DESY Seminar 12th December 2005

A new fast track-fit algorithm based on broken lines

Volker Blobel – Universität Hamburg

Abstract

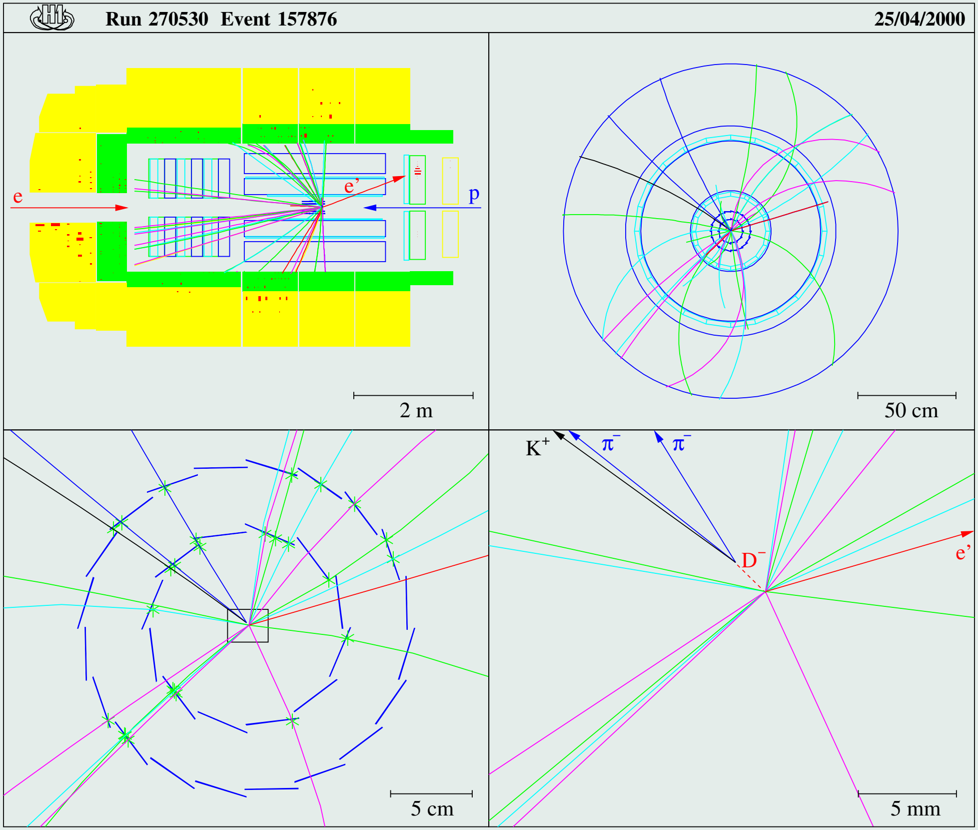
The determination of the particle momentum in HEP tracking chambers requires a fit of a parametrization to the measured points. Various effects can result in deviations to the ideal helix curve in the magnetic field of a solenoid, and the fit with a pure helix parametrization is not optimal. One effect is *multiple scattering*, which causes a "random walk" of the particle and affects especially low-momentum tracks and very accurate measurements.

The method based on broken lines is non-recursive and allows to reconstruct the particle trajectory taking into account details of the multiple scattering. It provides optimal parameters and their covariance matrices at track start and end, and optimal values at each measured point along the trajectory including the variances. Allowing sparse-matrix techniques the method has a execution time proportional to the number of track hits and is, under test conditions, a factor ten faster than the Kalman filter.

Combined with robust estimation techniques like Least-median-of-squares and M-estimates the algorithm can be used effectively during track recognition, where its speed allows a fast rejection of outliers.

---

1. Track measurement in particle physics experiments



Charm event in the H1 detector at HERA with drift chamber and vertex detector.

Track parametrization

Ideal parametrization: a helix (κ, d_{ca}, ϕ, z_0, $\tan\lambda$) in a homogeneous magnetic field $\boldsymbol{B}_z$ with

***xy*- or *rϕ*-plane:** circle, residual ε_i of a measured point (x_i, y_i) from a particle trajectory

$$\frac{1}{2}\kappa\left(x_i^2 + y_i^2 + d_{\mathrm{ca}}^2\right) - \left(1 + \kappa d_{\mathrm{ca}}\right)\left(x_i \sin\phi - y_i \cos\phi\right) + d_{\mathrm{ca}} = 0 \; .$$

***sz*-plane:** straight line, residual ε_i of a measured point (s_i, z_i) from

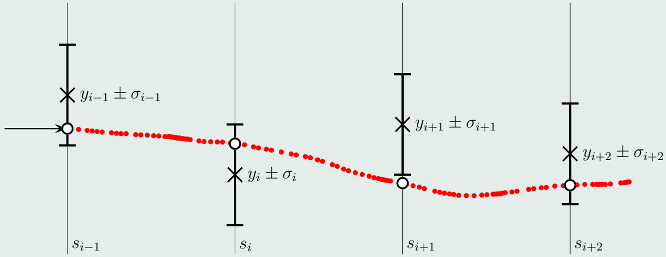
$$z_0 + (\tan\lambda) \cdot s_i - z_i = 0 \qquad \text{with } s_i = \text{track length in } xy$$

. . . but there are random and non-random perturbations:

Multiple scattering (msc): Elastic scattering of charged particles in the Coulomb field of the nuclei in the detector material. The mean of the projected deflection angle θ after traversing a material layer is zero and the distribution has a variance of

$$\text{variance} \quad V[\theta] \propto \frac{t}{\beta^2 p^2} \qquad \text{where} \quad t = \Delta s / X_0$$

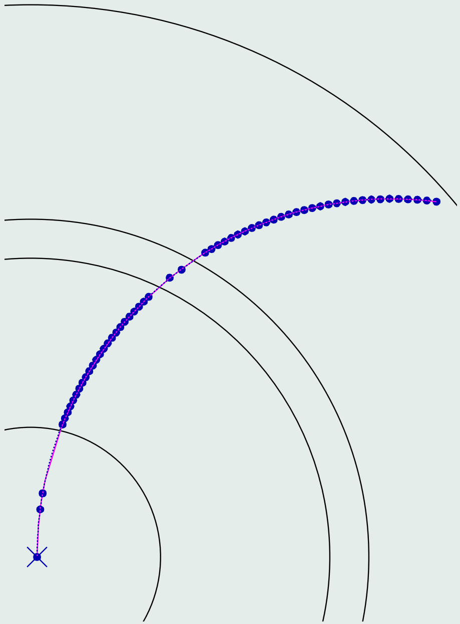
Multiple scattering deflections will influence all downstream measurements in a correlated way, and delimits the momentum measurement precision at low momenta.



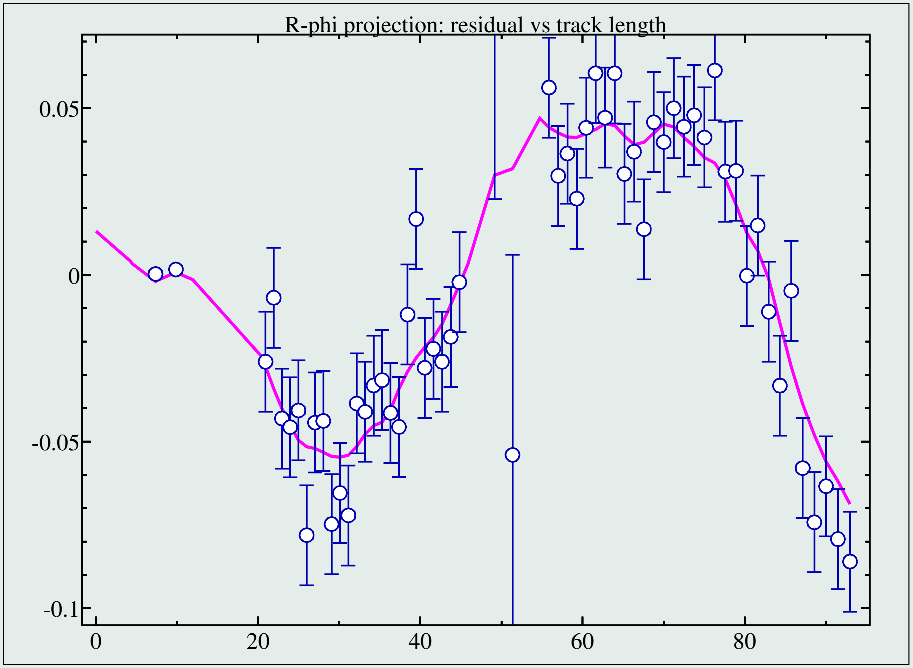
Energy loss, Radiation of electrons, Field inhomogeneity, . . .

Track example with msc *(magenta line is the true particle trajectory)*

Circle fit



Deviations from circle fit



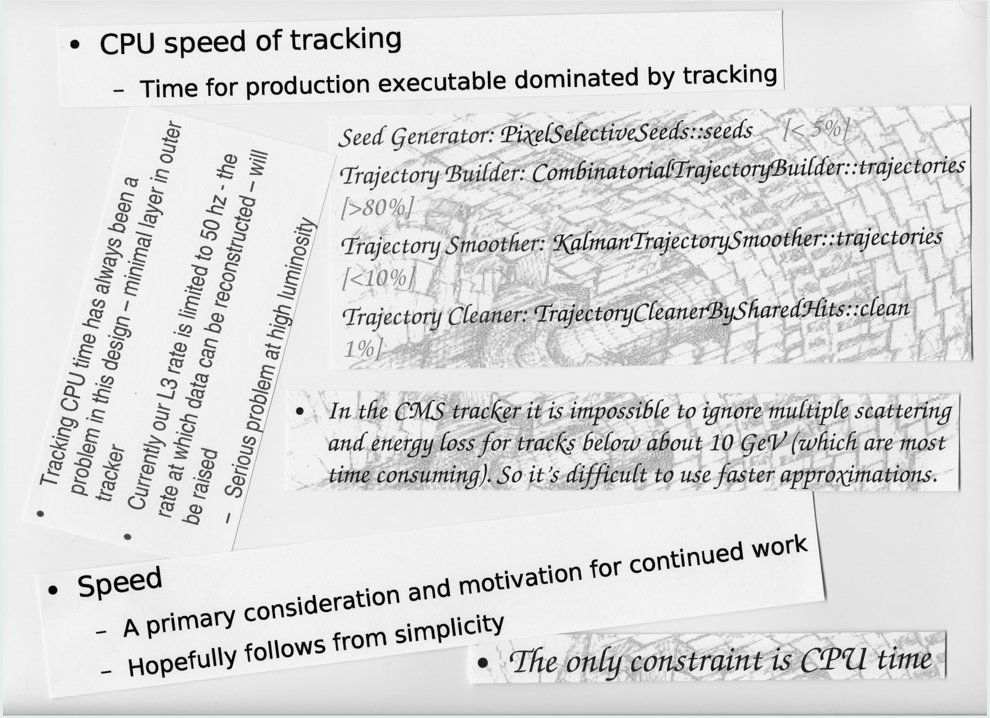
→ broken-line fit

Large influence of multiple scattering (msc)

- for low momenta,
- for high position measurement accuracy, and
- for dense material (large value of s/X_0).

CPU Speed of tracking

From talks – different experiments:



"At high luminosity, an average of 20 minimum bias events are expected per bunch crossing, which will produce more than 1000 reconstructable tracks in the tracker." [CMS at LHC]

Speed . . . and accuracy

and, for track finding,

Speed . . . and accuracy . . . and robustness

2. Track fitting methods

Track parameters are obtained by the fit of a track parametrization to the n measured track hits:

0. Simple least square fit with ideal parametrization — fast, with a computing time $\propto n$

1. Global track fitting methods — computing time $\propto n^{2 \cdots 3}$
(either trajectory described by non-diagonal $n \times n$ covariance matrix (matrix method), or by including parameters for N scattering planes, large matrix (breakpoint method);

2. The *progressive* method (P. Billoir 1984) ($\approx$ Kalman filter) — computing time $\propto n$
track is followed by incorporating measurement after measurement, starting from the outer detector, improving the parameter vector and covariance matrix;

2'. The Kalman filter (and smoothing) — computing time $\propto n$
the state vector and its covariance matrix are propagated to the next measurement, additional error (msc . . .) introduced as process noise; smoothing in direction opposite to the filter.

3. Broken-line fit — computing time $\propto n$

Results on track parameters from 1 to 3 are (almost) identical, and are, at low momentum, with (roughly) up to 30 % better curvature measurment than simple fit.

Track fitting is not only required to obtain the final track parameters for physics analysis, but also in the track-finding phase (pattern recognition). Very many fits of each track candidate are necessary to find the correct hits.

Requirements and computing times

Required results from a track fit algorithm:

- **A:** Optimal track parameters at **track-start** (vertex), for physics analysis;
- **B:** Correct covariance matrix for track parameters;
- **C:** Overall χ^2 of track, for test of quality of pattern recognition;
- **D:** Optimal track parameters at **track-end**, for extrapolation to other detectors;
- **E:** χ^2 of each single hit, for outlier test and improvement of hit selection.

Time for one track fit (no magnetic field), in **mikroseconds**, on standard PC:

Algorithm	$n = 25$	$n = 50$	$n = 100$	Remarks:
2-parameter least squares	0.270	0.420	0.720	bad fit in case of multiple scattering
Matrix method	150.000	943.000	6731.000	track-start parameters: **(A, B, C)**
Breakpoint method	117.000	556.000	2980.000	full reconstruction: **(A, B, C, D, E)**
Kalman backward	20.900	41.300	81.900	track-start parameters: **(A, B)**
Kalman back-/forward		approximately $\times 2$		full reconstruction: **(A, B, C, D, E)**
Broken-line fit	4.300	8.500	16.900	full reconstruction: **(A, B, C, D, E)**

Track-start parameters from methods in last 5 rows (almost) identical.

3. Track-fit algorithms based on broken lines

Is a method possible, which keeps the good properties of the Kalman filter/smoothing and allows to get the complete solution in one step?

- Treat the multiple scattering in all detail, which generates necessarily many parameters and matrices of large dimension;
- Use a mathematical model which results in equations with sparse matrices (many elements = 0), which can be **solved quickly**.

Simple and fast least squares track fits, with circle fit (Karimäki) for κ, d_{ca}, ϕ and straight line fit (z_0, $\tan\lambda$), are done to prepare the data for a detailed fit:

- Momentum already known (from $\kappa, \tan\lambda$) with some precision, allows to calculate multiple scattering variances;
- Residuals in xy-plane and in sz-plane can be calculated;
- A detailed fit follows, which takes into account multiple scattering (and other perturbations), applied to the residuals:

  (a) fit of z-residual versus track length s (straight line, no magnetic field);

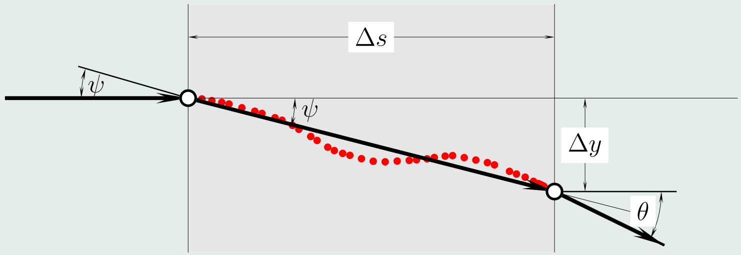
  (b) fit of circle residuals versus track length s including curvature correction $\Delta\kappa$.

A charged particle traversing a material layer is deflected by many small-angle scatters, mostly due to Coulomb scattering from nuclei, with variance

$$V[\theta] = \theta_0^2 = \left(\frac{13.6\ \mathrm{MeV}}{\beta \boldsymbol{p} c}\right)^2 \boldsymbol{t}\,[1 + 0.038 \ln \boldsymbol{t}]^2$$

where $\boldsymbol{t} = \Delta s / X_0$

$$\boldsymbol{V}\begin{bmatrix}\theta \\ \psi\end{bmatrix} = \begin{pmatrix}1 & 1/2 \\ 1/2 & 1/3\end{pmatrix}\theta_0^2$$



Multiple scattering in a layer of finite thickness

$\theta =$ projected angle of deflection, between direction before and behind layer

$\psi_{\text{left}} \equiv \psi$, $\psi_{\text{right}} \equiv \theta - \psi =$ angles between the **line, connecting the two intersection points**, and the direction before and behind layer

The **line, connecting the two intersection points**, can be determined by a measurement. Relevant for the reconstruction of the trajectory are the angles ψ_{left} and ψ_{right}, with covariance matrix

$$\boldsymbol{V}\begin{bmatrix}\psi_{\text{left}} \\ \psi_{\text{right}}\end{bmatrix}_i = \begin{pmatrix}1/3 & 1/6 \\ 1/6 & 1/3\end{pmatrix}\theta_{0,i}^2 \quad \text{for a homogeneous medium in layer } i$$

Material

Inhomogeneous material distribution in a layer of thickness Δs, with sublayers characterized by thickness in units of radiation length t_1, $t_2, \ldots t_k \ldots$. The total thickness in units of radiation length is

$$t = \sum_k t_k \qquad\qquad \theta_0^2 = \left(\frac{13.6\ \text{MeV}}{\beta \boldsymbol{p} c} \right)^2 \boldsymbol{t}\ [1 + 0.038 \ln \boldsymbol{t}]^2 \ .$$

Form the two sums:

$$C_1 = \frac{1}{2\,t\,\Delta s} \sum_k (r_k + r_{k-1})\, t_k$$

$$C_2 = \frac{1}{3\,t\,(\Delta s)^2} \sum_k \left(r_k^2 + r_k r_{k-1} + r_{k-1}^2 \right) t_k$$

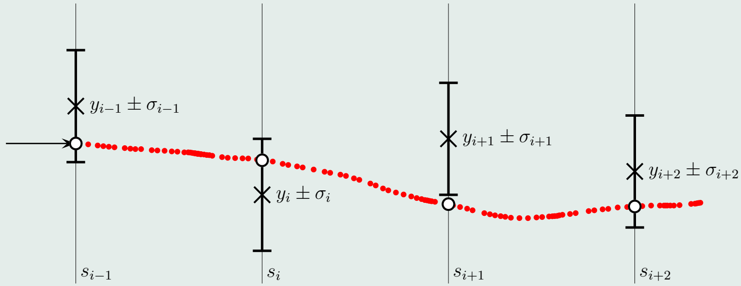
(r_{k-1} and r_k are the coordinates of the left and right side of the sublayer with t_k).

Covariance matrix

$$\boldsymbol{V} \begin{bmatrix} \psi_{\text{left}} \\ \psi_{\text{right}} \end{bmatrix}_i = \begin{pmatrix} V_{L,i} & V_{LR,i} \\ V_{LR,i} & V_{R,i} \end{pmatrix} = \begin{pmatrix} 1 - 2C_1 + C_2 & C_1 - C_2 \\ C_1 - C_2 & C_2 \end{pmatrix} \theta_0^2 \ ,$$

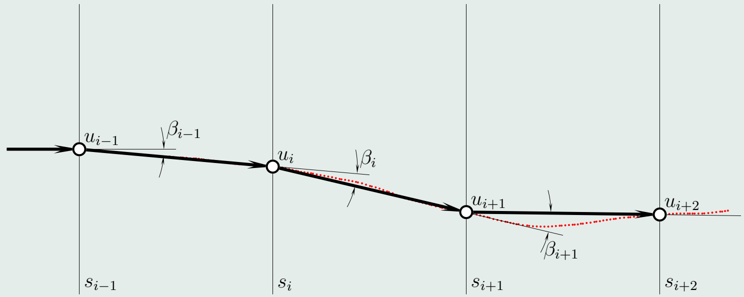
to be used within the track fit algorithm.

The particle track, given by the dotted curve, intersects the detector planes; the intersection points are drawn as circles. The result of the measurement are data points $y \pm \sigma$, given by crosses with error bar.



Instead of using a track parametrization with e.g. intercept and slopes as parameters, the proposed track-fit method uses **two phases in the track reconstruction**.

(1) Reconstruction of the trajectory: The trajectory, represented by the intersection points of the trajectory with the detector planes, is determined in a least squares fit; the fitted estimates of the intersection points are denoted by u_i.



(2) Track parameter determination: From the fitted u_i-values the two track parameters intercept and slope, required for the physics analysis, are determined at both sides of the track.

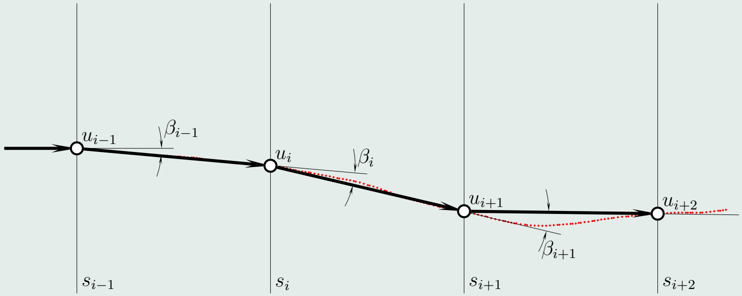
Kink angles

The intersection points u of the particle track with detector planes, drawn as circles, are connected by straight lines. The kink angles β are the angles between adjacent straight lines.

$$\beta_i = \psi_{\text{right},i-1} - \psi_{\text{left},i}$$

$$V[\beta_i] = \sigma^2_{\beta,i} = V[\psi_{\text{right},i-1}] + V[\psi_{\text{left},i}]$$

(multiple scattering)



There are $(n-2)$ kink angles β_i, which are linear functions of the values u_i:

$$\beta_i = \left[u_{i-1} \frac{1}{s_i - s_{i-1}} - u_i \frac{s_{i+1} - s_{i-1}}{(s_{i+1} - s_i)(s_i - s_{i-1})} + u_{i+1} \frac{1}{s_{i+1} - s_i} \right]$$

The values u_i are determined by minimization of the linear least squares expression (with weight $w_i = 1/\sigma_i^2$)

$$S(\boldsymbol{u}) = \sum_{i=1}^{n} w_i (y_i - u_i)^2 + \sum_{i=2}^{n-1} \frac{\beta_i^2}{\sigma^2_{\beta,i}}$$

with $n + (n-2)$ terms. Note: w_i may be zero or very small.

First phase: "Straight line" trajectory fit

The linear least squares expression $S(\boldsymbol{u})$ is minimized by the solution $\boldsymbol{u}$ of the matrix equation:

$$\boldsymbol{C}_u \, \boldsymbol{u} = \boldsymbol{r}_u \qquad \text{with the sparse } n\text{-by-}n \text{ matrix} \quad \boldsymbol{C}_u = \begin{pmatrix} d & x & x & & & & \\ x & d & x & x & & & \\ x & x & d & x & x & & \\ & x & x & d & x & x & \\ & & x & x & d & x & x \\ & & & & & \ddots & \\ & & & & & & \end{pmatrix}$$

i.e. $\boldsymbol{C}_u$ is a symmetric n-by-n band matrix of bandwidth $m = 2$. The elements of $\boldsymbol{C}_u$ and $\boldsymbol{r}_u$ are calculated by sums from the measured data.

After a Cholesky decomposition $\boldsymbol{C}_u = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ the matrix equation becomes $\boldsymbol{L}\left(\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{u}\right) = \boldsymbol{r}_u$. $\boldsymbol{L}$ is a left unit triangular matrix and $\boldsymbol{D}$ is diagonal; the band structure is kept, with computing time $\propto m^2 \cdot n$ instead of $\propto n^3$. The matrix $\boldsymbol{C}_u$ can be stored in a 3-by-n array and the decomposition can be made *in-place*.

The solution can be obtained by the following calculations:

decompose	$\boldsymbol{C}_u = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$	decomposition	$(6n)$
solve	$\boldsymbol{L}\boldsymbol{v} = \boldsymbol{r}_u$	for $\boldsymbol{v}$ by forward substitution	$(2n)$
solve	$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{u} = \boldsymbol{D}^{-1}\boldsymbol{v}$	for $\boldsymbol{u}$ by backward substitution	$(3n)$

The last column gives the number of dot-instructions (multiplication, division); the whole computation time is linear in n, i.e $\mathcal{O}(n)$ operations are necessary for the determination of $\boldsymbol{u}$.

Computing elements of the inverse of a band matrix

Computation the elements of the inverse matrix $\boldsymbol{Z} = \boldsymbol{C}^{-1}$ in the sparsity pattern of $\boldsymbol{L}$ can make use of the decomposition:

$$\boldsymbol{C} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$$

where $\boldsymbol{L}$ is unit left triangular, and $\boldsymbol{D}$ is diagonal.

$$\boldsymbol{Z} = \left(\boldsymbol{L}^{\mathrm{T}}\right)^{-1}\boldsymbol{D}^{-1} + \boldsymbol{Z}\left(\boldsymbol{I} - \boldsymbol{L}\right)$$

All elements of $\boldsymbol{Z}$ in the sparsity pattern can be computed without computing any elements outside this pattern.*)

Starting from Z_{nn} all elements

$$\text{for } i = n \ldots 1 : \qquad Z_{ii} = D_{ii}^{-1} - \sum_{k=i+1}^{i+2} Z_{ik}\, L_{ki} \qquad\qquad Z_{ij} = -\sum_{k=j+1}^{j+2} Z_{ik}\, L_{kj} \qquad j = i+1,\ i+2$$

are calculated in reverse order; when calculating Z_{ij} all required elements of $\boldsymbol{Z}$ are already calculated: $6n$ operations for band width of 2.

Computing time reduced by inline-code instead of bandwidth-loop, with 68 lines of code for matrix decomposition, solution of matrix equation, and computation of elements of inverse matrix for band width of 2.

*) K. Takahashi, J. Fagan and M. Chin, "Formation of a sparse bus impedance matrix and its applications to short circuit study", Proceedings 8th PICA Conference (1973), Minneapolis, Minnesota

Second phase: Track parameters

Corrections Δz_0 and $\Delta(\tan\lambda)$ are calculated from the two first u-values u_1 and u_2 amd added to the initial approximations $\widehat{z_0}$ and $\widehat{(\tan\lambda)}$:

$$\begin{pmatrix} z_0 \\ (\tan\lambda) \end{pmatrix} = \begin{pmatrix} \widehat{z_0} + u_1 \\ \widehat{(\tan\lambda)} + \dfrac{u_2 - u_1}{s_2 - s_1} \end{pmatrix}$$

By error propagation the covariance matrix is calculated from the elements of $\boldsymbol{V}_u \equiv \boldsymbol{Z}$:

$$\boldsymbol{V}(z_0, \, (\tan\lambda)) = \begin{pmatrix} Z_{11} & \dfrac{Z_{12} - Z_{11}}{s_2 - s_1} \\ \dfrac{Z_{12} - Z_{11}}{s_2 - s_1} & \dfrac{Z_{11} - 2Z_{12} + Z_{22}}{(s_2 - s_1)^2} + V[\psi_{\text{left},1}] \end{pmatrix}$$

Note the extra term $V[\psi_{\text{left},1}]$.

(b) Fit of circle residuals versus track length

In addition to the parameters in case (a) now a

curvature correction $\Delta\kappa$

has to be determined. The mean value of the "kink angle" β_i, as defined before (a), is now different from zero, due to the magnetic deflection.

This magnetic deflection is taken into account by the **re-definition**

$$\beta_i \approx \left[u_{i-1} \frac{1}{s_i - s_{i-1}} - u_i \frac{s_{i+1} - s_{i-1}}{\left(s_{i+1} - s_i\right)\left(s_i - s_{i-1}\right)} + u_{i+1} \frac{1}{s_{i+1} - s_i} \right] \quad + \frac{1}{2} \left(a_{i-1} + a_i\right) \cdot \boldsymbol{\Delta\kappa}$$

(a_i is the distance between the points i and $i+1$).

with $E\left[\beta_i\right] = 0$ and this has to be used in the function S

$$\boxed{S\left(\boldsymbol{u}, \Delta\kappa\right) = \sum_{i=1}^{n} \frac{\left(y_i - u_i\right)^2}{\sigma_i^2} + \sum_{i=2}^{n-1} \frac{\beta_i^2}{\sigma_{\beta,i}^2}}$$

which has to be minimized with respect to the values u_i and $\Delta\kappa$.

First phase: "Curved line" trajectory fit

The case of the additional parameter $\Delta\kappa$ is only slightly more complicated. The linear least squares expression $S(\boldsymbol{u}, \Delta\kappa)$ is minimized by the solution of the matrix equation:

$$\left(\begin{array}{c|c} \boldsymbol{C}_\kappa & \boldsymbol{c}^{\mathrm{T}} \\ \hline \boldsymbol{c} & \boldsymbol{C}_u \end{array}\right) \left(\begin{array}{c} \Delta\kappa \\ \hline \boldsymbol{u} \end{array}\right) = \left(\begin{array}{c} \boldsymbol{r}_\kappa \\ \hline \boldsymbol{r}_u \end{array}\right) \qquad \text{with matrix} \qquad \left(\begin{array}{c|cccccccc} \boldsymbol{C}_\kappa & c & c & c & c & c & c & c & \cdots \\ \hline c & d & x & x & & & & & \\ c & x & d & x & x & & & & \\ c & x & x & d & x & x & & & \\ c & & x & x & d & x & x & & \\ c & & & x & x & d & x & x & \\ c & & & & & & \ddots & & \\ \vdots & & & & & & & & \end{array}\right)$$

i.e. the matrix is a bordered band matrix.

Solution:			
	$\boldsymbol{C}_u = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$	decomposition	$(6n)$
	$\boldsymbol{C}_u \boldsymbol{z} = \boldsymbol{c}$	solution for $\boldsymbol{z}$	$(5n)$
	$\boldsymbol{B}_\kappa = \left(\boldsymbol{C}_\kappa - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{z}\right)^{-1}$	variance of curvature	$(n+1)$
	$\Delta\boldsymbol{\kappa} = \boldsymbol{B}_\kappa \left(\boldsymbol{r}_\kappa - \boldsymbol{z}^{\mathrm{T}}\boldsymbol{r}_u\right)$	curvature	$(n+1)$
	$\boldsymbol{C}_u \widetilde{\boldsymbol{u}} = \boldsymbol{r}_u$	solution for $\widetilde{\boldsymbol{u}}$	$(5n)$
	$\boldsymbol{u} = \widetilde{\boldsymbol{u}} - \boldsymbol{z}\Delta\boldsymbol{\kappa}$	smoothed coordinates	(n)

Second phase: Track parameters

The curvature correction $\Delta\kappa$ is already calculated; the corrections for position and direction are calculated, as before, from u_1 and u_2, with covariance matrix by error propagation.

The complete inverse matrix of the bordered band matrix is

$$\left(\begin{array}{c|c} \boldsymbol{C}_\kappa & \boldsymbol{c}^{\mathrm{T}} \\ \hline \boldsymbol{c} & \boldsymbol{C}_u \end{array}\right)^{-1} = \left(\begin{array}{c|c} \boldsymbol{B}_\kappa & -\boldsymbol{B}_\kappa \boldsymbol{z}^{\mathrm{T}} \\ \hline -\boldsymbol{z}\boldsymbol{B}_\kappa & \boldsymbol{C}_u^{-1} + \boldsymbol{z}\boldsymbol{B}_\kappa\boldsymbol{z}^{\mathrm{T}} \end{array}\right) ,$$

but only few elements are required, e.g. the element

$$\left(\boldsymbol{C}_u^{-1}\right)_{12} + \boldsymbol{B}_\kappa \cdot z_1\, z_2$$

Note: The matrices C_κ and B_κ are scalars in this application (only one common parameter). They become 2-by-2 matrices, if two common parameters are determined, e.g. $\Delta\kappa$ and ΔT_0 (drift chamber time-zero). The above formulas remain valid.

Covariance matrices

200 MeV/c track: fitted parameters and correlation matrix $\boldsymbol{C}$:

result of simple circle fit

$$\begin{pmatrix} \kappa \\ d_{ca} \\ \phi \end{pmatrix} = \begin{matrix} 0.1795 \times 10^{-1} & \pm 0.6585 \times 10^{-5} \\ 0.9164 & \pm 0.2116 \times 10^{-2} \\ 1.558 & \pm 0.2357 \times 10^{-3} \end{matrix} \qquad \boldsymbol{C} = \begin{pmatrix} 1 & & \\ 0.90 & 1 & \\ 0.97 & 0.94 & 1 \end{pmatrix}$$

result of broken line fit

$$\begin{pmatrix} \kappa \\ d_{ca} \\ \phi \end{pmatrix} = \begin{matrix} 0.1792 \times 10^{-1} & \pm 0.6528 \times 10^{-4} \\ 0.9330 & \pm 0.4813 \times 10^{-1} \\ 1.559 & \pm 0.6496 \times 10^{-2} \end{matrix} \qquad \boldsymbol{C} = \begin{pmatrix} 1 & & \\ 0.05 & 1 & \\ 0.09 & 0.95 & 1 \end{pmatrix}$$

Errors from circle fit underestimated by large factor; correlations of κ to the other parameters much smaller for broken line fit (κ is determined from whole track, the other parameters are determined essentailly from the first two hits.

10 GeV/c track: fitted parameters and correlation matrix $\boldsymbol{C}$:

result of simple circle fit

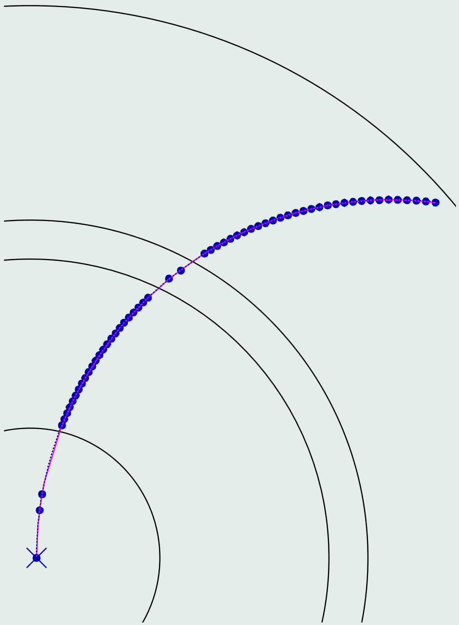
$$\begin{pmatrix} \kappa \\ d_{ca} \\ \phi \end{pmatrix} = \begin{matrix} 0.3599 \times 10^{-3} & \pm 0.8265 \times 10^{-5} \\ 1.024 & \pm 0.2392 \times 10^{-2} \\ 1.571 & \pm 0.2777 \times 10^{-3} \end{matrix} \qquad \boldsymbol{C} = \begin{pmatrix} 1 & & \\ 0.92 & 1 & \\ 0.98 & 0.95 & 1 \end{pmatrix}$$

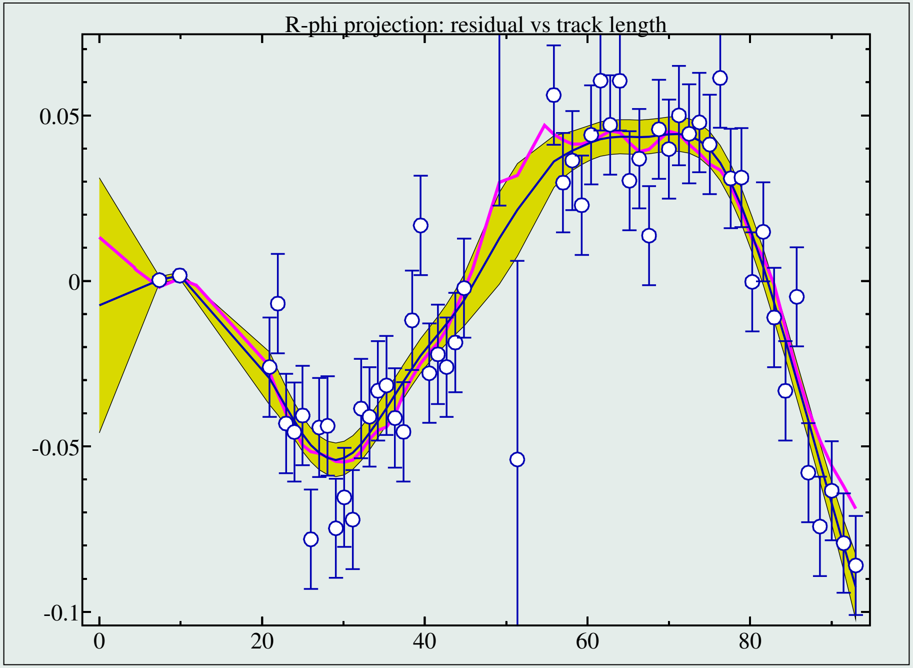
result of broken line fit

$$\begin{pmatrix} \kappa \\ d_{ca} \\ \phi \end{pmatrix} = \begin{matrix} 0.3597 \times 10^{-3} & \pm 0.8413 \times 10^{-5} \\ 1.024 & \pm 0.2652 \times 10^{-2} \\ 1.571 & \pm 0.3110 \times 10^{-3} \end{matrix} \qquad \boldsymbol{C} = \begin{pmatrix} 1 & & \\ 0.80 & 1 & \\ 0.85 & 0.89 & 1 \end{pmatrix}$$

Errors and correlations almost identical (they become identical at 100 GeV/c).

Blue line and yellow band is fit $\pm 1\sigma$

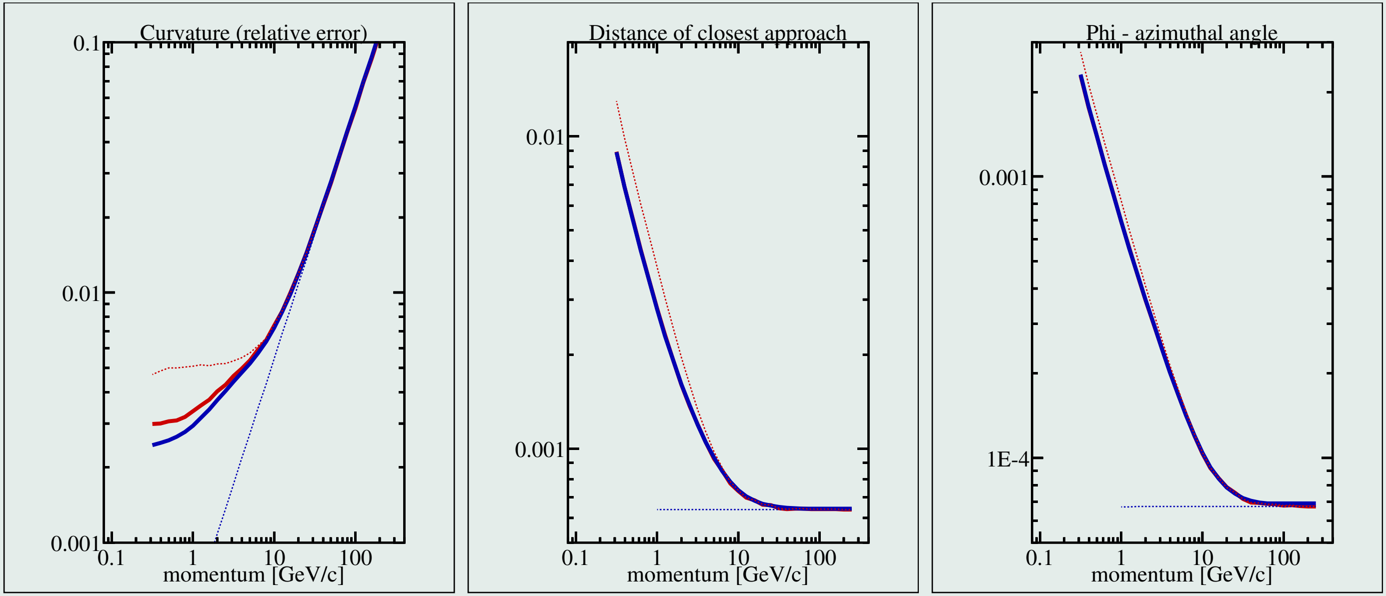




The broken-line fit result is given as blue broken-line with a ± 1 standard deviation yellow band, with extrapolation to the vertex at $s = 0$ (weight in fit was $w_1 = 0$). ← back

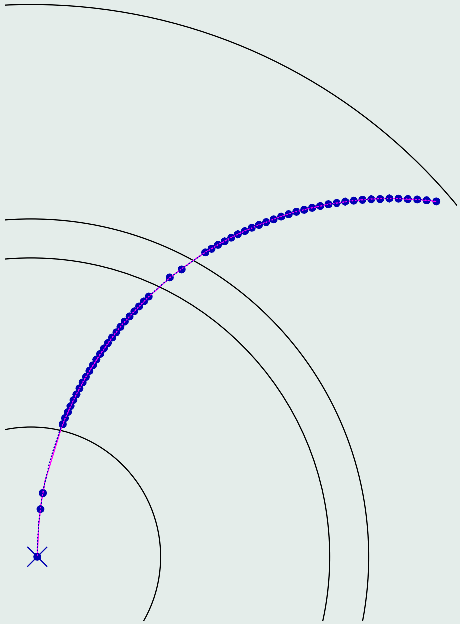
Thick red and thick blue lines: error from histogram fit (**red**) and calculated error (**blue**) for broken-line fit, with improvements by 37 % (momentum), by 31 % (d_{ca}) and 17 % (ϕ) at lowest momentum.

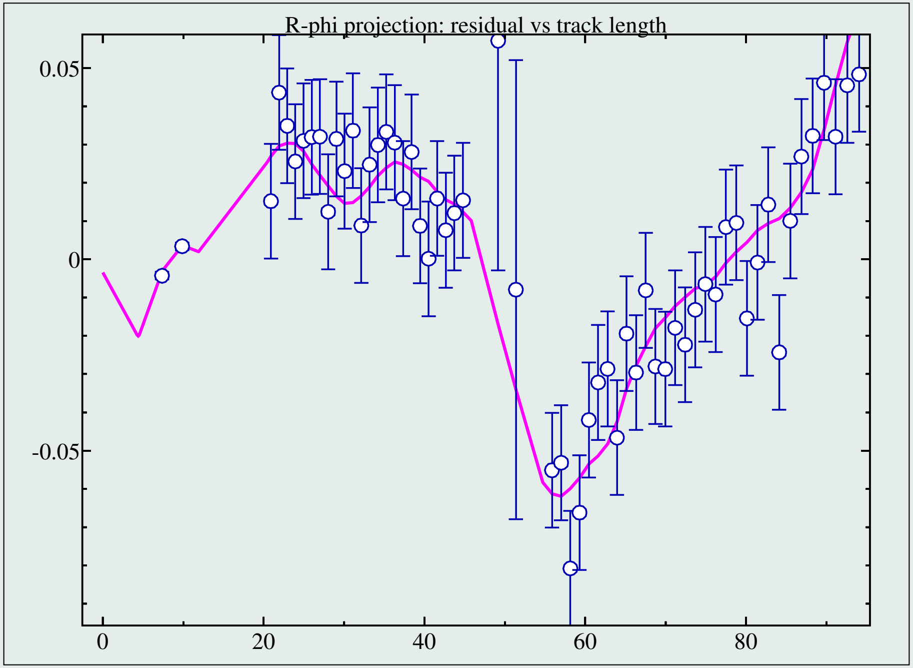
Thin dotted lines: error from histogram fit (red) and calculated error (blue) for simple LS fit



→ calculated errors are realistic for whole momentum range!

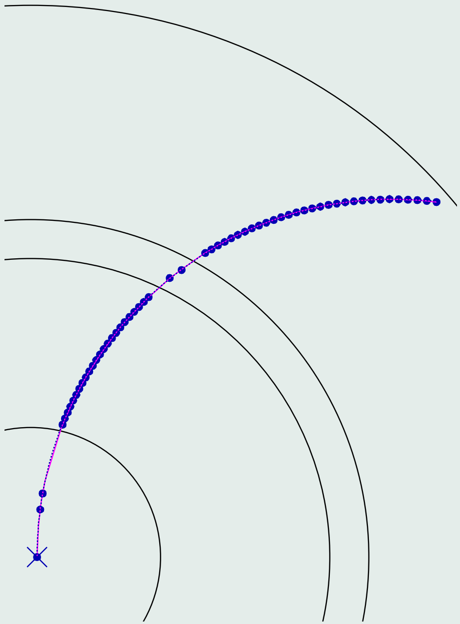
G. Lutz, Optimum track fitting in the presence of multiple scattering, NIMA 273 (1988) 349 – 361

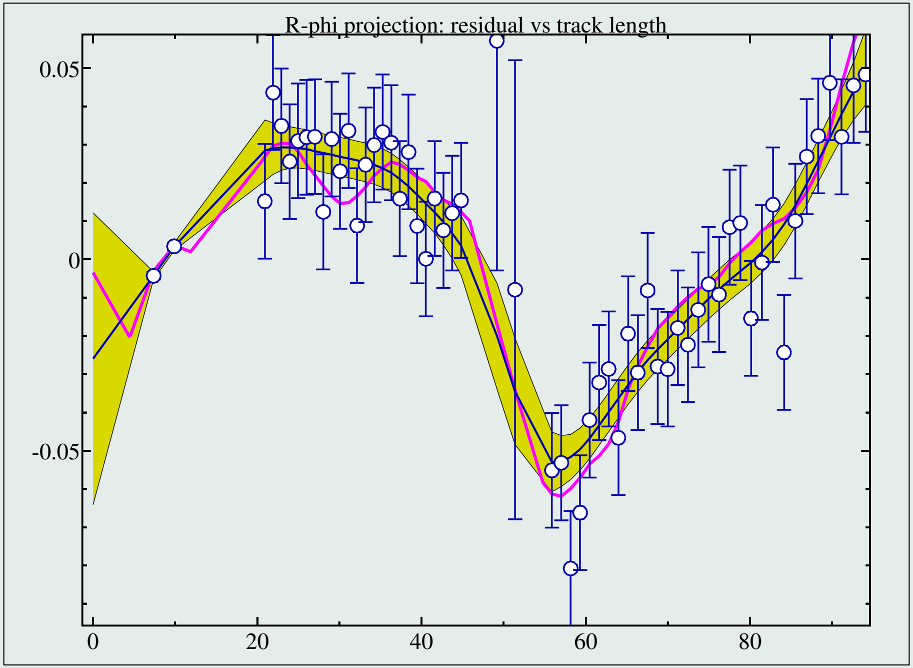




Magenta line is the true particle trajectory, with large kink angles in beam pipe, chamber walls etc.

. . . with broken-line fit — residual vs. track length



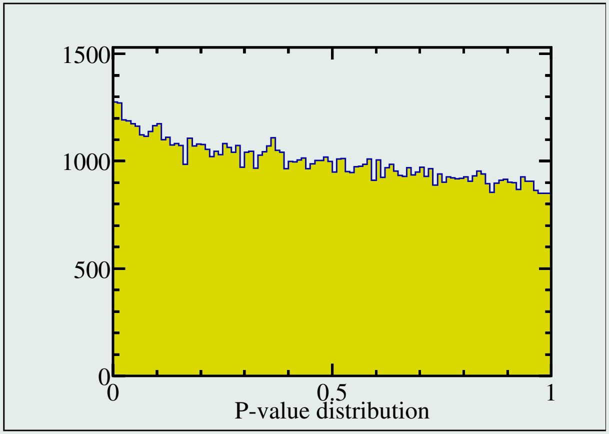


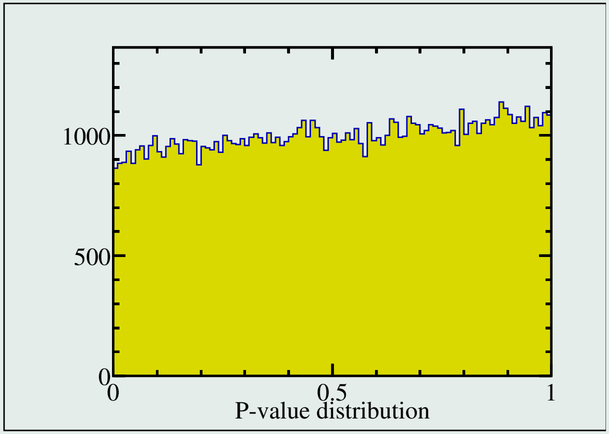
The broken-line fit result is given as a ±1 standard deviation band, with extrapolation to the vertex at $s = 0$ (weight in fit was $w_1 = 0$).

$$\chi^2_{(n-2)} \equiv S(\boldsymbol{u})_{\text{min}} = \chi^2\,(n \text{ measurements}) + \chi^2\,(n-2 \text{ kink angles})$$

with n fitted parameters.

Number of degrees of freedom $= n - 2$



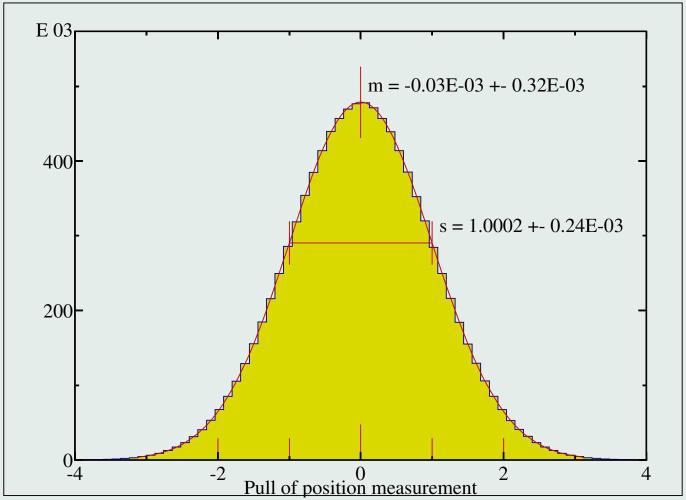


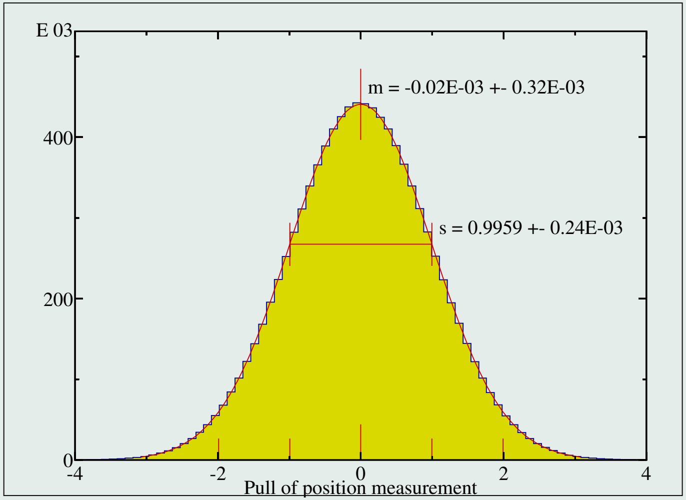
P-value distribution for the momenta of 0.5 GeV/c (left, mean value of $\chi^2 = 99.2$) and 10 GeV/c (right, mean value of $\chi^2 = 97.4$) are shown from the trajectory fit for a modified H1 configuration.

Pulls of position measurement

The pull of the position measurement is defined as the difference of the measured and fitted position, divided by the standard deviation of the difference:

$$p_{y,i} = \frac{y_i - u_i}{\sqrt{\sigma_i^2 - (\boldsymbol{V}_u)_{ii}}}$$





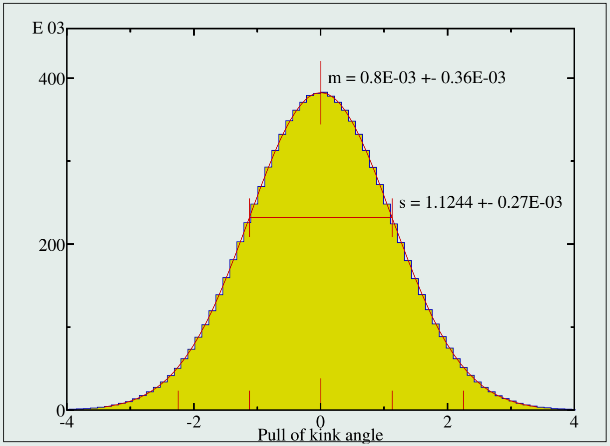
The pulls follow the expected $N(0,1)$ distribution for a momentum of 0.5 GeV/c (left) and for a momentum of 10 GeV/c (right) (H1 detector configuration with 2 CST and 56 CJC hits).

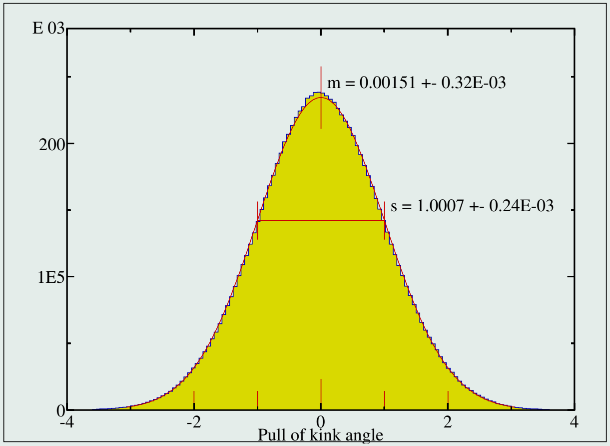
Pulls of kink angle measurement

The pull of the kink angle is defined as the difference of the expected (zero) and fitted angle, divided by the standard deviation of the difference:

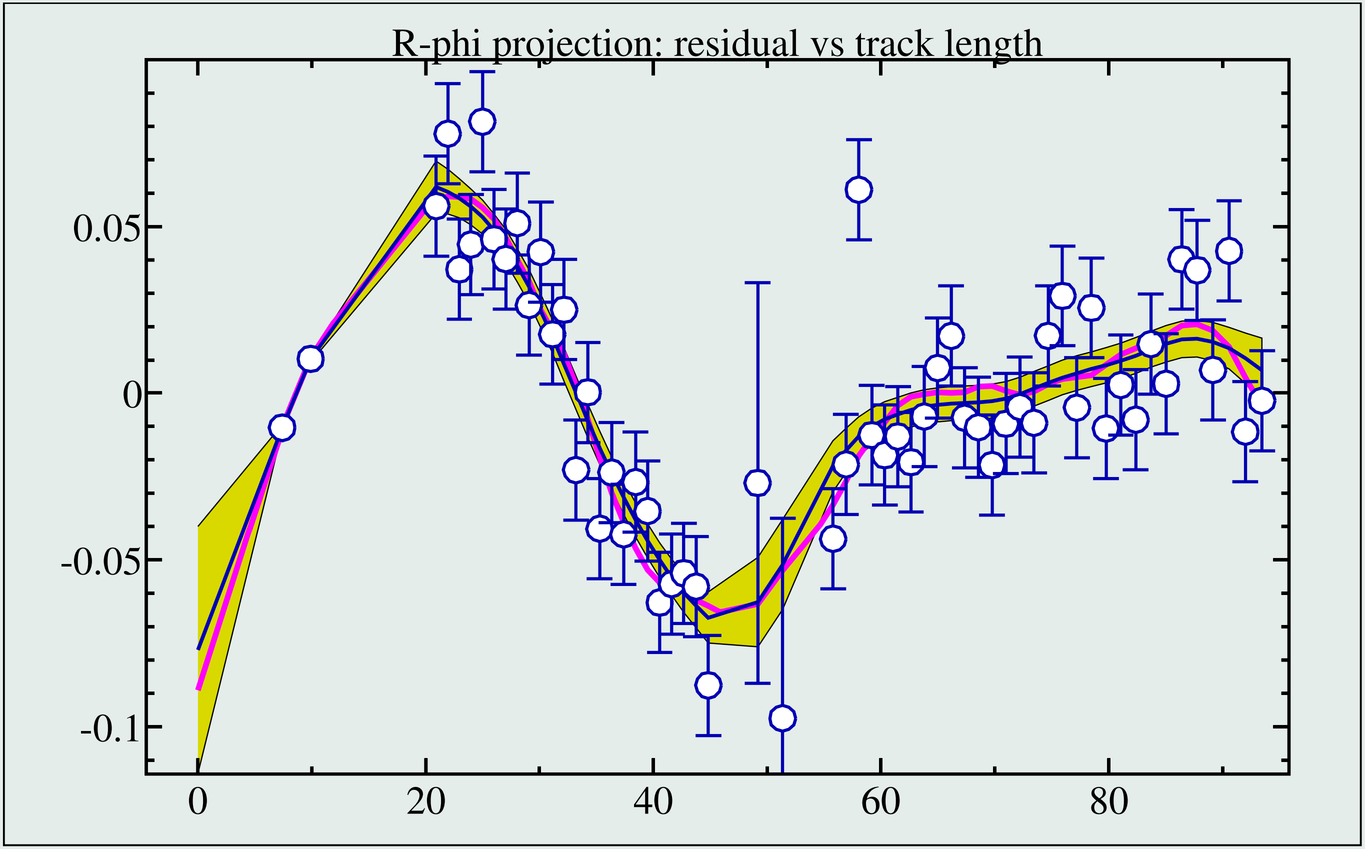
$$p_{\beta,i} = \frac{\beta_i}{\sqrt{\sigma^2_{\beta,i} - (\boldsymbol{V}_\beta)_{ii}}} ,$$

where $\boldsymbol{V}_\beta$ is calculated by error propagation from the covariance matrix $\boldsymbol{V}_u$.

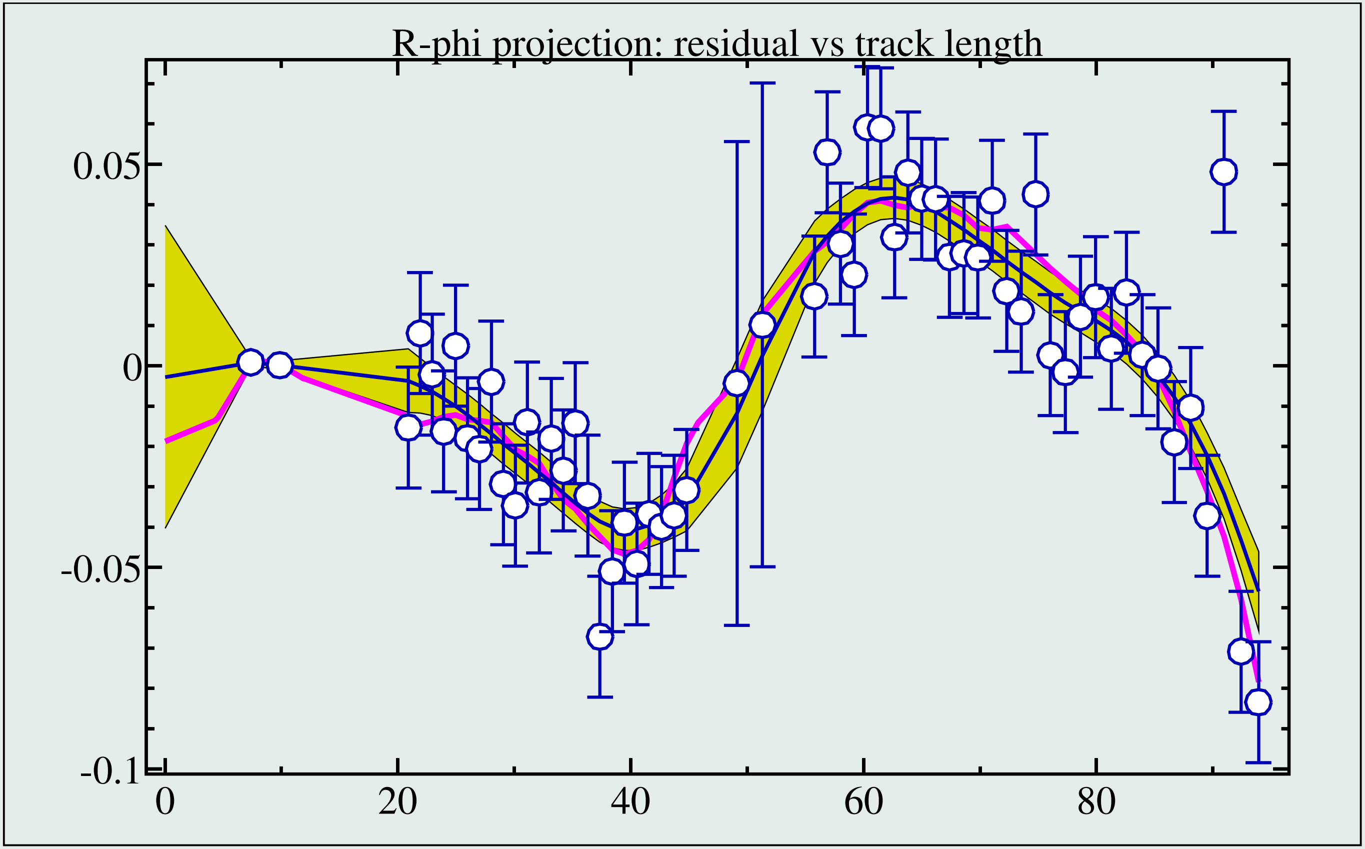




The pulls follow almost the expected $N(0, 1)$ distribution for a momentum of 0.5 GeV/c (left) and for a momentum of 10 GeV/c (right).

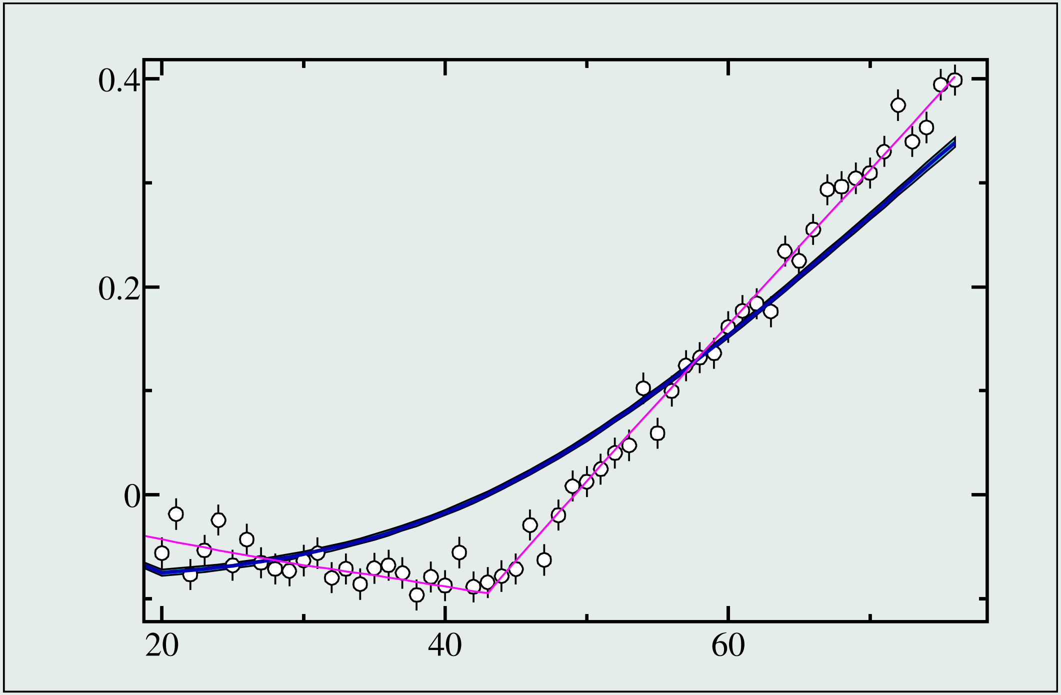

R-phi projection: residual vs track length
0.05
0
-0.05
-0.1
0
20
40
60
80

Outlier example 2 — 200 MeV/c, 6σ outlier



A large-angle scatter

A 1 GeV/c track with an artificial 1° kink around $s = 42$ cm:



The kink angle is much larger than the *rms* multiple scattering angle, expected for 1 GeV/c.
The overall χ^2 of the trajectory fit (blue curve) is large.

4. Robust estimation

- Least squares (LS) estimates: computed explicitly from the data, by means of some matrix algebra. They are optimal linear estimates, if the assumed variance is correct.
- "Outliers": arise from heavy tailed distributions or are simply bad data points due to errors, and have a large influence on the LS estimates; they pull the LS estimates towards them too much. A resulting examination of the residuals r_i is misleading because then they look more like normal ones.
- Robust estimators: can deal with data containing a certain fraction of outliers ➪ breakdown point. Outliers may be identified by looking at the residuals r_i from a robust fit.

Example is center of a symmetric distribution (1-dim):

LS estimate is mean, which has a break down point of 0 %,
the break down point of the median is as high as 50 %.

Outliers in a track fit, especially during track finding:

- correlated to the track: wider distribution around the true trajectory, caused e.g. by delta rays;
- uncorrelated: hits from neighbouring tracks, background hits, electronic noise;
- large angle scatter, decay, or a larger kink angle (multiple scattering).

Robust estimation methods

The breakdown point of an estimator is the smallest fraction of contamination that can cause the estimator to take on values arbitrarily far from the estimate of an uncontaminated data sample. The hightest possible value is 50 %.

- Generalized M-estimators: bound the influence of outlying data by means of a weight function. Breakdown point limited to a certain value that decreases with increasing number of parameters. [Huber, Hampel]

- Least median of squares (LMS): given by requirement

$$\text{minimize median}_i \; r_i^2$$

  with breakdown point of 50 %, but performs poorly from the point of view of asymptotic efficiency. The LMS should be considered as a data analytic tool, which is followed by an improvement to obtain higher efficency. [Rousseeuw, Leroy]

"It appears there are deep reasons why high-breakdown regression cannot be computed cheaply. ... Fortunately, the present evolution of computers has made robust regression quite feasible."

[Rousseeuw, 1986]

M-estimates

Generalization of least-squares, following from Maximum Likelihood arguments.

Abbreviation $z_i = \frac{y_i - f(x_i; a)}{\sigma_i}$ ($\sim N(0,1)$ for Gaussian measurement)

Least-squares: minimize $\sum_i \frac{1}{2} z_i^2$ solve $\sum_i \frac{y_i - f(x_i; a)}{\sigma_i^2} \frac{\partial f}{\partial a_j} = 0$ $j = 1, 2 \ldots p$

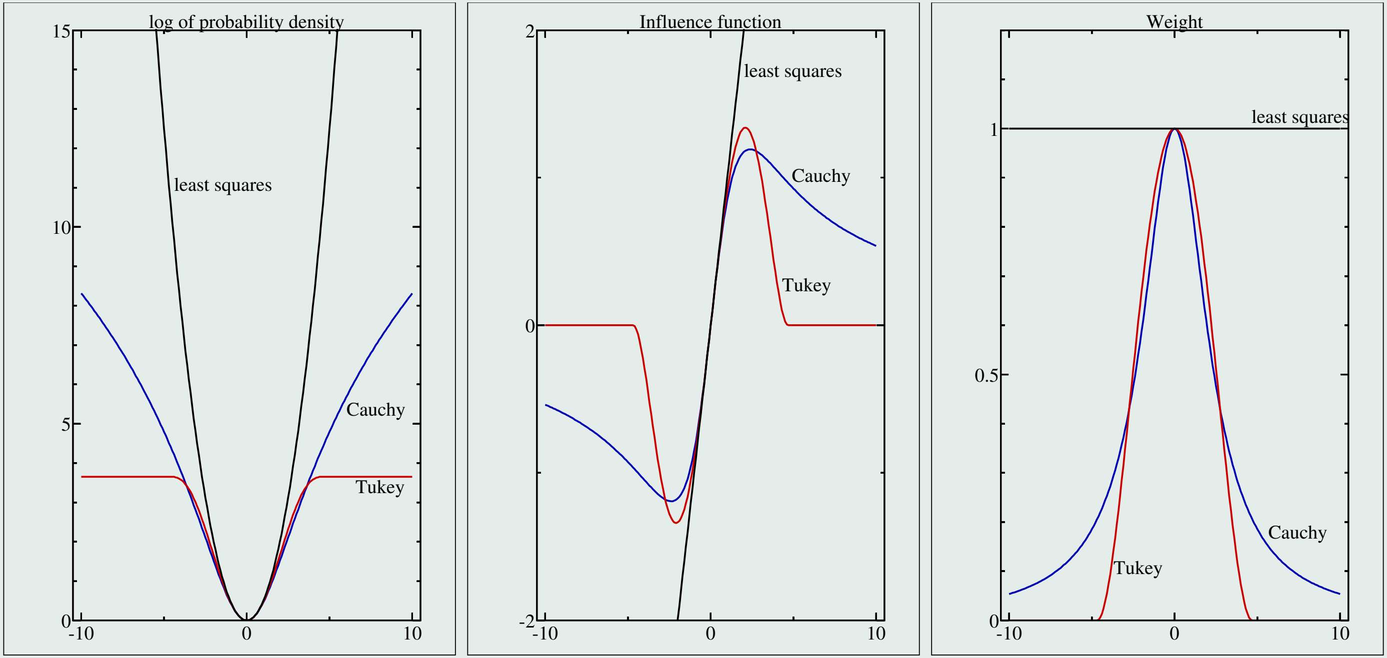
M-estimates: minimize $\sum_i \rho(z_i)$ solve $\sum_i \frac{y_i - f(x_i; a)}{\sigma_i^2} w(z_i) \frac{\partial f}{\partial a_j} = 0$ $j = 1, 2 \ldots p$

with weight function $w(z) = \psi(z)/z$ and influence function $\psi(z) = \frac{\mathrm{d}\rho}{\mathrm{d}z}$

Case of least-squares: $\rho(z) = \frac{1}{2} z^2$ $\psi(z) = z$ $w(z) = 1$

Requires iteration (non-linearity(!)) e.g with weight $w(z)$, calculated from previous values.

$$z \equiv \left(\frac{y - f(x)}{\sigma} \right)^2$$



$\rho(z)$ $\quad$ $\psi(z) = \mathrm{d}\rho(z)/\mathrm{d}z$ $\quad$ $w(z) = \psi(z)/z$

Commonly used M-estimators

	$\rho(z) = \ln \text{pdf}(z)$	influence function $\psi(z) = \mathrm{d}\rho(z)/\mathrm{d}z$	weight $w(z) = \psi(z)/z$
Least squares	$= \frac{1}{2} z^2$	$= z$	$= 1$
Cauchy	$= \frac{c^2}{2} \ln \left(a + (z/c)^2\right)$	$= \frac{z}{1 + (z/c)^2}$	$= \frac{1}{1 + (z/c)^2}$
Tukey $\begin{cases} \text{if } \|z\| \le c \\ \text{if } \|z\| > c \end{cases}$	$= \begin{cases} c^2/6 \left(1 - [1 - (z/c)^2]^3\right) \\ c^2/6 \end{cases}$	$= \begin{cases} z\,[1 - (z/c)^2]^2 \\ 0 \end{cases}$	$= \begin{cases} [1 - (z/c)^2]^2 \\ 0 \end{cases}$
Huber $\begin{cases} \text{if } \|z\| \le c \\ \text{if } \|z\| > c \end{cases}$	$= \begin{cases} z^2/2 \\ c\,(\|z\| - c/2) \end{cases}$	$= \begin{cases} z \\ c \cdot \text{sign}\,(z) \end{cases}$	$= \begin{cases} 1 \\ c/\|z\| \end{cases}$

Asymptotic efficiency of 95 % on the normal distribution obtained with $c = 2.3849$ (Cauchy), $c = 4.6851$ (Tukey) and $c = 1.345$ (Huber).

Least median of squares

It is impossible to write down a straightforward formula for the LMS estimator $\min \text{median}_i \, r_i^2$.

Algorithm: repeatedly

- select subsample of size p from p random(!) indices for regression with p parameters (or all possible subsamples);
- determine parameters and squares of residuals;
- find median

and take parameters from least median of squares.

A subsample is good if it consists of p good observations of the sample, which may contain a fraction of ϵ of bad observations. The probability that one of the m tried subsample is good is (for $n \gg p$)

$$P = 1 - \left(1 - (1 - \epsilon)^p\right)^m$$

	$p = 2$	$p = 3$
$P = 95\%$	$m = 11$	$m = 23$
$P = 99\%$	$m = 16$	$m = 35$

for $\epsilon = 50\%$

Improvement: reject early samples with too large median; consider only squares of residuals, which are $<$ previous smallest median. Use quick-select to find median. Stop if least-median-of-squares is already small enough.

5. Track finding and fitting

Track fitting during track finding takes much more time than the final track fit – fast and robust track fitting is essential.

Kalman filter is a recursive procedure, where the estimates of the of the track parameters, starting from an initial trajectory (seed), are updated and improved with each successive hit. Kalman filter is, in principle, able to integrate pattern recognition and track fitting.

The most often used algorithm for track reconstruction is the Combinatorial Kalman filter (CKF):

- Start from "seed".
- Iterate the following steps:
  - extrapolate to next layer, (only next layer!)
  - look for compatible hits, and for each candidate, generate a branch with each compatible hit and with the missing hit,
  - cleanup: drop bad candidates.
- Final cleanup: select "best' candidate (χ^2 and ndf)

Note: "extrapolation" is always dangerous; "interpolation" is much safer!

Limit of number of candidates in combinatorical trajectory building is compromise between speed and risk to loose the right track.

Hard assignment is known to be sub-optimal in dense environments.

Advanced track fitting techniques to optimize track recognition:

- Deterministic annealing filter (DAF) (reducing effect of noise hits by a low weight)
- Multi-track-fit (MTF) (simultaneous fit of several tracks suited to very dense jets)
- Gaussian-Sum filter (GSF) (if energy loss is non-Gaussian - electrons mixture of Gaussian components, parallel Kalman filters)
- Competition with hard assignment (CHA)
- Competition with soft assignment (CSA)

All sophisticated methods require more computing time than the direct Kalman filter.

Track finding with broken line fit

Algorithm: loop, until no further track found

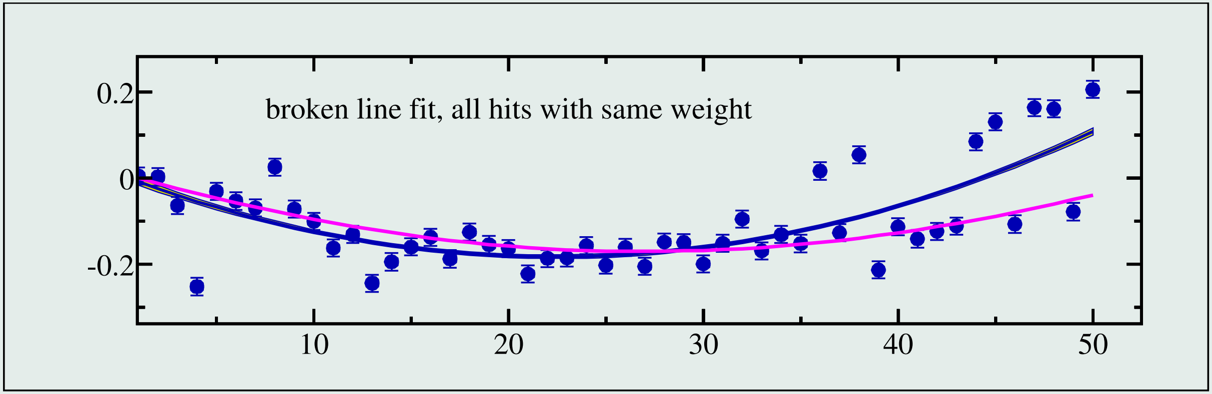
- Search for seeds = trajectory candidates within the hits unused so far, avoiding dense regions, using different techniques. Short seeds have to be compatible with beam spot.
- Iterate with trajectory candidate improvement:
  - sort the (remaining) candidates according to length (number of hits).
  - collect (unused) hits by forward- and backward-extending the candidate, and make a robust fit, replacing candidate parameters by improved values.
- Final hit collection and robust fit, assign hits to track and mark track hits as used.

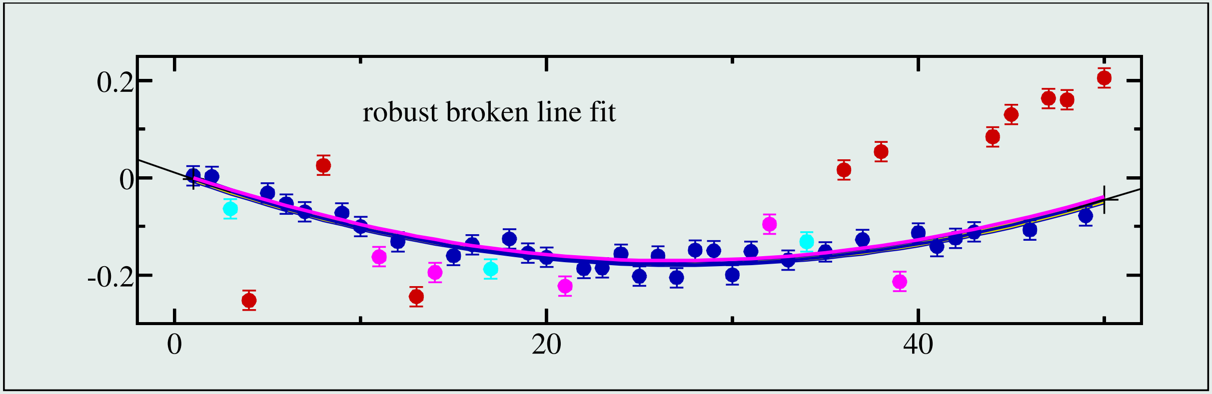
Fast fit in two steps, robust against large fraction of outliers, with

1. least-median-of-squares raw determination of straight line or parabola,
2. iterative broken-line fit with robust method of M-estimates, using Tukeys or Cauchy formula,

- include ΔT_0 parameter in broken-line fit, if necessary, to improve track and event T_0.

Weight function of M-estimate applied to position residuals.
No function like exp or sqrt called in robust fit, because of time consumption.





Example: 30 % of the hits are outliers (20 % from crossing track, 5 % bad hits, 5 % random hits).

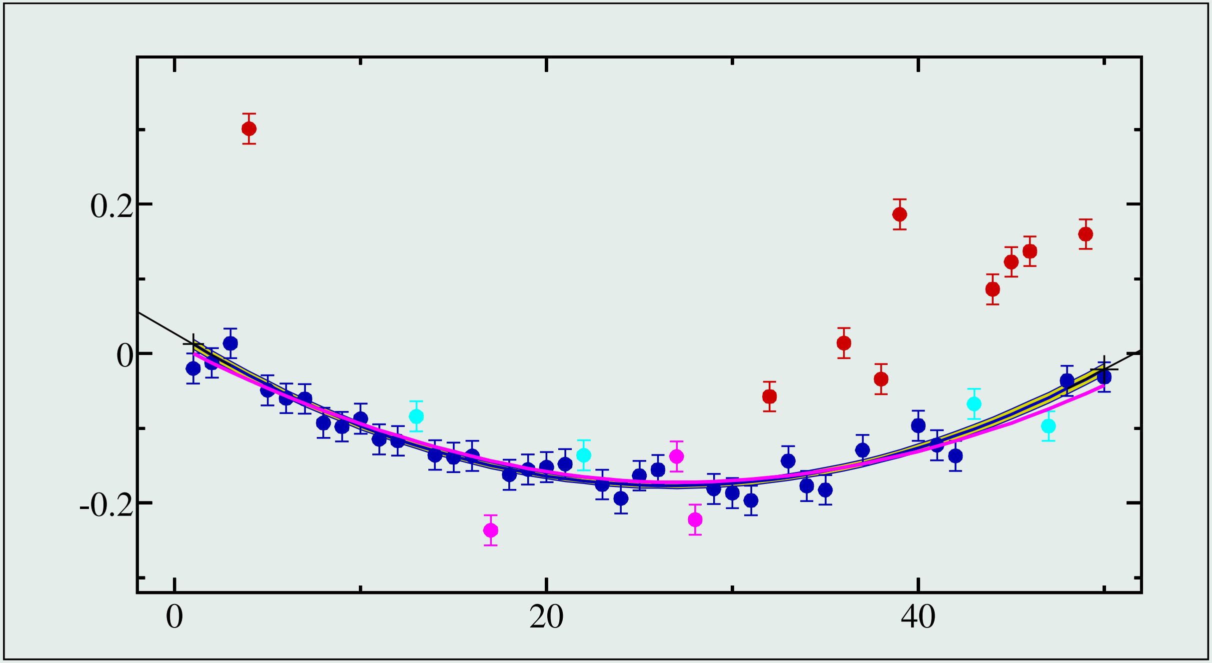
Computing times

Time for one track fit in **mikroseconds**, on standard PC:

Algorithm	$n = 25$	$n = 50$	$n = 100$	Remarks:
Kalman back-/forward	42.8	82.6	163.8	full reconstruction, no curvature
Broken-line fit	4.3	8.5	16.9	full reconstruction, no curvature
Broken-line fit	6.2	12.1	24.3	full reconstruction, with curvature
Robust broken-line fit	13.9	27.2	65.0	reconstr. with curvature, 0 % outlier
Robust broken-line fit	21.9	42.4	99.6	reconstr. with curvature, 30 % outlier

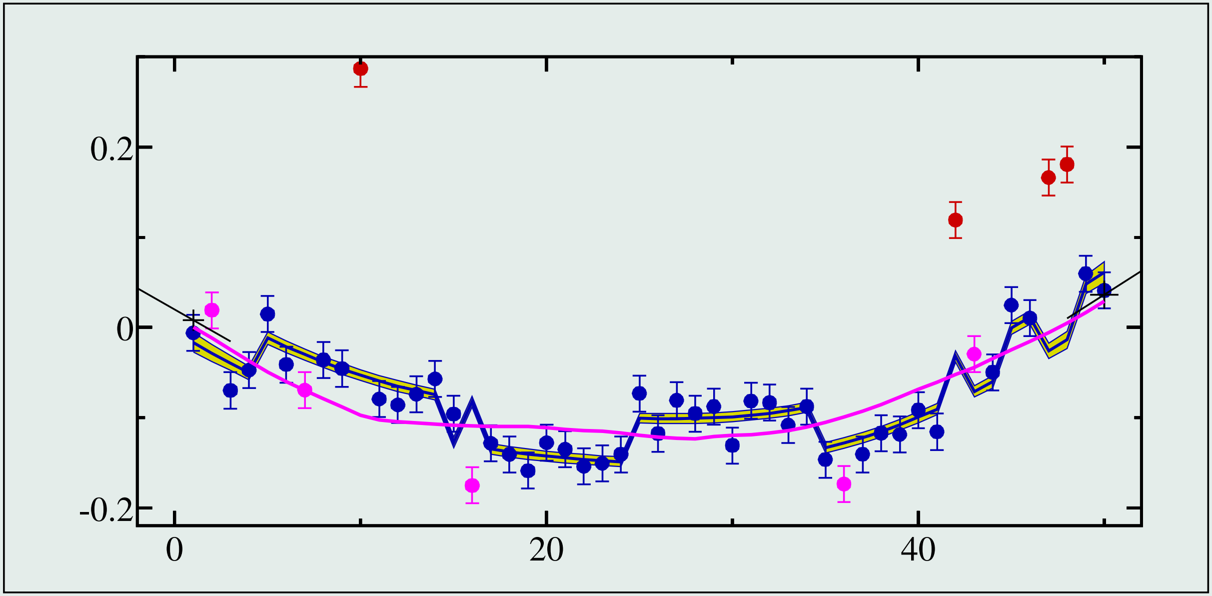
Robust, iterative two-step fit is still rather fast, even for large fraction of outliers.

Broken line fit, with down-weighting of large residuals by Tukeys weight:



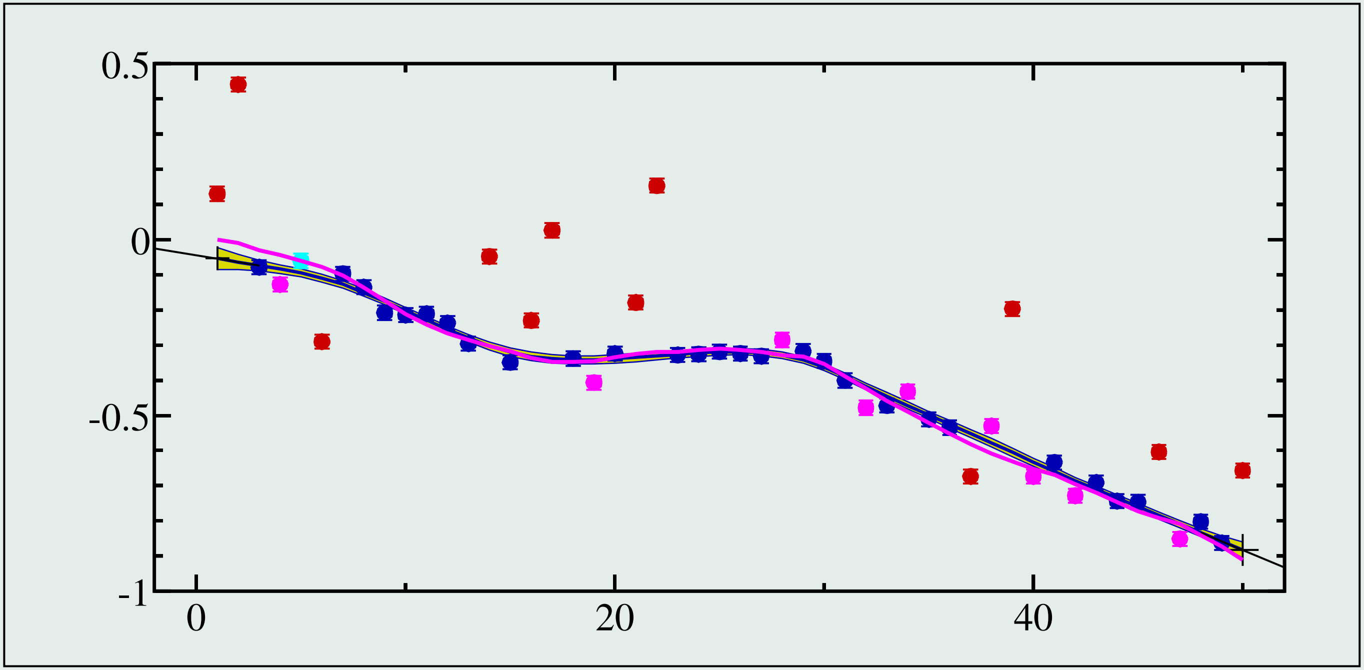
Colour code for data points:
blue hits have residual $< 1.645\sigma$ [90% expected for Gaussian];
cyan hits have residual $< 1.96\sigma$ [5%], magenta hits have residual $< 4.7\sigma$ [5%], red hits have residual $> 4.7\sigma$ and have zero weight.

Broken line fit, with determination of T_0, with multiple scattering, with down-weighting of large residuals by Tukeys weight:



Each hit with sign of drift distance. Outlier fraction is 20 %.

Broken line fit with large multiple scattering, with down-weighting of large residuals by Tukeys weight:



Summary

Broken-line fits: New algorithm, faster by factor ≈ 10 in comparison to Kalman filter (under test conditions), with fit in one step (no initial values, no iteration, no recursion, no direction backward or forward) in time $\mathcal{O}(n)$.

Improved accuracy with correct covariance matrix: Improvement of the track reconstruction precision for tracks of low momentum in a dense detector, for high position measurement accuracy. Extension to include energy loss (deterministic) for heavy particles, and magnetic-field inhomogenities are possible.

Full information: The algorithm gives the full information on every measured point on a trajectory:

fitted value with propagated error, pull of position and kink angle

and this information can be used already during track finding, to remove bad hits or to cut the trajectory, and to adjust material assumption.

Fast, robust track fit for track finding: using combination of least-median-of-squares and M-estimation, with down-weighting or removal of outliers, and improvement of T_0 for drift-chamber data.

Contents