

# Smoothing

---

or

## Fitting without a parametrization

Volker Blobel – University of Hamburg  
March 2005

1. Why smoothing
2. Running median smoother
3. Orthogonal polynomials
4. Transformations
5. Spline functions
6. The SPLFT algorithm
7. A new track-fit algorithm

## Why smoothing?

---

A standard task of data analysis is fitting a **model**, the form of which is determined by a small number of parameters, to data. Fitting involves the estimation of the small number of parameters from the data and requires two steps:

- identification of a suitable model,
- checking to see if the fitted model adequately fits the data,

Parametric models can be flexible, but they will not adequately fit all data sets.

**Alternative:** models not be written explicitly in terms of parameters; flexible for a wide range of situations.

- produce smoothed plots to aid understanding;
- identifying a suitable parametric model from the shape of the smoothed data;
- eliminate complex effects that are not of direct interest so that attention is focused on the effects of interest;
- deliver more precise interpolated data for subsequent computation.

A **smoother** fits a smooth curve through data, the fit is called the smooth (the signal), the residuals the rough:

$$\text{Data} = \text{Smooth} + \text{Rough}$$

- The **Signal** is assumed to vary smoothly most of the time, perhaps with a few abrupt shifts.

The **Rough** can be considered as the sum of

- additive noise from a symmetric distribution with mean zero and a certain variance, and
- impulsive, spiky noise: outliers.

Examples for smoothing methods:

- **Moving averages**; they are popular, but they are highly vulnerable to outliers.
- Tukey suggested **running medians** for removing outliers, preserving level shifts, but standard medians have other deficiencies.

---

J. W. Tukey: Exploratory Data Analysis, Addison-Wesley, 1977

## Running median smoother

---

Data from a set  $\{x_1, x_2, \dots, x_n\}$  are assumed to be equidistant, e.g. from a time series.

In a single smooth operation values smoothed values  $y_i$  are calculated from a window of the set  $\{x_1, x_2, \dots, x_n\}$ ; then the  $y_i$  replace the  $x_i$  in the set.

In a median smooth with window size  $2 \cdot k + 1$ , the smoothed value  $y_i$  at location  $i$  in the set  $\{x_1, x_2, \dots, x_n\}$  is the **middle** value of  $x_{i-k}, x_{i-k+1}, \dots, x_{i+k-1}, x_{i+k}$ :

$$y_i = \text{median}[x_{i-k}, x_{i-k+1}, \dots, x_{i+k-1}, x_{i+k}]$$

Especially the window size 3 is important. It can be realized by the statement

$$y(i) = \min(\max(x(i-1), x(i)), \max(x(i), x(i+1)), \max(x(i+1), x(i-1)))$$

For median with window size three, there is a special prescription to modify (smooth) the endpoints, using Tukey's extrapolation method:

$$y_1 = \text{median}[y_2, y_3, 3y_2 - 2y_3] \quad y_n = \text{median}[y_n, y_{n-1}, 3y_{n-1} - 2y_n]$$

A large window size requires a sorting routine.

### **Running medians are efficiently in removing outliers.**

Running medians are supplemented by other single operations like averaging. Several single operations are combined to a complete smooth.

## Single operations

---

The following single operations are defined:

[**2,3,4,5,...** ] Perform median smooth with the given window size. For median with window size three, smooth the two end values using Tukey's extrapolation method

$$y_1 = \text{median}[y_2, y_3, 3y_2 - 2y_3] \quad y_n = \text{median}[y_n, y_{n-1}, 3y_{n-1} - 2y_n]$$

[**3'** ] Perform median smooth with the window size 3, but do not smooth the end-values.

**R** following a median smooth: continue to apply the median smooth until no more changes occur.

**H** Hanning: convolution with a symmetrical kernel with weights  $(1/4, 1/2, 1/4)$ , (end values are not changed)

$$\text{smoothed value } y_i = \frac{1}{4}x_{i-1} + \frac{1}{2}x_i + \frac{1}{4}x_{i+1}$$

**G** Conditional hanning (only non-monotone sequences of length 3)

**S** Split: dissect the sequence into shorter subsequences at all places where two successive values are *identical*, apply a 3R smooth to each sequence, reassemble and polish with a 3 smooth.

> Skip mean: convolution with a symmetrical kernel with weights  $(1/2, 0, 1/2)$ ,

$$\text{smoothed value } y_i = \frac{1}{2}x_{i-1} + \frac{1}{2}x_{i+1}$$

**Q** quadratic interpolation of flat 3-areas.

## Combined operations

---

Single operations are combined to the complete smoothing operation like **353H**.

The operation **twice** means doubling the sequence:

- After the first pass the residuals are computed (called reroughing);
- the same smoothing sequence is applied to the residuals in a second pass, adding the result to the smooth of the first pass.

[**4253H,twice** ] Running median of 4, then 2, then 5, then 3 followed by hanning. The result is then roughed by computing residuals, applying the same smoother to them and adding the result to the smooth of the first pass.

This smoother is recommended.

[**3RSSH,twice** ] Running median of 3, two splitting operations to improve the smooth sequence, each of which is followed by a running median of 3, and finally hanning. Reroughing as described before.

Computations are simple and can be calculated by hand.

[**353QH** ] older recipe, used in *SPLFT*.

Standard running medians have deficiencies in trend periods.

Suggested improved strategy: fit at each location  $i$  a local linear trend within a window of size  $2k + 1$ :

$$y_{i+j} = y_i + j \beta_i \quad j = -k, \dots, +k$$

with slope  $\beta_i$  to the data  $\{x_{i-k}, \dots, x_{i+k}\}$  by **robust regression**.

Example: Siegel's repeated median removes the data slope before the calculation of the median:

$$y_i = \text{median} [x_{i-k} + k \beta_i, \dots, x_{i+k} - k \beta_i]$$
$$\beta_i = \text{median}_{j=-k, \dots, +k} \left[ \text{median}_{\ell \neq j} \frac{x_{i+j} - x_{i+\ell}}{j - \ell} \right]$$

# Orthogonal polynomials

---

Given data  $y_i$  with standard deviation  $\sigma_i$ ,  $i = 1, 2, \dots, n$ .

If data interpolation required, but no parametrization known: fit of normal general  $p$ -th polynomial to the data

$$f(x_i) \cong y_i \quad \text{with} \quad f(x) = \sum_{j=0}^p a_j x^j$$

Generalization of least squares straight-line fit to general  $p$ -th order polynomial straightforward: matrix  $\mathbf{A}^T \mathbf{W} \mathbf{A}$  contains sum of all powers of  $x_i$  up to  $\sum x_i^{2p}$ .

Disadvantage:

- fit numerically unstable (bad condition number of matrix);
- statistical determination of optimal order  $p$  difficult;
- value of coefficients  $a_j$  depend on highest exponent  $p$ .

Better: fit of orthogonal polynomial

$$f(x) = \sum_{j=0}^p a_j p_j(x)$$

with  $p_j(x)$  constructed from the data.

## Straight line fit

---

Parametrization

$$f(x) = a_0 + a_1x \quad \text{replaced by} \quad f(x) = a_0 \cdot p_0(x) + a_1 \cdot p_1(x)$$

$$x \text{ mean value } \langle x \rangle = \frac{\sum w_i x_i}{\sum w_i} \quad \text{with weight } w_i = 1/\sigma_i^2$$

The orthogonal polynomials  $p_0(x)$  and  $p_1(x)$  are defined by

$$\begin{aligned} p_0(x) &= b_{00} \\ p_1(x) &= b_{10} + b_{11}x = b_{11}(x - \langle x \rangle) \end{aligned} .$$

with coefficients  $b_{00}$  and  $b_{11}$

$$b_{00} = \left( \sum_{i=1}^n w_i \right)^{-1/2} \quad b_{11} = \left( \sum_{i=1}^n w_i (x_i - \langle x \rangle)^2 \right)^{-1/2}$$

with the coefficient  $b_{10} = -\langle x \rangle b_{11}$ .

$$\begin{aligned} \mathbf{A}^T \mathbf{W} \mathbf{A} &= \begin{pmatrix} \sum w_i p_0^2(x_i) & \sum w_i p_0(x_i) p_1(x_i) \\ \sum w_i p_0(x_i) p_1(x_i) & \sum w_i p_1^2(x_i) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \mathbf{A}^T \mathbf{W} \mathbf{y} &= \begin{pmatrix} \sum w_i p_0(x_i) y_i \\ \sum w_i p_1(x_i) y_i \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \end{aligned}$$

i.e. covariance matrix is unit matrix and parameters  $\mathbf{a}$  are calculated by sums (no matrix inversion). Intercept  $a_0$  and slope  $a_1$  are uncorrelated.

## General orthogonal polynomial

---

$$f(x) = \sum_{j=0}^p a_j p_j(x) \quad \text{with} \quad \mathbf{A}^T \mathbf{W} \mathbf{A} = \text{unit matrix}$$

... means construction of orthogonal polynomial from (for) the data!

Construction of higher order polynomial  $p_j(x)$  by recurrence relation:

$$\gamma p_j(x) = (x_i - \alpha) p_{j-1}(x) - \beta p_{j-2}(x)$$

from the two previous functions  $p_{j-1}(x)$  and  $p_{j-2}(x)$  with parameters  $\alpha$  and  $\beta$  defined by

$$\alpha = \sum_{i=1}^n w_i x_i p_{j-1}^2(x_i) \quad \beta = \sum_{i=1}^n w_i x_i p_{j-1}(x_i) p_{j-2}(x_i)$$

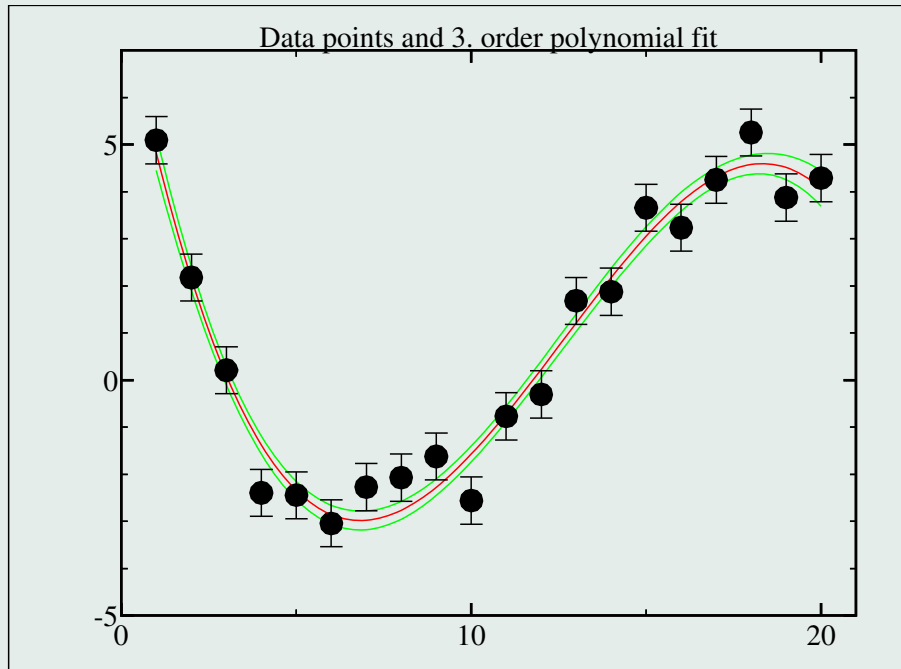
with normalization factor  $\gamma$  given by

$$\gamma^2 = \sum_{i=1}^n w_i [(x_i - \alpha) p_{j-1}(x_i) - \beta p_{j-2}(x_i)]^2$$

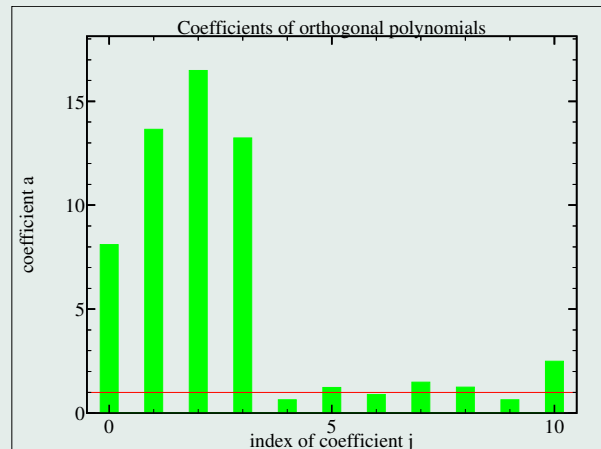
Parameters  $\hat{a}_j$  are determined from data by

$$\hat{a}_j = \sum_{i=1}^n w_i p_j(x_i) \cdot y_i$$

Interpolated function value and error:  $f = \sum_{j=0}^p a_j p_j(x_i)$        $\sigma_f = \left( \sum_{j=0}^p p_j^2(x_i) \right)^{1/2}$



- Fit sufficiently high order  $p$  and test value of each coefficient  $a_j$ , which are independent and have a standard deviation of 1;
- remove of all higher, insignificant coefficients, which are compatible with zero.



Note: each coefficient is determined by *all* data points. True values  $a_j$  with larger index  $j$  approach zero rapidly for a smooth dependence (similar to Fourier analysis and synthesis). A narrow structure requires many terms.

## Least squares smoothing

---

From publications:

In least squares smoothing each point along the smoothed curve is obtained from a polynomial regression with the closest points more heavily weighted.

Eventually outlier removal by running medians should precede the least squares fits.

A method called *loess* includes “locally weighed regression.”

Orthogonal polynomials allow to determine the necessary maximum degree for each window.

In the histogram smoother **SPLFT** least squares smoothing using orthogonal polynomials is used to estimate the higher *derivatives* of the smooth signal from the data.

These derivative are then used to determine an optimal knot sequence for a subsequent B-spline fit of the data, which gives the final smooth result.

## Transformations

---

Smoothing works better if the true signal shape is rather smooth. Transformations can improve the result of a smoothing operation by smoothing the shape of the distribution and/or to stabilize the variance to the data.

Taking the logarithm is an efficient smoother for exponential shapes.

Stabilization of the variance:

- **Poisson** data  $x$  (counts) with variance  $V = x$  are transformed by

$$y = \sqrt{x} \quad \sigma_y = 1/2$$

or better  $y = \sqrt{x} + \sqrt{x+1} - 1 \quad \sigma_y = 1$

to data with constant variance. The last transformation has a variance of one plus or minus 6 %. In addition the shape of the distribution becomes smoother.

- Data  $x$  from a **binomial** distribution ( $n$  trials) have a variance which is function of its expected value and of the sample size  $n$ . A proposed transformation (Tukey) is

$$y = \arcsin \sqrt{x/(n+1)} + \arcsin \sqrt{(x+1)/(n+1)}$$

with a variance of  $y$  within 6 % of  $1/(n+1/2)$  for almost all  $p$  and  $n$  (where  $np > 1$ ).

- For **calorimeter** data  $E$  with relative standard deviation  $\sigma(E)/E = h/\sqrt{E}$  are transformed to constant variance by

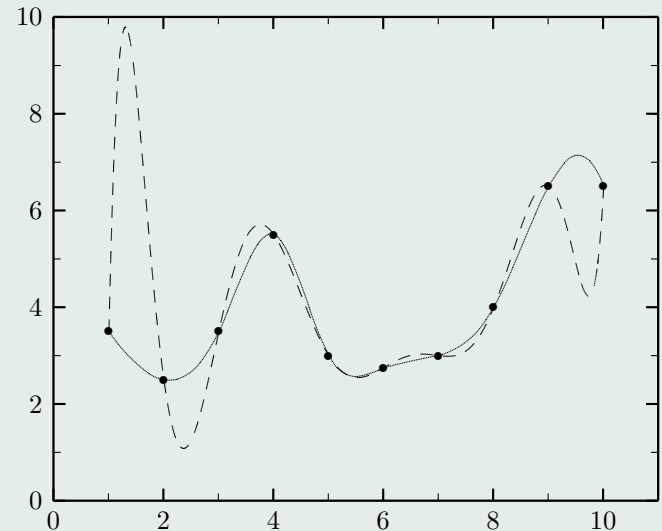
$$y = \sqrt{E} \quad \sigma_y = \frac{1}{2}h$$

## Spline functions

---

A frequent task in scientific data analysis is the interpolation of tabulated data in order to allow the accurate determination of intermediate data. If the tabulated data originate from a measurement, the values may include measurement errors, and a **smoothing interpolation** is required.

The figure shows  $n = 10$  data points at equidistant abscissa value  $x_i$ , with function values  $y_i$ ,  $i = 1, 2, \dots, 10$ . A polynomial  $P(x)$  of order  $n - 1$  can be constructed with  $P(x_i) = y_i$  (exactly), as is shown in the figure (dashed curve). As interpolating function one would expect a function without strong variations of the curvature. The figure shows that high degree polynomials tend to oscillations, which make the polynomial unacceptable for the interpolation task. The full curve is an interpolating cubic spline function.



## Cubic splines

---

Splines are usually constructed from cubic polynomials, although other orders are possible. Cubic splines  $f(x)$  interpolating a set of data  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$  with  $a \equiv x_1 < x_2 < \dots < x_{n-1} < x_n \equiv b$  are defined in  $[a, b]$  piecewise by cubic polynomials of the form

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad x_i \leq x \leq x_{i+1}$$

for each of the  $(n - 1)$  intervals  $[x_i, x_{i+1}]$  (in total  $4(n - 1) = 4n - 4$  parameters). There are  $n$  interpolation conditions  $f(x_i) = y_i$ . For the  $(n - 2)$  inner points  $2 \leq i \leq (n - 1)$  there are the continuity conditions

$$s_{i-1}(x_i) = s_i(x_i) \quad s'_{i-1}(x_i) = s'_i(x_i) \quad s''_{i-1}(x_i) = s''_i(x_i) .$$

The third derivative has a discontinuity at the inner points.

Two additional conditions are necessary to define the spline function  $f(x)$ . One possibility is to fix the values of the first or second derivative at the first and the last point. The co-called *natural* condition requires

$$f''(a) = 0 \quad f''(b) = 0 \quad \text{natural spline condition}$$

i.e. zero curvature at the first point  $a \equiv x_1$  and the last point  $b \equiv x_n$ . This condition minimizes the total curvature of the spline function:

$$\int_a^b |f''(x)|^2 dx = \text{minimum} .$$

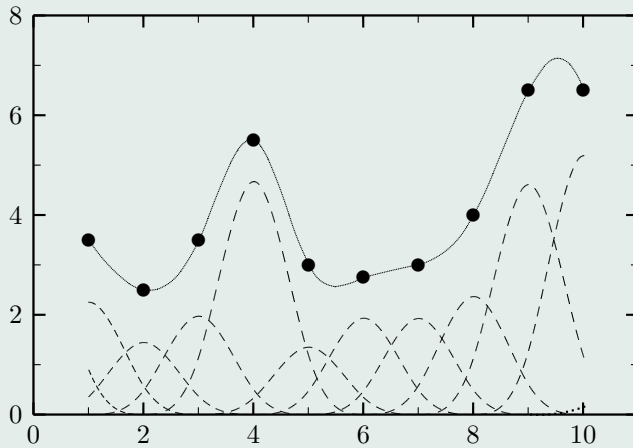
However this condition is of course not optimal for all applications. A more general condition is the so-called *not-a-knot* condition; this condition requires continuity also for the third derivative at the first and the last inner points, i.e. at  $x_2$  and  $x_{n-1}$ .

## B-splines

---

B-splines allow a different representation of spline functions, which is useful for many applications, especially for least squares fits of spline functions to data. A single B-spline of order  $k$  is a piecewise defined polynomial of order  $(k - 1)$ , which is continuous up to the  $(k - 2)$ -th derivative at the transitions. A single B-spline is non-zero only in a limited  $x$ -region, and the complete function  $f(x)$  is the sum of several *B-splines*, which are non-zero on different  $x$ -regions, weighted with coefficients  $a_i$ :

$$f(x) = \sum_i a_i B_{i,k}(x)$$



The figure illustrates the decomposition of the spline function  $f(x)$  from the previous figure into B-splines. The individual B-spline functions are shown as dashed curves; their sum is equal to the interpolating spline function.

B-splines are defined with respect to an ordered set of knots

$$t_1 < t_2 < t_3 < \dots < t_{m-1} < t_m$$

(knots may also coincide – multiple knots). The knots are also written as a knot vector  $\mathbf{T}$

$$\mathbf{T} = (t_1, t_2, t_3, \dots, t_{m-1}, t_m) \quad \text{knot vector .}$$

B-splines of order 1 are defined by

$$B_{i,1}(x) = \begin{cases} 1 & \text{for } t_i \leq x \leq t_{i+1} \\ 0 & \text{else} \end{cases} .$$

Higher order B-splines can be defined through the recurrence relation

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(x)$$

In general B-splines have the following properties:

$$\begin{aligned} B_{i,k}(x) &> 0 && \text{for } t_i \leq x \leq t_{i+k} \\ B_{i,k}(x) &= 0 && \text{else .} \end{aligned}$$

The B-splines  $B_{i,k}(x)$  is continuous at the knots  $t_i$  for the function value and the derivative up to the  $(k-2)$ -th derivative. It is nonzero (positive) only for  $t_i \leq x \leq t_{i+k}$ , i.e. for  $k$  intervals.

Furthermore B-splines in the definition above add up to 1:

$$\sum_i B_{i,k}(x) = 1 \quad \text{for } x \in [t_k, t_{n-k+1}] .$$

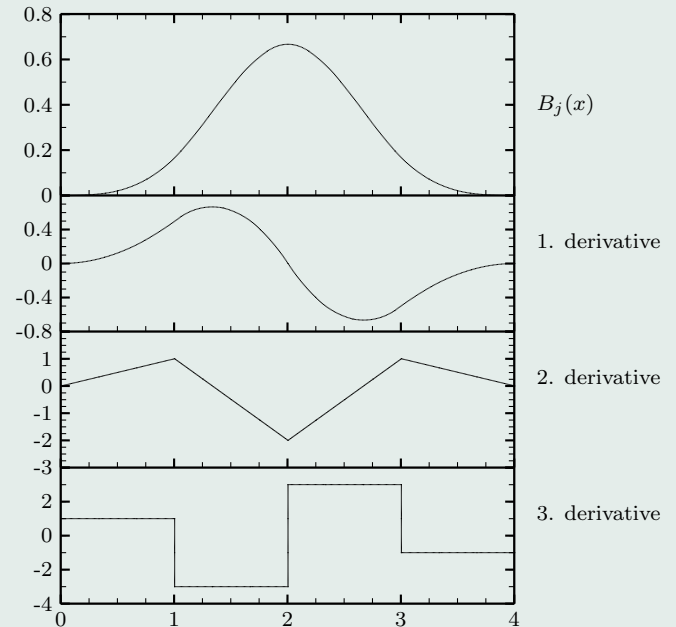
## B-splines with equidistant knots

For equidistant knots, i.e.  $t_i = i$ , *uniform* B-splines are defined and the recurrence relation is simplified to

$$B_{i,k}(x) = \frac{x-i}{k-1}B_{i,k-1}(x) + \frac{i+k-x}{k-1}B_{i+1,k-1}(x)$$

A single B-spline is shown below together with the derivatives up to the third derivative in the region of four knot intervals.

A cubic B-spline (order  $k = 4$ ) and the derivatives for an equidistant knot distance of one. The B-spline and the derivatives consist out of four pieces. The third derivative shows discontinuities at the knots.



## Least squares fits of B-splines

---

In the B-spline representation fits of spline functions to data are rather simple, because the function depends linearly on the parameters  $a_i$  (coefficients) to be determined by the fit.

The data are assumed to be of the form  $x_j, y_j, w_j$  for  $i = j, 2, \dots, n$ . The values  $x_j$  are the abscissa values for the measured value  $y_j$  with weight  $w_j$ . The weight  $w_j$  is defined as  $1/(\Delta y_j)^2$ , where  $\Delta y_j$  is one standard deviation of the measurement  $y_j$ . The least squares principle requires to minimize the expression

$$S(\mathbf{a}) = \sum_{j=1}^n w_j \left( y_j - \sum_i a_i B_i(x_j) \right)^2$$

For simplicity the B-splines are written without the  $k$ , i.e.  $B_i(x) \equiv B_{i,k}$ . The number of parameters is  $m - k$ , the parameters are combined in a  $m - k$ -vector  $\mathbf{a}$ .

The parameter vector  $\mathbf{a}$  is determined by the solution of the system of linear equations

$$\mathbf{A}\mathbf{a} = \mathbf{b} ,$$

where the elements of the vector  $\mathbf{b}$  and the square matrix  $\mathbf{A}$  are formed by the following sums:

$$b_i = \sum_{j=1}^n w_j B_i(x_j) y_j \qquad A_{ii} = \sum_{j=1}^n w_j B_i^2(x_j)$$
$$A_{ik} = \sum_{j=1}^n w_j B_i(x_j) \cdot B_k(x_j)$$

The matrix  $\mathbf{A}$  is a symmetric band matrix, with a band width of  $k$ , i.e.  $A_{ik} = 0$  for all  $i$  and  $k$  with  $|i - k| >$  spline order.

## The SPLFT algorithm

---

The SPLFT program smoothes histogram data and has the following steps:

- determination of first and last non-zero bin and transformation of data to stable variance, assuming Poisson distributed data;
- 3G53QH smoothing and piecewise fits of orthogonal polynomials (5, 7, 9, 11, 13 points) in overlapping intervals of smoothed data, enlarging/reducing intervals depending on the  $\chi^2$  value; result is weighted mean of overlapping fits;
- fourth (numerical) derivative from result and definition of knot density  $k_i$  as fourth root of abs of fourth derivative with some smoothing;
- determine number  $n_i$  of inner knots from integral of  $k_i$ , perform spline fits on original data using  $n_i - 1$ ,  $n_i$  and  $n_i + 1$  knot positions, determined to contain the same  $k_i$ -integral between knots, and repeat fit with best number;
- inner original-zero regions of length  $> 2$  are set to zero;
- backtransformation
- search for peaks in smoothed data; and fits of gaussian to peaks in original data, with improvement of knot density;
- repeat **spline fits using improved knot positions**;
- backtransformation and normalization.

## A new track-fit algorithm

---

A track-fit algorithm, based on a new principle, has been developed, which has no explicit parametrization and allows to take into account the influence of multiple scattering (“random walk”).

The fit is an attempt to *reconstruct* the *true particle trajectory* by a least squares fit of  $n$  (position) parameters, if  $n$  data points are measured.

There are three variations of the least squares fit:

- a track fit with  $n$  (position) parameters, assuming a straight line (no mean curvature);
- + determination of a curvature  $\kappa$ , i.e.  $n + 1$  parameters;
- + determination of a curvature  $\kappa$  and a time zero  $T_0$  (for drift chamber data), i.e.  $n + 2$  parameters.

The result is obtained in a single step (no iterations).

A band matrix equation (band width 3) is solved for the straight line fit, with CPU time  $\propto n$ .

More complicated matrix equations have to be solved in the other cases, but the CPU time is still  $\propto n$  (and only slightly larger than in the straight line fit).

The fit can be applied to determine the residual curvature, after e.g. a simple fixed-radius track fit.

## Multiple scattering

In the Gaussian approximation of multiple scattering the width parameter  $\theta_0$  is proportional to the square root of  $x/X_0$  and to the inverse of the momentum, where  $X_0$  is the radiation length of the medium:

$$\theta_0 = \frac{13.6 \text{ MeV}}{\beta pc} \sqrt{\frac{x}{X_0}} \left[ 1 + 0.038 \ln \frac{x}{X_0} \right].$$

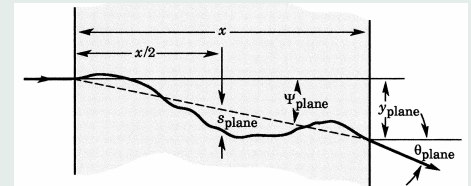
In case of a mixture of many different materials with different layers one has to find the total length  $x$  and the radiation length  $X_0$  for the combined scatterer. The radiation length of a mixture of different materials is approximated by

$$\frac{1}{X_0} = \sum_j \frac{w_j}{X_{0,j}},$$

where  $w_j$  is the fraction of the material  $j$  by weight and  $X_{0,j}$  is the radiation length in the material. The effect of multiple scattering after traversal of a medium of thickness  $x$  can be described by two parameters, e.g. the deflection angle  $\theta_{\text{plane}}$ , or short  $\theta$ , and the displacement  $y_{\text{plane}}$ , or short  $\Delta y$ , which are correlated. The covariance matrix of the two variables  $\theta$  and  $\Delta y$  is given by

$$\mathbf{V}_{\theta, \Delta y} = \begin{pmatrix} \theta_0^2 & x\theta_0^2/2 \\ x\theta_0^2/2 & x^2\theta_0^2/3 \end{pmatrix}$$

The correlation coefficient between  $\theta$  and  $\Delta y$  is  $\rho_{\theta, \Delta y} = \sqrt{3}/2 = 0.87$ .



A charged particle incident from the left in the plane of the figure traverses a medium of thickness  $x$  (from PDG).

## Reconstruction of the trajectory

---

The true trajectory coordinate at the plane with  $x_i$  is denoted by  $u_i$ . The difference between the measured and the true coordinate,  $y_i - u_i$ , has a Gaussian distribution with mean zero and variance  $\sigma_i^2$ , or

$$y_i - u_i \sim N(0, \sigma_i^2)$$

An approximation of the true trajectory is given by a polyline, connecting the points  $(x_i, u_i)$ . The angle of the line segments to the  $x$ -axis is denoted by  $\alpha$ . The  $\tan \alpha_i$  and  $\tan \alpha_{i+1}$  values of the line segments between  $x_{i-1}$  and  $x_i$ , and between  $x_i$  and  $x_{i+1}$  are given by

$$\tan \alpha_i = \frac{u_i - u_{i-1}}{x_i - x_{i-1}} \quad \tan \alpha_{i+1} = \frac{u_{i+1} - u_i}{x_{i+1} - x_i}$$

and the angle  $\beta_i$  between these two line segments is given by

$$\tan \beta_i = \tan(\alpha_{i+1} - \alpha_i) = \frac{\tan \alpha_{i+1} - \tan \alpha_i}{1 + \tan \alpha_i \cdot \tan \alpha_{i+1}}$$

The difference between the  $\tan \alpha$ -values and hence the angle  $\beta_i$  will be small, for small angles  $\tan \beta$  is well approximated by the angle  $\tan \beta$  itself and thus

$$\beta_i \approx f_i \cdot (\tan \alpha_{i+1} - \tan \alpha_i) \quad \text{with factor } f_i = \frac{1}{1 + \tan \alpha_i \cdot \tan \alpha_{i+1}}$$

is a good approximation.

The angle  $\beta_i$  between the two straight lines connecting the points at  $u_{i-1}$ ,  $u_i$  and  $u_{i+1}$  can be expressed by

$$\beta_i = \frac{\Delta y_{i+1}}{\Delta x_{i+1}} - \frac{\Delta y_i}{\Delta x_i} + \theta_i$$

and error propagation using the covariance matrix can be used to calculate the variance of  $\beta_i$ .

The mean value of the random angle  $\beta_i$  is zero and the distribution can be assumed to be a Gaussian with variance  $\theta_{\beta,i}^2$ , given by

$$\theta_{\beta,i}^2 = \frac{(\theta_{0,i-1}^2 + \theta_{0,i}^2)}{3}.$$

The correlation between adjacent  $\beta$ -values is not zero; the covariance matrix between  $\beta_i$  and  $\beta_{i+1}$  is given by

$$\mathbf{V} = \begin{pmatrix} (\theta_{0,i}^2 + \theta_{0,i+1}^2)/3 & \theta_{0,i+1}^2/6 \\ \theta_{0,i+1}^2/6 & (\theta_{0,i+1}^2 + \theta_{0,i+2}^2)/3 \end{pmatrix}$$

The correlation coefficient  $\beta_i$  and  $\beta_{i+1}$  is  $\rho = 1/4$  in the case  $\theta_{0,i} = \theta_{0,i+1} = \theta_{0,i+2}$  and becomes  $\rho = 1/2$  for the extreme case  $\theta_{0,i+1} \gg \theta_{0,i}$  and  $\theta_{0,i+1} \gg \theta_{0,i+2}$ .

The fact, that the correlation coefficient is non-zero and positive corresponds to the fact that a large scatter in a layer influence both adjacent  $\beta$ -values.



## The solution

---

The solution of a matrix equation

$$\mathbf{C}\mathbf{u} = \mathbf{r}$$

with a band matrix  $\mathbf{C}$  is considered. In a full matrix inversion band matrices lose their band structure and thus the calculation of the inverse would require a work  $\propto n^3$ . An alternative solution is the decomposition of the matrix  $\mathbf{C}$  according to

$$\mathbf{C} = \mathbf{L}\mathbf{D}\mathbf{L}^T$$

where the matrix  $\mathbf{L}$  is a left triangular matrix and  $\mathbf{D}$  is a diagonal matrix; the band structure is kept in this decomposition. The decomposition allows to rewrite the matrix equation in the form

$$\mathbf{L}(\mathbf{D}\mathbf{L}^T\mathbf{u}) = \mathbf{L}\mathbf{r} = \mathbf{b} \quad \text{with } \mathbf{v} = \mathbf{D}\mathbf{L}^T\mathbf{u}$$

and now the matrix equation can be solved in two steps:

$$\begin{array}{ll} \text{solve } \mathbf{L}\mathbf{v} = \mathbf{r} & \text{for } \mathbf{v} \text{ by forward substitution, and} \\ \text{solve } \mathbf{L}^T\mathbf{u} = \mathbf{D}^{-1}\mathbf{v} & \text{for } \mathbf{u} \text{ by backward substitution.} \end{array}$$

Thus making use of the band structure reduces the calculation of the solution to a work  $\propto n$  instead of  $\propto n^3$ . It is noted that the inverse matrix can be decomposed in the same way:

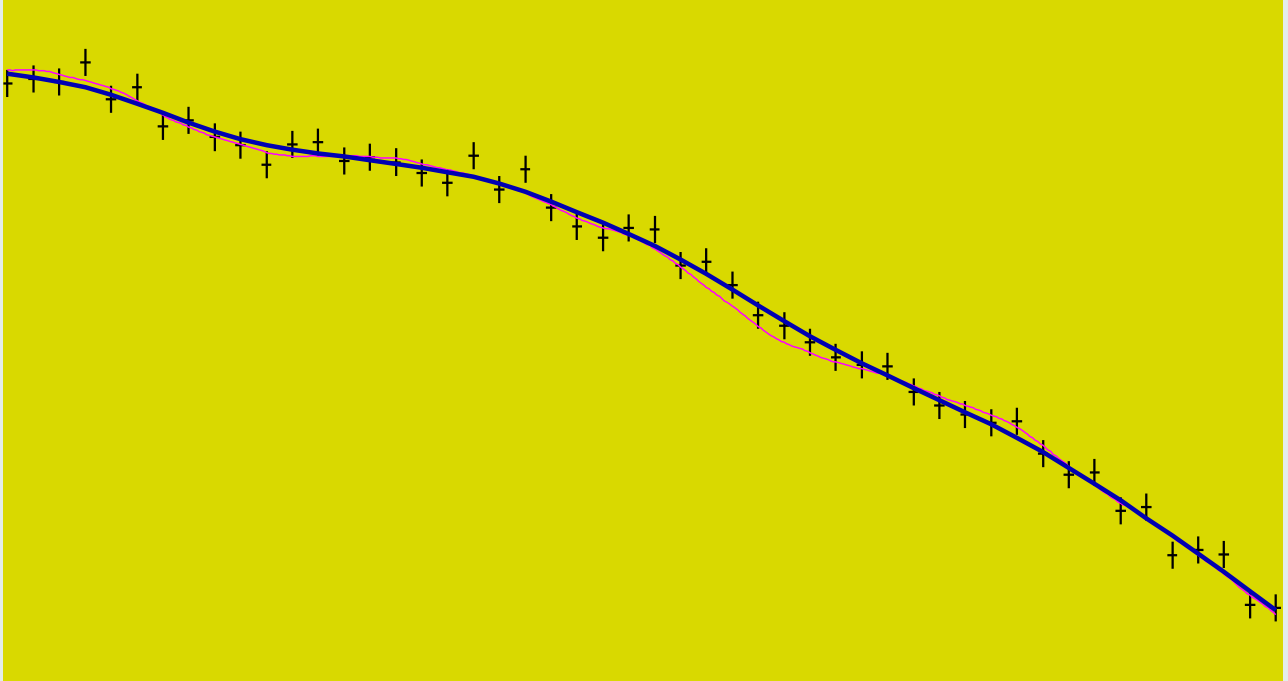
$$\mathbf{C}^{-1} = (\mathbf{L}^{-1})^T \mathbf{D}^{-1} \mathbf{L}^{-1}$$

The inverse matrix  $\mathbf{L}^{-1}$  is again a left triangular matrix.

## Fit of straight multiple-scattering tracks

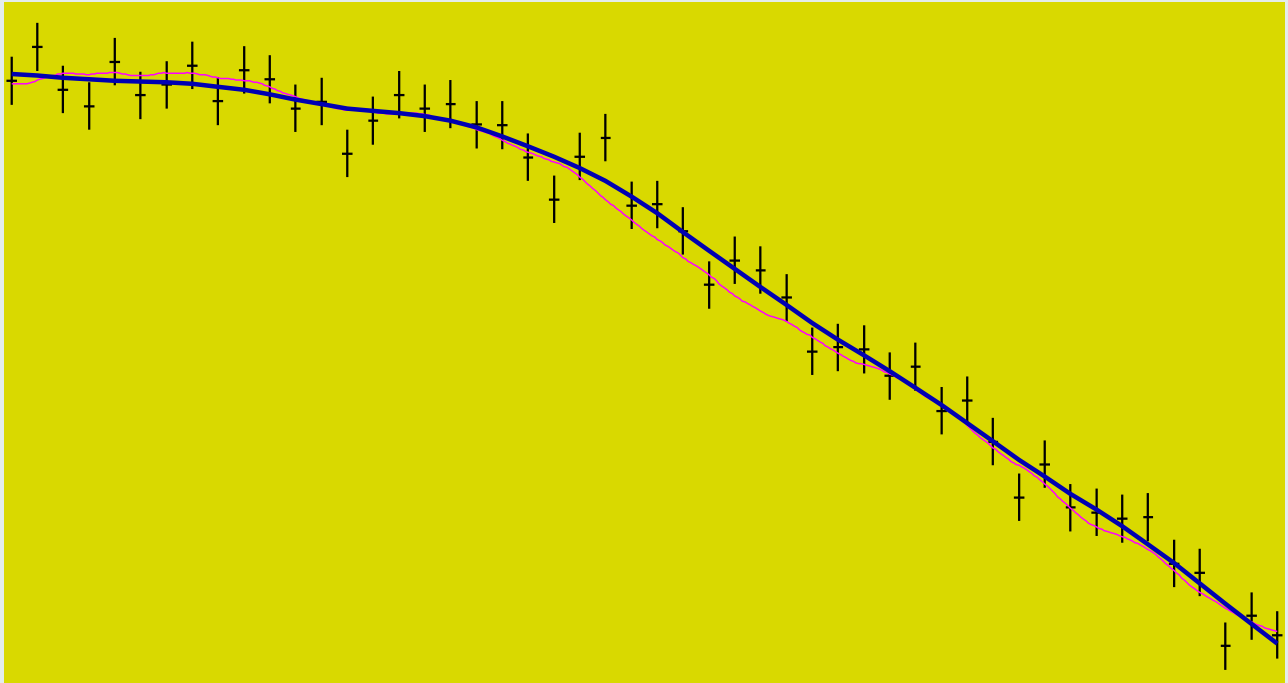
---

A simulated 50 cm long track with 50 data points of  $200\ \mu\text{m}$  measurement error (no magnetic field), fitted without curvature (vertical scale expanded by a factor  $\approx 50$ ):



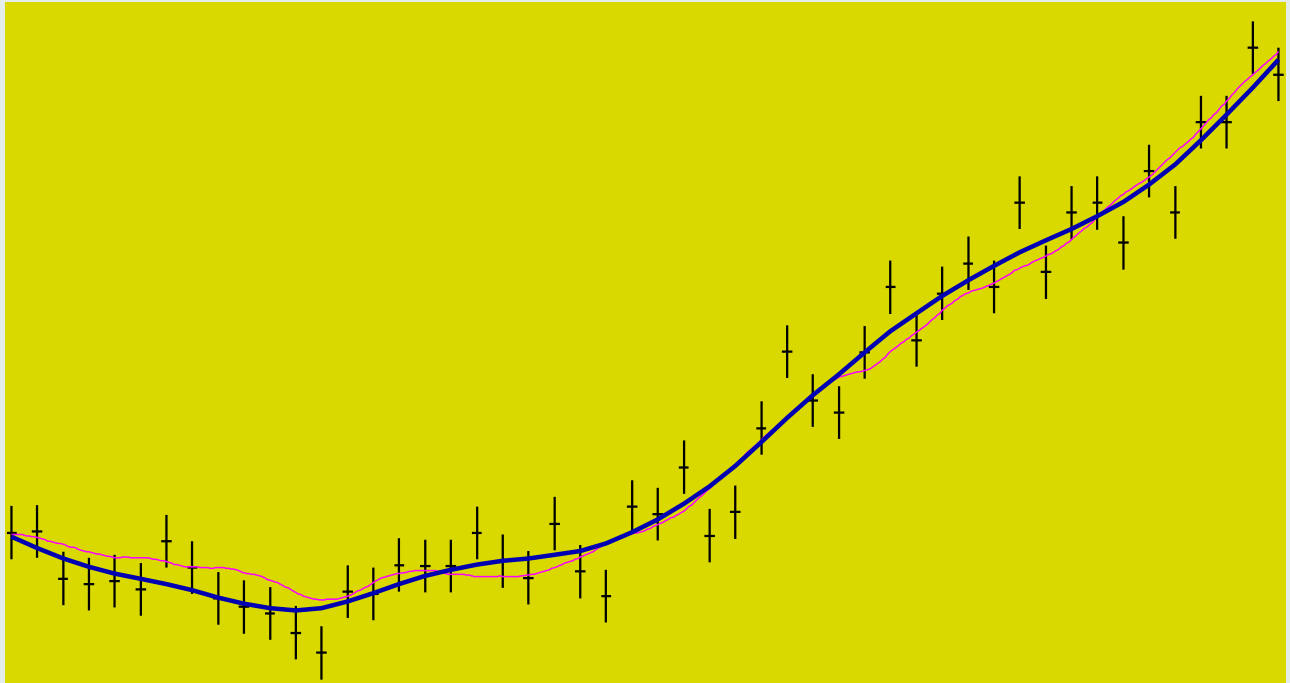
The thick blue line is the fitted line, the thin magenta line is the original particle trajectory.

A simulated 50 cm long track with 50 data points of  $200\ \mu\text{m}$  measurement error (no magnetic field), fitted without curvature:



The thick blue line is the fitted line, the thin magenta line is the original particle trajectory.

A simulated 50 cm long track with 50 data points of  $200\ \mu\text{m}$  measurement error (no magnetic field), fitted without curvature:



The thick blue line is the fitted line, the thin magenta line is the original particle trajectory.

## Curved tracks

---

The ideal trajectory of a charged particle with momentum  $p$  (in GeV/c) and charge  $e$  in a constant magnetic field  $\mathbf{B}$  is a helix, with radius of curvature  $R$  and pitch angle  $\lambda$ . The radius of curvature  $R$  and the momentum component perpendicular to the direction of the magnetic field are related by

$$p \cos \lambda = 0.3BR$$

where  $B$  is in Tesla and  $R$  is in meters. In the plane perpendicular to the direction of the magnetic field the ideal trajectory is part of a circle with radius  $R$ . Aim of the track measurement is to determine the curvature  $\kappa \equiv 1/R$ .

The formalism has now to be modified to describe the superposition of the deflection by the magnetic field and the random walk, caused by the multiple scattering.

The segments between the points  $(x_i, u_i)$  and  $(x_{i+1}, u_{i+1})$ , which were straight lines in the case on field  $\mathbf{B} = 0$ , are now curved, with curvature  $\kappa$ . The chord  $a_i$  of the segment is given by

$$\begin{aligned} a_i &= \sqrt{\Delta x_i^2 + \Delta y_i^2} \quad \text{with } \Delta x_i = x_{i+1} - x_i \quad \Delta y_i = y_{i+1} - y_i \\ &\approx \Delta x_i + 1/2 \frac{\Delta y_i^2}{\Delta x_i} \end{aligned}$$

where the latter is a good approximation for  $\Delta x_i \gg \Delta y_i$ . The angle  $\gamma_i$  between the chord and the circle at one end of the chord is given by

$$\gamma_i \approx \sin \gamma_i = a_i \kappa / 2$$

and thus in a very good approximation proportional to the curvature  $\kappa$ .

## Fit of curved multiple-scattering tracks

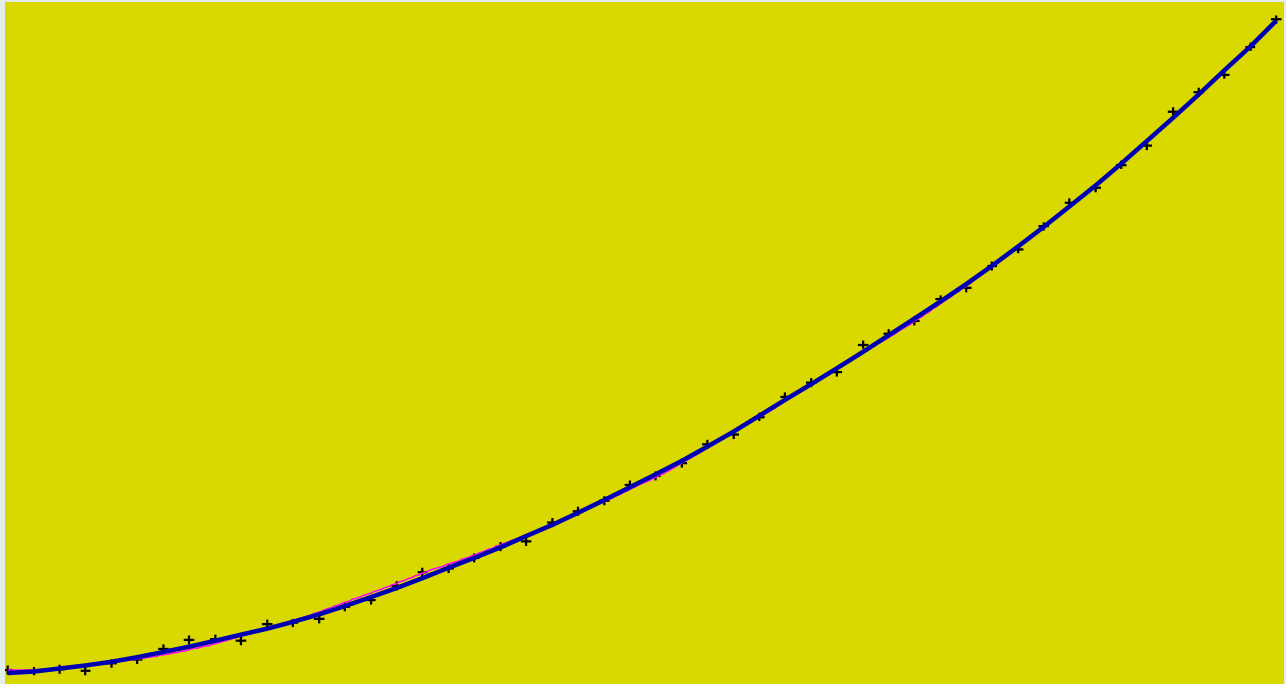
---

A simulated 50 cm long track with 50 data points of  $200\ \mu\text{m}$  measurement error in magnetic field, fitted with curvature:



The thick blue line is the fitted line, the thin magenta line is the original particle trajectory.

A simulated 50 cm long track with 50 data points of  $200\ \mu\text{m}$  measurement error in magnetic field, fitted with curvature:



The thick blue line is the fitted line, the thin magenta line is the original particle trajectory.

## Fit of curved track with kink

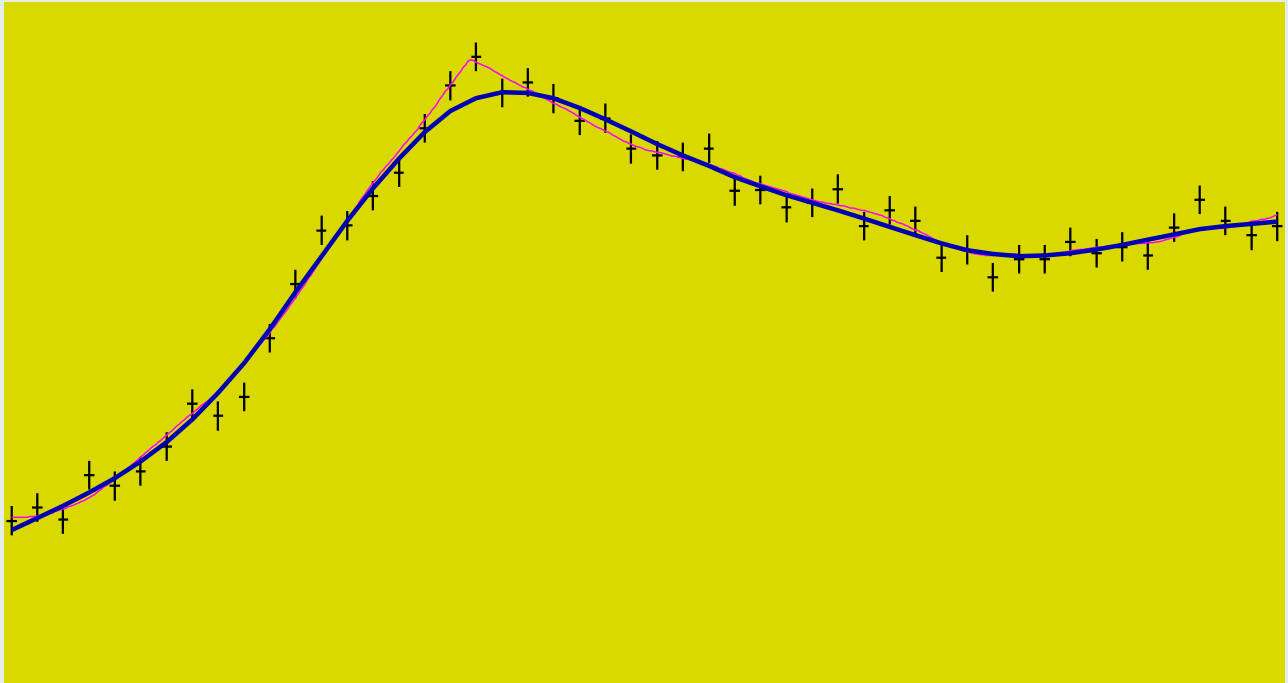
---

A simulated 50 cm long track with 50 data points of  $200\ \mu\text{m}$  measurement error in magnetic field, with a kink:



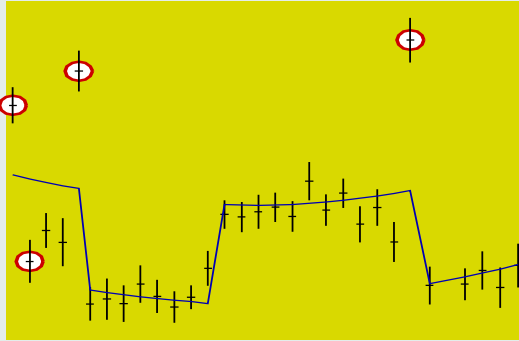
The thick blue line is the fitted line, the thin magenta line is the original particle trajectory.

A simulated 50 cm long track with 50 data points of  $200\ \mu\text{m}$  measurement error in magnetic field, with a kink:

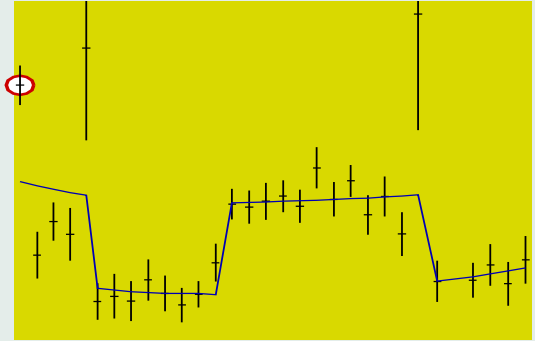


The thick blue line is the fitted line, the thin magenta line is the original particle trajectory.

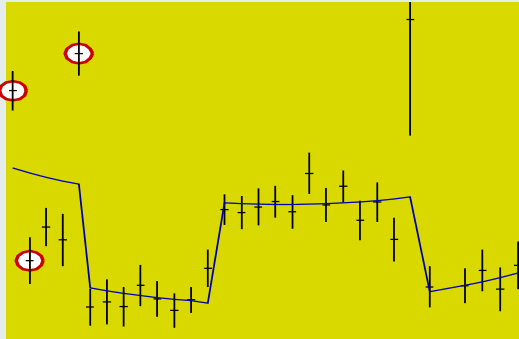
(1) Track fit with  $\kappa$  and  $T_0$  determination ...



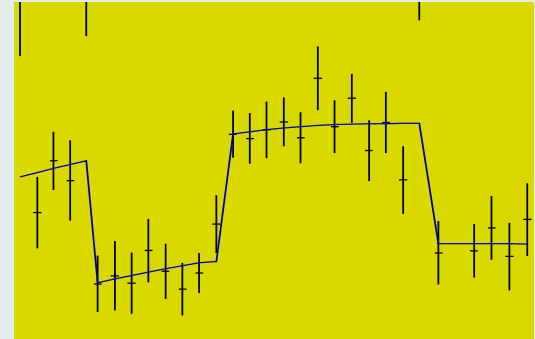
(3) ...and down-weighting of next outlier ...



(2) ...down-weighting of first outlier ...



(4) ...finally without outlier.

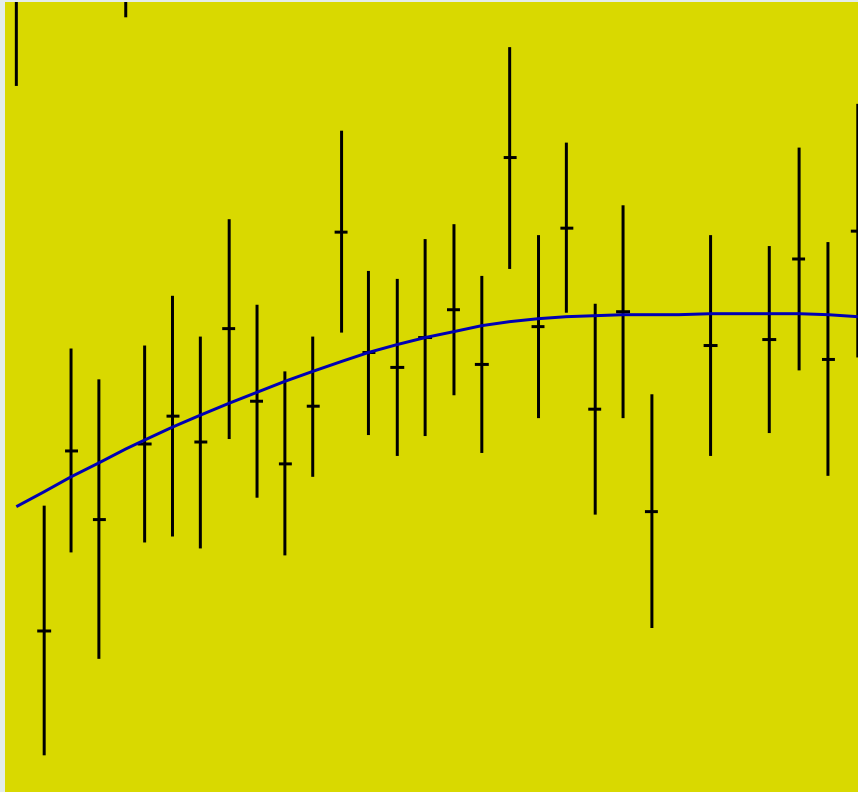


Outliers are hits with  $> 3\sigma$  deviations to fitted line. One outlier is down-weighted at a time.

## The same track fit after $T_0$ correction

---

Three outliers are removed (by down-weighting) and all hits are corrected for the  $T_0$ , determined in the previous fit:



# Smoothing

<b>Why smoothing?</b>	<b>2</b>
<b>Running median smoother</b>	<b>4</b>
Single operations . . . . .	5
Combined operations . . . . .	6
Tracking trends . . . . .	7
<b>Orthogonal polynomials</b>	<b>8</b>
Straight line fit . . . . .	9
General orthogonal polynomial . . . . .	10
Third order polynomial fit . . . . .	11
Determination of optimal order . . . . .	12
Least squares smoothing . . . . .	13
<b>Transformations</b>	<b>14</b>
<b>Spline functions</b>	<b>15</b>
Cubic splines . . . . .	16
B-splines . . . . .	17
B-splines with equidistant knots . . . . .	19
Least squares fits of B-splines . . . . .	20
<b>The SPLFT algorithm</b>	<b>21</b>
<b>A new track-fit algorithm</b>	<b>22</b>
Multiple scattering . . . . .	23
Reconstruction of the trajectory . . . . .	24
The track fit . . . . .	26
The solution . . . . .	27
Fit of straight multiple-scattering tracks . . . . .	28
Curved tracks . . . . .	31
Fit of curved multiple-scattering tracks . . . . .	32
Fit of curved track with kink . . . . .	34

The new track fit used for outlier rejection . . . . .	36
The same track fit after $T_0$ correction . . . . .	37