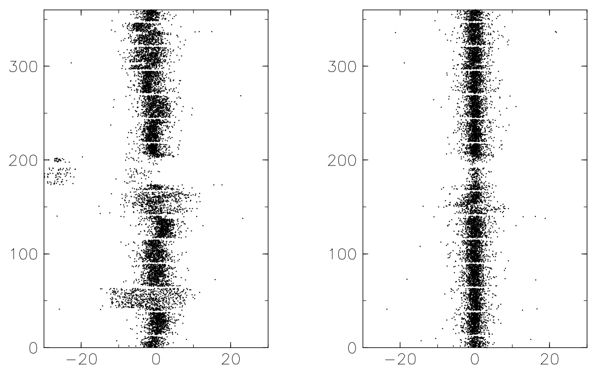
6. Example: High precision aligment of a track detector

- Motivation for alignment
- Solution by partitioning
- Simultaneous fit of global and local parameters
- Example: MC
- Example: Alignment of a central track detectors (H1)

(a) Motivation for alignment

Example: Residuals of track fit versus φ for a silicon tracker before …and after alignment:



Aim:

- smaller residuals!
- more accurate track parameters!!!

A simple method

An approximate alignment method:

- perform least squares fits on the e.g.track data *ignoring* initially the alignment parameters
- the *deviations* between the fitted and measured data, the *residuals*, are used to determine the alignment parameters afterwards.

The result of applying the alignment parameters on e.g. fits on the track data

- improved track residuals
- same track parameter values as before.

…because least squares alignment fits were based on a wrong track model.
The alignment problem is more general: in addition to pure alignment parameters there are drift velocities, variations of drift velocities, Lorentz angle …, which can not determined with residuals.

A better method?

The calculation of space coordinates from electronic signals requires alignment and many other parameters.

A better idea: perform a simultaneous least squares fit to the parameters of e.g. 20 000 tracks (each track has 50 hits and is described by three parameters) and all (e.g. 1000) alignment parameters i.e. solve normal equations $\boldsymbol{C}\boldsymbol{a} = \boldsymbol{b}$
Add all necessary constraints e.g. zero average displacement and rotation of the detector.

Now the track model is correct, but …

- $3\times$ 20 000 + 1000 parameters = 61 000 parameters
- 20 000 $\times$50 hits = 1 Mio hits

Can such a simultanenous fit be performed on a standard PC?

Constraints

Explicit relations between the parameters have to be taken into account e.g. zero average displacement and zero rotation of the whole detector.

Case of a single linear constraint between parameters:

$$\boldsymbol{f}^T \cdot \boldsymbol{a} = f_0$$

Modification of system of equations by Lagrange multiplier (λ) method (add equation $\lambda \left(\boldsymbol{f}^T \cdot \boldsymbol{a} - f_0 \right)$ to system of equations):

$$\left(\begin{array}{c|c} \boldsymbol{C} & \boldsymbol{f} \\ \hline \boldsymbol{f}^T & \boldsymbol{0} \end{array} \right) \left(\begin{array}{c} \boldsymbol{a} \\ \hline \lambda \end{array} \right) = \left(\begin{array}{c} \boldsymbol{b} \\ \hline f_0 \end{array} \right) ,$$

Each constraint adds another parameter and equation to the whole system.

(b) Solution by partitioning

Structure of the matrix equation: $\boldsymbol{C}\boldsymbol{a} = \boldsymbol{b}$

$$\text{or} \qquad \left(\begin{array}{c|c} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \hline \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{array} \right) \left(\begin{array}{c} \boldsymbol{a}_1 \\ \hline \boldsymbol{a}_2 \end{array} \right) = \left(\begin{array}{c} \boldsymbol{b}_1 \\ \hline \boldsymbol{b}_2 \end{array} \right)$$

where the submatrix $\boldsymbol{C}_{11}$ is a p-by-p square matrix and the submatrix $\boldsymbol{C}_{22}$ is a q-by-q square matrix, with $p + q = n$.

If the sub-vector $\boldsymbol{a}_1$ would not exist:

$$\boldsymbol{C}_{22}\, \boldsymbol{a}_2^* = \boldsymbol{b}_2 \qquad\qquad \boldsymbol{a}_2^* = \boldsymbol{C}_{22}^{-1}\, \boldsymbol{b}_2$$

The solution $\boldsymbol{a}_2^*$ requires only inversion of $\boldsymbol{C}_{22}$ and is called the *local* solution.

Solution by partitioning contnd.

Submatrix $\boldsymbol{B}$ of the complete inverse matrix correponding to the upper left part $\boldsymbol{C}_{11}$ is easy to calculate afterwards:

$$\boldsymbol{B} = \left(\boldsymbol{C}_{11} - \boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{12}^{T}\right)^{-1}$$

Complete inverse matrix equation in terms of $\boldsymbol{B}$:

$$\left(\begin{array}{c} \boldsymbol{a}_1 \\ \hline \boldsymbol{a}_2 \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{B} & -\boldsymbol{B}\boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1} \\ \hline -\boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{12}^{T}\boldsymbol{B} & \boldsymbol{C}_{22}^{-1} - \boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{12}^{T}\boldsymbol{B}\boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1} \end{array}\right) \left(\begin{array}{c} \boldsymbol{b}_1 \\ \hline \boldsymbol{b}_2 \end{array}\right)$$

Solution for subvector: $\boldsymbol{a}_1 = \boldsymbol{B}\boldsymbol{b}_1 - \boldsymbol{B}\boldsymbol{C}_{12}\underbrace{\boldsymbol{C}_{22}^{-1}\boldsymbol{b}_2}_{\boldsymbol{a}_2^*} = \boldsymbol{B}\left(\boldsymbol{b}_1 - \boldsymbol{C}_{12}\boldsymbol{a}_2^*\right)$

(c) Simultaneous fit of global and local parameters

Local parameters α_j are e.g. direct track parameters (belonging to a single track).

For measurement z_k with (known) constant factors δ_j:

$$z_k = \alpha_1 \cdot \delta_{1k} + \alpha_2 \cdot \delta_{2k} + \ldots \alpha_\nu \cdot \delta_{\nu k} = \sum_{j=1}^{\nu} \alpha_j \cdot \delta_{jk}$$

Normal equation of linear least squares:

$$\boldsymbol{\Gamma}\,\boldsymbol{\alpha} = \boldsymbol{\beta}\,, \qquad \text{solved by} \quad \boldsymbol{\alpha} = \boldsymbol{\Gamma}^{-1}\boldsymbol{\beta}$$

Global parameters are e.g. alignment parameters and contribute to all the measurements. They represent corrections to ideal (design) values.

$$z = \underbrace{a_1 \cdot d_1 + a_2 \cdot d_2 + \ldots a_n \cdot d_n}_{\text{global parameters}} + \underbrace{\alpha_1 \cdot \delta_1 + \alpha_2 \cdot \delta_2 + \ldots \alpha_\nu \cdot \delta_\nu}_{\text{local parameters}}$$

Complete matrix equation for global and local parameters . . .

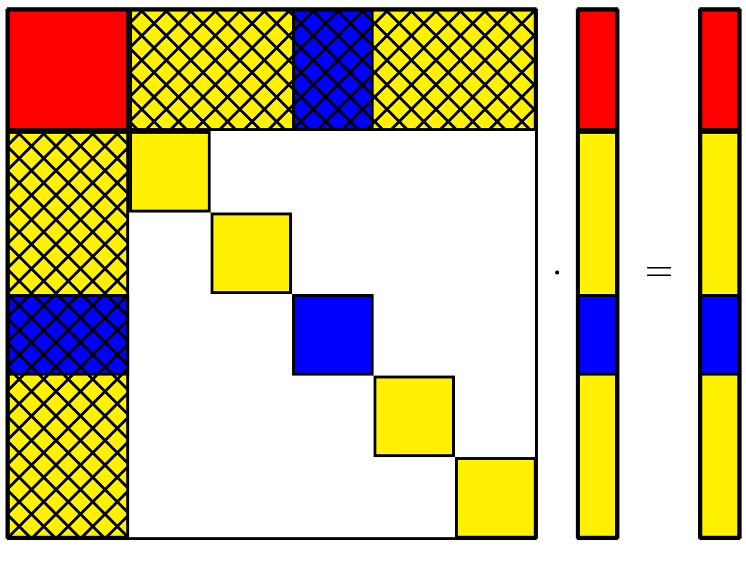
. . . has huge matrices: n-by-n matrices $\boldsymbol{C}$ for n global parameters + N of small ν-by-ν matrices $\boldsymbol{\Gamma}$.

$$\left(\begin{array}{c|c|c|c} \sum \boldsymbol{C}_i & \cdots & \boldsymbol{G}_i & \cdots \\ \hline \vdots & \ddots & 0 & 0 \\ \hline \boldsymbol{G}_i^T & 0 & \boldsymbol{\Gamma}_i & 0 \\ \hline \vdots & 0 & 0 & \ddots \end{array}\right) \cdot \left(\begin{array}{c} \boldsymbol{a} \\ \hline \vdots \\ \hline \boldsymbol{\alpha}_i \\ \hline \vdots \end{array}\right) = \left(\begin{array}{c} \sum \boldsymbol{b}_i \\ \hline \vdots \\ \hline \boldsymbol{\beta}_i \\ \hline \vdots \end{array}\right)$$

Example: 3.7×10^9 matrix elements for $n = 1000$, $\nu = 3$ and $N = 20\ 000$.

Reduction of the Matrix Size

$$\boldsymbol{C} \cdot \boldsymbol{a} = \boldsymbol{b}$$



red = global parameters yellow (and blue) = local parameters

How to solve this linear system of equations?

The matrix of normal equations has a special structure with many vanishing sub-matrices. Only the global parameters $\boldsymbol{a}$ have to be calculated.

Use N times the partitioning method for the N sets of local parameters:

$$\boldsymbol{\alpha}_i^* = \boldsymbol{\Gamma}_i^{-1} \boldsymbol{\beta}_i$$

Finally the n normal equations

$$\begin{pmatrix} & & \\ & \boldsymbol{C}' & \\ & & \end{pmatrix} \begin{pmatrix} \\ \boldsymbol{a} \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \boldsymbol{b}' \\ \\ \end{pmatrix}$$

are obtained with modified matrix and vector:

$$\boldsymbol{C}' = \sum_i \boldsymbol{C}_i - \sum_i \boldsymbol{G}_i \boldsymbol{\Gamma}_i^{-1} \boldsymbol{G}_i^T \qquad \boldsymbol{b}' = \sum_i \boldsymbol{b}_i - \sum_i \boldsymbol{G}_i \left(\boldsymbol{\Gamma}_i^{-1} \boldsymbol{\beta}_i \right)$$

The solution $\boldsymbol{a}$ is the complete solution.

Solution of final system

The solution requires no iterations. Even the case $n = 5000$ will not take much time (but: use special solution method, where undetermined parameters are set to zero).

Iterations may be necessary for other reasons, namely

- the equations depend *non-linearly* on the global parameters; the equations have to be linearized;
- the data contain outliers, which have to be removed in a sequence of cuts, becoming narrower during the iteration;
- the accuracy of the data is not known before, and has to be determined from the data (after the alignment).

Method realized in the program **Millepede**. It provides a set of subroutines for the mathematical methods and allows to adapt the method to a particular problem with introduction of linear constraints.

(d) Monte Carlo simulation

- Track detector with ten planes is simulated by Monte Carlo methods.
- Straight tracks hit the ten planes and allow to measure one coordinate z per plane transverse to the axis of the detector, with an efficiency around 90 %.
- Each plane may be displaced perpendicular to the detector axis.

The parametrization of the track in terms of the local parameters is

$$z = \alpha_1 + \alpha_2 \, x$$

Three methods are used:

- Millepede with constraints
- Millepede without constraints
- Simple residual method

Result of Millepede in alignment simulation

Δ simulated shift of detector plane

Δ_f Millepede result with constraints: no overall rotation

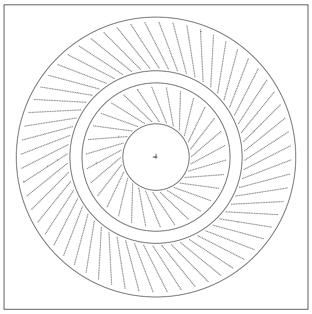
Δ_n Millepede result without constraints

Δ_r result of simple residual method

σ [cm]	Δ [cm]	Δ_f	Δ_n	Δ_r
0.0020	0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	-0.0274
0.0020	-0.0400	-0.0399 ± 0.0001	-0.0407 ± 0.0012	-0.0636
0.0300	0.1500	0.1495 ± 0.0009	0.1465 ± 0.0047	0.1269
0.0300	0.0300	0.0293 ± 0.0009	0.0252 ± 0.0062	0.0099
0.0300	-0.0750	-0.0755 ± 0.0012	-0.0806 ± 0.0078	-0.0921
0.0300	0.0450	0.0483 ± 0.0010	0.0423 ± 0.0093	0.0346
0.0300	0.0350	0.0342 ± 0.0010	0.0271 ± 0.0108	0.0216
0.0300	-0.0800	-0.0814 ± 0.0010	-0.0895 ± 0.0123	-0.0954
0.0300	0.0900	0.0902 ± 0.0011	0.0811 ± 0.0139	0.0840
0.0300	-0.0500	-0.0494 ± 0.0011	-0.0595 ± 0.0154	-0.0513

(e) Alignment of a track detector

Alignment in the $r\varphi$-plane (perpendicular to the beam) of a 56-plane drift chamber and a 2-plane silicon vertex detector in the H1 detector



Components:

- drift chamber has a length (z) of about 2 m, and extending from 20.3 cm to 84.4 cm in radius r – resolution $\sigma = 150\mu$m
- silicon vertex detector with two planes around the beampipe – resolution $\sigma = 15\mu$m

Alignment fit ...

... using ten thousands of track from *ep*-data, from cosmics with and without B-field.

Large number of correction parameters with small values:

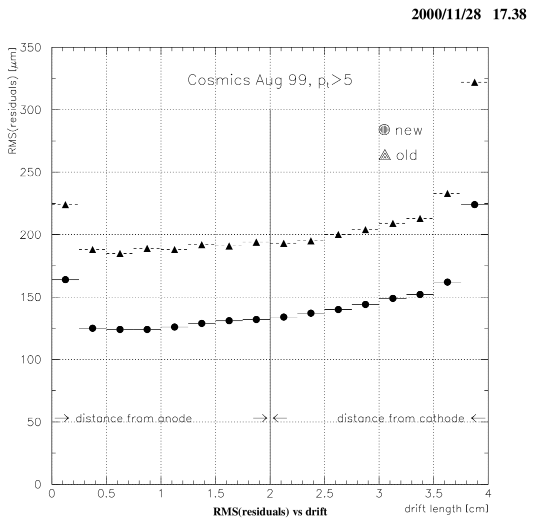
- geometrical shifts
- drift velocity and (relative) velocity corrections
- Lorentz angle corrections
- time corrections
- special correction for distortions of E field due to bad HV
- drift velocity change by electron beam current
- ...

Table of parameters and precision

row.	number	parameter	σ	unit
1	2	Δx	1	μm
2	2	$\Delta x/\Delta z_r$	2	μm
3	2	Δy	1	μm
4	2	$\Delta y/\Delta z_r$	2	μm
5	2	$\Delta\varphi$	10	μrad
6	2	$\Delta\varphi/\Delta z_r$	10	μrad
7	2	$\Delta\alpha_{\mathrm{Lor}}$	100	μrad
8	2	$\Delta v_{\mathrm{drift}-}/v_{\mathrm{drift}}$	10^{-5}	
9	2	$\Delta v_{\mathrm{drift}+}/v_{\mathrm{drift}}$	10^{-5}	
10	2	$\Delta T_0 \times v_{\mathrm{drift}}$	< 1	μm
11	2	wire staggering in wire plane	few	μm
12	2	wire staggering perp wire plane	few	μm
13	2	sagging in wire plane	few	μm
14	2	sagging perp. wire plane	few	μm
15	180	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ per cell half	few	10^{-4}
16	112	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ per layer half	few	10^{-4}
17	330	$\Delta T_0 \times v_{\mathrm{drift}}$ per group	10	μm
18	56	wire position in driftdir. per layer	10	μm
19	56	$\Delta T_0 \times v_{\mathrm{drift}}$ per layer	10	μm
20	56	wire pos. perp. driftdir. per layer	few 10	μm
21	112	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ for $I_e/50$ mA	few 10^{-4}	
22	90	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ per layer	few 10^{-4}	
23	90	Δy_W per layer	few 10	μm
24	90	Δy_W per layer^2	few 10	μm
25	64	Δ in ladder	few	μm
26	64	Δ perp. ladder	few	μm
27	64	rel. Δ in ladder (z_r)	few	μm
28	64	rel. Δ perp. ladder (z_r)	10	μm
29	64	rel. Δ perp. ladder (φ)	few	μm

Improvement of drift chamber hit resolution

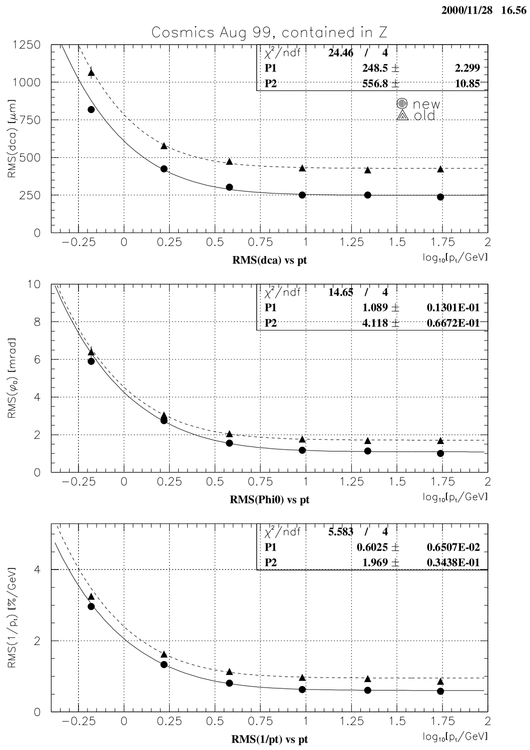
Mean track residuals from the pure drift chamber fit as function of the drift length – before and after improved alignment



$\rightarrow$ improvement from $\sigma = 180\ \mu$m to $\sigma = 125\ \mu$m

Improvement of track parameter accuracy

Standard deviation of track parameters as a function of $\log_{10} p_t/\mathrm{GeV}$:

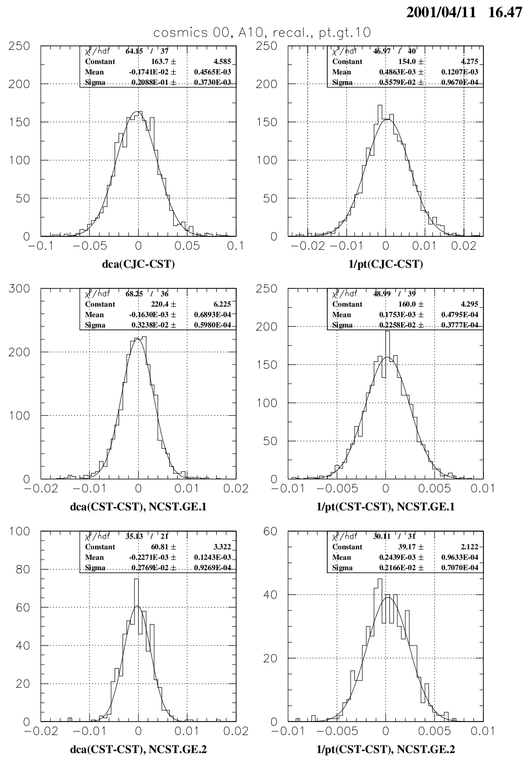


top distance of closest approach to center

middle azimuthal angle

bottom inverse transverse momentum

Difference of two parts of cosmic muon track



Residual distribution, calculated from two parts of cosmic muons

- for d_{ca} (distance of closeest approach to center) [left]
- for $1/p_t$ (inverse transverse momentum) [right]

top fit without silicon tracker (drift chamber resolution)

middle with at least 1 silicon tracker hit (accuracy due to silicon tracker)

bottom with at least 2 silicon tracker hits

Summary

Method allows accurate determination of detector parameters and can even run on a PC.

Experience shows:

- it is important to use different (all available) data sets
- it is essential to make a simultaneous alignment of all track detector components, which are also used for the measurement of tracks
- independent internal alignment of single track detector components may not lead to a good overall result
- introduce constraints or fix certain global parameters to get stable results

A warning: do not print the whole covariance matrix of the detector parameters!

A program description for **Millepede** and the code is available via `http:/www.desy.de/~blobel/`.

7. Nonlinear least squares

In practice quite often the function $f(x, \boldsymbol{a})$ for the expectation of the measured values y depends nonlinearly on the parameters a_j.

Example: $f(x, \boldsymbol{a}) = a_1 \cdot \exp(a_2 x)$ both 1. derivatives non-constant)

Linearization

$$(a+\varepsilon)^2 \approx a^2 + 2a\varepsilon \qquad (a+\varepsilon)^n \approx a^n + na^{n-1}\varepsilon$$

$$\frac{1}{a+\varepsilon} \approx \frac{1}{a} - \frac{\varepsilon}{a^2} \qquad \sqrt{a+\varepsilon} \approx \sqrt{a} + \frac{\varepsilon}{2\sqrt{a}}$$

$$e^{a+\varepsilon} \approx e^a\,(1+\varepsilon) \qquad \ln\,(a+\varepsilon) \approx \ln a + \frac{\varepsilon}{a}$$

$$\sin\,(a+\varepsilon) \approx \sin a + \varepsilon \cos a \qquad \cos\,(a+\varepsilon) \approx \cos a - \varepsilon \sin a$$

$$\tan\,(a+\varepsilon) \approx \tan a + \frac{\varepsilon}{\cos^2 a}$$

Linearization formulas, showing the *linear* change of a function for a small change of the argument.

Linearization

Linearization of the function $f(x, \boldsymbol{a})$ requires reasonable starting values for the parameters $\boldsymbol{a}$, these are denoted by $\boldsymbol{a}^*$. The function $f(x, \boldsymbol{a})$ is replaced by a (linear) Taylor expansion,

$$f(x,a) \approx f(x, \boldsymbol{a}^*) + \sum_{j=1}^{p} \frac{\partial f}{\partial a_j}(a_j - a_j^*),$$

where the partial derivatives are taken at $\boldsymbol{a}^*$,

$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{\Delta a} - \boldsymbol{f}$$

$$\boldsymbol{A} = \begin{pmatrix} \partial f(x_1)/\partial a_1 & \partial f(x_1)/\partial a_2 & \dots & \partial f(x_1)/\partial a_p \\ \partial f(x_2)/\partial a_1 & \partial f(x_2)/\partial a_2 & \dots & \partial f(x_2)/\partial a_p \\ \dots & & & \\ \partial f(x_n)/\partial a_1 & \partial f(x_n)/\partial a_2 & & \partial f(x_n)/\partial a_p) \end{pmatrix}$$

Solution of one iteration

From the linearized problem

$$S = \boldsymbol{r}^T \boldsymbol{W} \boldsymbol{r} = (\boldsymbol{y} - \boldsymbol{A}\Delta\boldsymbol{a} - \boldsymbol{f})^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{A}\Delta\boldsymbol{a} - \boldsymbol{f} -) = \min$$

the normal equations

$$(\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})\boldsymbol{\Delta a} = \boldsymbol{A}^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{f})$$

follow,

which are solved by

$$\boldsymbol{\Delta a} = (\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})^{-1} \left[\boldsymbol{A}^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{f}) \right].$$

Converging?

$$S(\boldsymbol{a}^* + \Delta\boldsymbol{a}) \leq S(\boldsymbol{a}^*) \qquad \text{is required}$$

It can be shown, that there is a certain λ, such that

$$S(\boldsymbol{a}^* + \tau\boldsymbol{\Delta a}) \quad ,$$

considered as a function of τ, is a monotonically decreasing function for $0 \leq \tau \leq \lambda$, in particular

$$S(\boldsymbol{a}^* + \lambda\Delta\boldsymbol{a}) < S(\boldsymbol{a}^*) \quad .$$

$$\left.\frac{\partial S}{\partial \tau}\right|_{\tau=0} = -2\,\Delta\boldsymbol{a}^T \left(\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}\right)\Delta\boldsymbol{a} < 0\,,,$$

(if $\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}$ is positive definite), because the expression in parentheses is a quadratic form. From continuity, there exists a $\lambda > 0$ satisfying

$$\frac{\partial S}{\partial \tau} < 0 \quad \text{for} \quad 0 \leq \tau \leq \lambda$$

Divergent iteration - what to do?

A simple method in cases, where after solution of the linearized problem the new value of S, $S(\boldsymbol{a}^* + \Delta\boldsymbol{a})$ is larger than $S(\boldsymbol{a}^*)$:

$\Delta\boldsymbol{a}$ is reduced by some factor, say 1/2, and the test is repeated, until a value smaller than $S(\boldsymbol{a}^*)$ is reached.

A better (more stable and faster) method: use optimal numerical method to minimize the one-dimensional function

$$f(\tau) = S(\boldsymbol{a}^* + \tau \cdot \Delta\boldsymbol{a})$$

w.r.t. the argument τ.

(see chapter on Function minimization.)

Recognition of convergence

The scalar product of the vectors $\Delta \boldsymbol{a}$ and $\left[\boldsymbol{A}^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{f}) \right]$,

$$\Delta S = \Delta \boldsymbol{a}^T \cdot \left[\boldsymbol{A}^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{f}) \right]$$

measures (in the linear case) the distance in S to the minimum (called EDM = estimated distance to the minimum in MINUIT).

For $\Delta S < 1$ the step $\Delta \boldsymbol{a}$ is within *one* standard deviation from the minimum; the linear approximation is usually good for this small distance, and the convergence to the minimum is fast (doubling correct digits in one iteration).

Method: assume convergence to be reached after

$$\Delta S < 0.01 \quad \text{and actual function change} < 0.01$$

(valid, independent of number of parameters).

8. Least squares with constraints

Linear or non-linear (equality) constraints:

$$\boxed{f_k(\boldsymbol{a}_{\text{true}}, \boldsymbol{y}_{\text{true}}) = 0 \qquad k = 1, 2, \ldots m}$$

with

$$\begin{aligned} \boldsymbol{y} &= n\text{-vector of measured data} \\ \boldsymbol{V}[\boldsymbol{y}] &= \text{covariance matrix of the data} \\ \boldsymbol{W} &= \boldsymbol{V}[\boldsymbol{y}]^{-1} \text{ weight matrix} \\ \boldsymbol{a} &= p\text{-vector of parameters, unmeasured} \end{aligned}$$

Least squares principle: minimize with respect to $\Delta \boldsymbol{y}$ and $\boldsymbol{a}$

$$S(\boldsymbol{a}, \boldsymbol{\Delta y}) = \Delta \boldsymbol{y}^T \boldsymbol{W} \Delta \boldsymbol{y}$$

under the conditions

$$f_k(\boldsymbol{a}, \boldsymbol{y} + \Delta \boldsymbol{y}) = 0 \quad k = 1, 2, \ldots m$$

Example: π^0 decay

Assume decay

$$\pi^0 \quad \to \quad \gamma_1 \, \gamma_2$$

with both photons measured in a calorimeter.

Simple method: use 4-vector $P_{\pi^0} = P_1 + P_2$ for the π^0 to look for resonances of the π^0 with other particles.

Better method: use constraint

$$(P_1 + P_2)^2 = m_{\pi^0}^2 = 0.135^2 \text{ GeV/c}^2$$

to improve and check the calorimeter measurement:

- 6 measured values $(E_1, \, \theta_1, \varphi_1, \, E_2, \, \theta_2, \varphi_2)$
- no unknown parameter
- 1 constraint (= one degree of freedom)

Kinematical constraints

Example: reconstruction of kinematic variables is often overconstrained – kinematical fit possible.

A different analysis style is standard at DESY:

- measurement errors are ignored,
- several formulas are used, based on subsets of data (no fit),
- comparison of alternatives (electron method, hadron method, double angle method, Σ method) by MC.

e.g.

$$y_{\text{el}} = 1 - \frac{E'_e}{2E_e} \left(1 - \cos\theta'_e\right) \qquad y_{\text{JB}} = \frac{\delta_h}{2E_e}$$

$$y_{\text{DA}} = \frac{\sin\theta'_e \left(1 - \cos\gamma_h\right)}{\sin\theta'_e + \sin\gamma_h - \sin\left(\theta'_e + \gamma_h\right)} \qquad y_{\Sigma} = \frac{\delta_h}{\delta}$$

Should there be an optimal method, which uses all measured data?

Method of Lagrange multipliers

Additional m parameters λ_k, the Lagrange multipliers, are introduced, one for each equation and a new function is defined:

$$L(\boldsymbol{a}, \Delta\boldsymbol{y}) = S(\boldsymbol{a}, \Delta\boldsymbol{y}) + 2\sum_{k=1}^{m} \lambda_k f_k(\boldsymbol{a}, \boldsymbol{y} + \Delta\boldsymbol{y})$$

The necessary condition for a local extremum of this function with respect to all parameters $\Delta\boldsymbol{y}$, $\boldsymbol{a}$ and λ is equivalent to the condition of a minimum of $S(\boldsymbol{a}, \Delta\boldsymbol{y})$ under conditions $f_k(\boldsymbol{a}, \boldsymbol{y} + \Delta\boldsymbol{y}) = 0$.

Number of unknowns

n	measured parameters and corrections Δy_i
p	parameters a_j
m	constraints, multipliers λ_k
$n+p+m$	parameters in total

Linearization

Non-linear conditions $f_k(y + \Delta y) = 0$ have to be linearized. The linear equation has to be written in terms of the correction Δy

$$\begin{aligned} f_k\left(y + \Delta y^* + (\Delta y - \Delta y^*)\right) &= 0 \\ f_k|_{y+\Delta y^*} + (\Delta y - \Delta y^*)\left.\frac{\partial f}{\partial y}\right|_{y+\Delta y^*} &= 0 \\ \left(\frac{\partial f^*}{\partial y}\right)\Delta y &= \left(\frac{\partial f^*}{\partial y}\right)\Delta y^* - f_k^* \end{aligned}$$

Previous correction is Δy^*. Function f_k^* and derivative $\partial^*/\partial y$ is evaluated at $y + \Delta y^*$, where Δy^* is previous correction.

Linearization of f_k w.r.t. the parameters $\boldsymbol{a}$ in the same way, i.e. introducing $\boldsymbol{\Delta a}$ and $\boldsymbol{\Delta a}^*$.

$\rightarrow$ same treatment of measured $(\boldsymbol{y})$ and unmeasured $(\boldsymbol{a})$ parameters.

Linearization – vector notation

$$\boldsymbol{f} + \boldsymbol{A}(\Delta\boldsymbol{a} - \Delta\boldsymbol{a}^*) + \boldsymbol{B}(\Delta\boldsymbol{y} - \Delta\boldsymbol{y}^*) = 0$$

or

$$\boxed{\boldsymbol{A}\Delta\boldsymbol{a} + \boldsymbol{B}\Delta\boldsymbol{y} = \boldsymbol{c}} \qquad \text{with} \qquad \boldsymbol{c} = \boldsymbol{A}\Delta\boldsymbol{a}^* + \boldsymbol{B}\Delta\boldsymbol{y}^* - \boldsymbol{f}$$

$$\boldsymbol{A} = \begin{pmatrix} \partial f_1/\partial a_1 & \partial f_1/\partial a_2 & \dots & \partial f_1/\partial a_p \\ \partial f_2/\partial a_1 & \partial f_2/\partial a_2 & \dots & \partial f_2/\partial a_p \\ \dots & & & \\ \partial f_m/\partial a_1 & \partial f_m/\partial a_2 & \dots & \partial f_m/\partial a_p \end{pmatrix} \qquad \boldsymbol{f} = \begin{pmatrix} f_1(\boldsymbol{a}^*, \boldsymbol{y}^*) \\ f_2(\boldsymbol{a}^*, \boldsymbol{y}^*) \\ \dots \\ f_m(\boldsymbol{a}^*, \boldsymbol{y}^*) \end{pmatrix}$$

$$\boldsymbol{B} = \begin{pmatrix} \partial f_1/\partial y_1 & \partial f_1/\partial y_2 & \dots & \partial f_1/\partial y_n \\ \partial f_2/\partial y_1 & \partial f_2/\partial y_2 & \dots & \partial f_2/\partial y_n \\ \dots & & & \\ \partial f_m/\partial y_1 & \partial f_m/\partial y_2 & \dots & \partial f_m/\partial y_n \end{pmatrix}$$

Linearization

Function L with linearization of constraints:

$$L = \Delta\boldsymbol{y}^T \boldsymbol{W} \Delta\boldsymbol{y} + 2\lambda^T (\boldsymbol{A}\Delta\boldsymbol{a} + \boldsymbol{B}\Delta\boldsymbol{y} - \boldsymbol{c})$$

The necessary conditions for an extremum are obtained by differentiation:

$$\boxed{\begin{array}{ccccccc} \boldsymbol{W}\Delta\boldsymbol{y} & & & + & \boldsymbol{B}^T\lambda & = & 0 \\ & & & & \boldsymbol{A}^T\lambda & = & 0 \\ \boldsymbol{B}\Delta\boldsymbol{y} & + & \boldsymbol{A}\Delta\boldsymbol{a} & & & = & \boldsymbol{c} \end{array}}$$

This system of coupled matrix equations (in total $n + p + m$ equations) has to be solved for the unknowns $\Delta\boldsymbol{y}$, $\Delta\boldsymbol{a}$ and λ.

Case of no unknowns

$$\boxed{\begin{array}{lcccc} \boldsymbol{W}\Delta\boldsymbol{y} & + & \boldsymbol{B}^T\lambda & = & 0 \\ \boldsymbol{B}\Delta\boldsymbol{y} & & & = & \boldsymbol{c} \end{array}}$$

Solution: multiply first equation with $\boldsymbol{W}^{-1}$

... from the left	$\Delta\boldsymbol{y} = -\boldsymbol{W}^{-1}\boldsymbol{B}^T\lambda$	$(*)$
insert into second equation	$\left(\boldsymbol{B}\boldsymbol{W}^{-1}\boldsymbol{B}^T\right)\lambda = -\boldsymbol{c}$	
solve for $\boldsymbol{\lambda}$ by	$\lambda = -\boldsymbol{W_B}\,\boldsymbol{c}$ with $\boldsymbol{W}_B = \left(\boldsymbol{B}\boldsymbol{W}^{-1}\boldsymbol{B}^T\right)^{-1}$	
insert into equation $(*)$	$\Delta\boldsymbol{y} = \left(\boldsymbol{W}^{-1}\boldsymbol{B}^T\boldsymbol{W}_B\right)\,\boldsymbol{c}$	

For *linear* problems with constant $\boldsymbol{B}$ this is the solution; for *non-linear* problems iterations are necessary.

$$\hat{\boldsymbol{y}} = \boldsymbol{y} + \Delta\boldsymbol{y} \qquad \boldsymbol{V}(\hat{\boldsymbol{y}}) = \boldsymbol{W}^{-1} - \boldsymbol{W}^{-1}\left(\boldsymbol{B}^T\boldsymbol{W}_B\boldsymbol{B}\right)\boldsymbol{W}^{-1}$$

with covariance matrix by error propagation.

Pulls = normalized corrections

On average all corrections Δy_i have the same magnitude w.r.t. their standard deviation. A check of systematic deviations is possible by looking at distributions of pulls (= normalized corrections).

Covariance matrix of the corrections $\Delta\boldsymbol{y}$:

$$\boldsymbol{V}(\Delta\boldsymbol{y}) = \boldsymbol{W}^{-1}\left(\boldsymbol{B}^T\boldsymbol{W}_B\boldsymbol{B}\right)\boldsymbol{W}^{-1} = \boldsymbol{V}(\boldsymbol{y}) - \boldsymbol{V}(\hat{\boldsymbol{y}})$$

Normalized *pull values* are obtained by

$$p_i = \frac{\Delta y_i}{\sqrt{V(\boldsymbol{y})_{ii} - V(\hat{\boldsymbol{y}})_{ii}}}$$

The pulls should follow the standardized Gaussian distribution $N(0,1)$, if the measured data are normally distributed and the conditions are linear.

The general case

System of equations to be solved in the general case:

$$\begin{pmatrix} \boldsymbol{W} & 0 & \boldsymbol{B}^T \\ 0 & 0 & \boldsymbol{A}^T \\ \boldsymbol{B} & \boldsymbol{A} & 0 \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{y} \\ \Delta \boldsymbol{a} \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \boldsymbol{c} \end{pmatrix}$$

What is the difference between *measured* and *unmeasured* parameters?

Elements of weight matrix w.r.t. to unmeasured parameters is zero!

Inverse of partitioned matrix:

$$\begin{pmatrix} \boldsymbol{W} & 0 & \boldsymbol{B}^T \\ 0 & 0 & \boldsymbol{A}^T \\ \boldsymbol{B} & \boldsymbol{A} & 0 \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{21}^T & \boldsymbol{C}_{31}^T \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} & \boldsymbol{C}_{32}^T \\ \boldsymbol{C}_{31} & \boldsymbol{C}_{32} & \boldsymbol{C}_{33} \end{pmatrix}$$

Solution

Elements of the inverse matrix:

$$\begin{aligned} \boldsymbol{C}_{11} &= \boldsymbol{W}^{-1} - \boldsymbol{W}^{-1}\boldsymbol{B}^T\boldsymbol{W}_B\boldsymbol{B}\boldsymbol{W}^{-1} \\ &\quad + \boldsymbol{W}^{-1}\boldsymbol{B}^T\boldsymbol{W}_B\boldsymbol{A}\boldsymbol{W}_A^{-1}\boldsymbol{A}^T\boldsymbol{W}_B\boldsymbol{B}\boldsymbol{W}^{-1} \\ \boldsymbol{C}_{21} &= -\boldsymbol{W}_A^{-1}\boldsymbol{A}^T\boldsymbol{W}_B\boldsymbol{B}\boldsymbol{W}^{-1} \\ \boldsymbol{C}_{22} &= \boldsymbol{W}_A^{-1} \\ \boldsymbol{C}_{31} &= \boldsymbol{W}_B\boldsymbol{B}\boldsymbol{W}^{-1} - \boldsymbol{W}_B\boldsymbol{A}\boldsymbol{W}_A^{-1}\boldsymbol{A}^T\boldsymbol{W}_B\boldsymbol{B}\boldsymbol{W}^{-1} \\ \boldsymbol{C}_{32} &= \boldsymbol{W}_B\boldsymbol{A}\boldsymbol{W}_A^{-1} \\ \boldsymbol{C}_{33} &= -\boldsymbol{W}_B + \boldsymbol{W}_B\boldsymbol{A}\boldsymbol{W}_A^{-1}\boldsymbol{A}^T\boldsymbol{W}_B \end{aligned}$$

with abbreviations

$$\begin{aligned} \boldsymbol{W}_B &= (\boldsymbol{B}\boldsymbol{W}^{-1}\boldsymbol{B}^T)^{-1} \\ \boldsymbol{W}_A^{-1} &= (\boldsymbol{A}^T\boldsymbol{W}_B\boldsymbol{A})^{-1} \end{aligned}$$

Note: the weight matrix $\boldsymbol{W}$ appears only as inverse: $\boldsymbol{W}^{-1} = V[\boldsymbol{y}] =$ covariance matrix of the measured data.

Corrections and covariance matrix

Corrections $\Delta\boldsymbol{y}$, $\Delta\boldsymbol{a}$ and the Lagrangian multipliers λ are obtained by multiplication:

$$\begin{array}{rcccl} \Delta\boldsymbol{y} & = & \boldsymbol{C}_{31}^T\,\boldsymbol{c} & = & (\boldsymbol{W}^{-1}\boldsymbol{B}^T\boldsymbol{W}_B - \boldsymbol{W}^{-1}\boldsymbol{B}^T\boldsymbol{W}_B\boldsymbol{A}\boldsymbol{W}_A^{-1}\boldsymbol{A}\boldsymbol{W}_B)\,\boldsymbol{c} \\ \Delta\boldsymbol{a} & = & \boldsymbol{C}_{32}^T\,\boldsymbol{c} & = & \boldsymbol{W}_A^{-1}\boldsymbol{A}^T\boldsymbol{W}_B\,\boldsymbol{c} \\ \lambda & = & \boldsymbol{C}_{33}\,\boldsymbol{c} & = & (-\boldsymbol{W}_B + \boldsymbol{W}_B\boldsymbol{A}\boldsymbol{W}_A^{-1}\boldsymbol{A}^T\boldsymbol{W}_B)\,\boldsymbol{c} \end{array}$$

Iteration: convergence reached if change ΔS small and e.g.

$$F = \sum_k |f_k(\boldsymbol{a}+\Delta\boldsymbol{a}, \boldsymbol{y}+\Delta\boldsymbol{y})| < \epsilon \quad ,$$

(requires same *scale* for all conditions).

Covariance matrix of the combined vector $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{a}}$ is

$$\boldsymbol{V}\begin{pmatrix} \hat{\boldsymbol{y}} \\ \hat{\boldsymbol{a}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{21}^T \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}$$

Checks

- Check pulls: if they follow $N(0,1)$, the input covariance matrix is probably correct. Otherwise *all* pulls are distorted, even if only one element of the input covariance matrix is incorrect.
- Check value of squares of deviations:

$$E(\hat{S}) = (m-p)$$

Note: the model may be incorrect!

On average S is 1.0 per degree of freedom, independent of type of the distribution, and

$$\hat{S} \quad \text{distributed} \quad \chi^2_{n-p}$$

if and only if the measured data are Gaussian distributed and the conditions are linear.

How to do such a fit in practice?

Use APLCON ("apply constraints"):

- all matrix operations hidden from the user
- matrices of derivatives calculated numerically from the constraints
- specific user code reduced to the minimum
- example: only 11 statements for π^0 fit

A program description for **Millepede** and the code is available via `http:/www.desy.de/~blobel/`.

The program code for π^0 fit, using APLCON

```
*     6 parameters and 1 constraint equations
      CALL SIMNIT(6,1,IDEB,1.E-5)
*     add four vectors
   20 EN = X(1) + X(4)
      PX = X(1)*SIN(X(2))*COS(X(3)) + X(4)*SIN(X(5))*COS(X(6))
      PY = X(1)*SIN(X(2))*SIN(X(3)) + X(4)*SIN(X(5))*SIN(X(6))
      PZ = X(1)*COS(X(2))           + X(4)*COS(X(5))
*     mass squared by square of four vector
      FM2  = EN**2 - PX**2 - PY**2 - PZ**2
*     constraint = calculated mass**2 - pi zero mass**2
      F(1) = FM2 - 0.135**2
*     call fit program
      CALL APLCON(X,VX,F,IREP)
      IF(IREP.LT.0) GOTO 20
      PMF=SQRT(PX**2+PY**2+PZ**2)
      XMF=SQRT(FM2)
```