero derivatives added

[3a] Add global derivatives to current measurement.

```
CALL GBLADG(NDER,LDER,DER,IRET)
```

`NDER` : Number of global derivatives

`LDER(*)` : Labels for global derivatives

`DER(2,*)` : Global derivatives $\partial \mathbf{m}/\partial \mathbf{x}_g$

`IRET` : Returned. Number of non zero derivatives added

[4] Add (thin) scatterer to current point.

```
CALL GBLADS(RES,PREC)
```

`RES(2)` : Values for initial kinks $\mathbf{k}_0$ (non-zero in case of iterating)

`PREC(2)` : Diagonal of inverse covariance matrix of kinks $\mathbf{V}_k^{-1}$

[5] Dump trajectory definition.

```
CALL GBLDMP
```

[6] Add (inverse covariance matrix from) external seed to reference point.

```
CALL GBLADX(IPOINT,DPRC)
```

`IPOINT` : Reference point (signed, <0: side towards previous point, >0: side towards next point)

`DPRC(NP2)` : Precision matrix $\mathbf{V}_s^{-1}$ from external seed (symmetric storage mode, double precision)

(7) For all points inquire and provide jacobians.

```
CALL GBLQRJ(IPOINT,IPREV,INEXT)
```

`IPOINT`**:** Point to query for jacobians

`IPREV` **:** Returned. Previous point, jacobian from IPOINT to IPREV needed

`INEXT` **:** Returned. Next point, jacobian from IPOINT to INEXT needed (or -1 for illegal IPOINT)

```
CALL GBLADJ(IPOINT,AJACP,AJACN)
```

`IPOINT` : Point to add jacobians to

`AJACP(5,5)` : Jacobian to previous point

`AJACN(5,5)` : Jacobian to next point (double precision)

[8] Write Millepede-II record (to FORTRAN unit 51).

```
CALL GBLMP2(IRET)
```

`IRET:` Returned. Number of MillePede measurements in record.

(9) Perform fit.

```
CALL GBLFIT(CDW,MRANK,NP,NDF,CHI2,WLS)
```

`CDW` : String defining iterations for outlier down-weighting, one character per iteration (C: Cauchy, H: Huber, T: Tukey)

`MRANK:` Returned. Rank of trajectory (5: curved track in space, 4: straight track in space, 3: curved track in a plane, 2: straight track in a plane else: fit failed)

`NP` : Returned. Number of track parameters at each point

`NDF` : Returned. Number of degrees of freedom

`CHI2` : Returned. χ^2 value at minimum (divided by $2\int_{-\infty}^{+\infty}\rho(z)N(z)dz$)

`WLS` : Returned. Weight lost by down-weighting ($\sum_i(1-\omega(z_i))$ over all measurements and kinks)

[10] Get track parameter corrections with covariance matrix at point.

```
CALL GBLRES(IPOINT,DPAR,DCOV)
```

`IPOINT` : Point (signed, <0: side towards previous point, >0: side towards next point)

`DPAR(NP)` : Returned. Corrections (double precision)

`DCOV(NP2)` : Returned. Covariance matrix (symmetric storage mode, double precision)

3.6 Fortran example

The example main program `GBLTST` demonstrates the usage of this implementation for a simplified setup. The detector consists out of $N = 10$ measurement planes with single thin scatterers between consecutive planes in a constant magnetic field $(0, 0, B_z)$. For the local coordinate system and track parameters different choices are available:

1. Curvilinear track parameters $(q/p, \lambda, \phi, x_\perp, y_\perp)$ using the track angles instead of local slopes.

2. Slopes and offsets $(q/p, x'_\perp, y'_\perp, x_\perp, y_\perp)$ in the curvilinear frame.

3. Slopes and offsets $(q/p, \mathbf{u}', \mathbf{u})$ in the frame with the u_1 direction in the bending plane and perpendicular to the track and u_2 in the direction of the magnetic field.

The jacobians are approximated for the limit $q/p \to 0$. In this case they are simple quadratic functions of the arc-length difference $\Delta s = s_{i+1} - s_i$ between two points, for example:

$$\frac{\partial(q/p, x'_\perp, y'_\perp, x_\perp, y_\perp)_{i+1}}{\partial(q/p, x'_\perp, y'_\perp, x_\perp, y_\perp)_i} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -B_z\Delta s\cos\lambda & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -B_z\frac{1}{2}\Delta s^2\cos\lambda & \Delta s & 0 & 1 & 0 \\ 0 & 0 & \Delta s & 0 & 1 \end{pmatrix} \tag{13}$$

4 Summary

A trajectory based on General Broken Lines is a track refit to add the description of multiple scattering to an initial trajectory based on the propagation in a magnetic field (and average energy loss). The initial trajectory can be the result of a fit of the measurements (internal seed) or a prediction from another detector part (external seed). It is constructed from a sequence of (pairs of) thin scatterers describing the multiple scattering in the material between adjacent measurement planes. Predictions for the measurements are obtained by interpolation between the enclosing scatterers and triplets of adjacent scatterers define kink angles with variance according to the material of the central scatterer. The required propagation (on the initial trajectory) from a measurement plane or scatterer to the previous and next scatterer is using locally a linearized track model. The matrix of the resulting linear equations system is a bordered band matrix allowing for a fast solution by a root-free Cholesky decomposition in a time proportional to the number of measurements. At all scatterers and measurement planes corrections for the local track parameters are determined. The corresponding covariance matrices and the pulls for the measurements and kinks require only the calculation of the bordered band part of the inverse matrix. For a simulated setup the track refitting is a factor 3–4 faster than with Kálmán filtering and smoothing.

A General Broken Lines fit with external seed and one (additional) measurement is equivalent to the filtering step of the track fit with a Kálmán filter.

As the track refit can provide the complete covariance matrix of all track parameters General Broken Lines are well suited as track model for alignment and calibration with Millepede-II [3].

References

[1] A. Strandlie, R. Frühwirth, Track and vertex reconstruction: From classical to adaptive methods, *Reviews of Modern Physics*, 82:1419, 2010.

[2] R. Frühwirth, Application of Kalman filtering to track and vertex fitting, *Nucl. Instr. and Methods A*, 262:444, 1987.

[3] V. Blobel, Software Alignment for Tracking Detectors, *Nucl. Instr. and Methods A*, 566:5, 2006.

[4] V. Blobel, A new fast track-fit algorithm based on broken lines, *Nucl. Instr. and Methods A*, 566:14, 2006.

[5] V. Blobel, C. Kleinwort, F. Meier, Fast alignment of a complex tracking detector using advanced track models, *Computer Physics Communications* (2011), doi:10.1016/j.cpc.2011.03.017

[6] C. Amsler et al. (Particle Data Group), *Physics Letters B667, 1 (2008) and 2009 partial update for the 2010 ed.*

[7] A. Strandlie, W. Wittek, Derivation of Jacobians for the propagation of the covariance matrices of track parameters in homogeneous magnetic fields, *Nucl. Instr. and Methods A*, 566:687, 2006.

[8] I.S. Duff, A.M. Erisman and J.K. Reid, Direct Methods for Sparse Matrices, *Oxford Science Publications*, 1986, Oxford.

[9] W.D. Hulsbergen, The global covariance matrix of tracks fitted with a Kálmán filter and an application in detector alignment, *Nucl. Instr. and Methods A*, 600:471, 2009.

[10] `https://www.wiki.terascale.de/index.php/GeneralBrokenLines`

[11] W. Rey, Introduction to Robust and Quasi-Robust Statistical Methods, *Springer-Verlag*, 1983, ISBN 3-540-12866-2.

[12] `root.cern.ch`