3rd LHC Detector Alignment Workshop – CERN – 15 - 16 June 2009

Millepede II in 2009

Volker Blobel – Universität Hamburg

Abstract

The large track detectors of the LHC experiments require an accurate alignment with the determination of several 10 k parameters in order to allow to make use of the potential high spatial resolution, necessary for the physics goals. The experiment-independent MILLEPEDE program performs a simultaneous fit of (global) alignment parameters and (local) track parameters, and allows to include e.g. laser and survey data in the fit. The MILLEPEDE II version, on the web since May 2007, is improved in 2009. The 2009 version allows, in addition to equality constraints, a regularized solution in order to reduce weakly-defined modes that could distort the result. The stability of the mathematical solution methods is increased by improved pre-conditioning, line-search and outlier treatment.

---

1. Introduction

MILLEPEDE (I): Since 1998 used in H1 for Vertex det. and Central Jet Chamber; used with up to 4 800 parameters (HERAb 1999). on the web in 2000

Goal of MILLEPEDE II development: on the web 2007, 25.th May; 2009 version soon
Construct and **minimize** objective function $F(\boldsymbol{p}, \boldsymbol{q})$, which depends on the alignment corrections $\boldsymbol{p}$ and **all** track parameters $\boldsymbol{q}$ and ...

$$F(\boldsymbol{p}, \boldsymbol{q}) = \sum_{\text{data sets}} \left[\sum_{\text{events}} \left(\sum_{\text{tracks}} \left(\sum_{\text{hits}} \Delta_i^2 / \sigma_i^2 \right) \right) \right] + \sum \left[\text{terms depending on Laser data and Survey data} \right]$$

with fastest and most precise method: for up to 100 000 alignment parameters, with equality constraints, from Millions of tracks + suppl. data

- Experiment-independent
- Simultaneous fit of all alignment and local (track, Laser, ...) parameters (MILLEPEDE principle) in a single step, using large Hessian matrix,
- include detailed outlier treatment: reject or down-weight bad data (method of M-estimates).
  Note: initial deviations may be large due to misalignment!

Note: standard methods would require space $\propto n^2$ $\rightarrow$ 80 Gbyte and cpu-time $\propto n^3$ $\rightarrow$ 1 year

A reference alignment job

From talk at Ringberg 2006, June 12-16 (ATLAS meeting):
Start of development in May 2005 after discussions with Hamburg cms group, with aim:

- alignment with up to 100 000 parameters in a reasonable time on a standard PC;

What is a reasonable time? ... hours ...

Data for MILLEPEDE alignment using cms alignment files, shown with this colour:

1 596 489	**track**
47 655	**alignment parameters**
8	**constraint equations**
5	**parameters/track**
$\leq$ 54	**hits/track**
$\leq$ 166	**global parameters/track**

with outlier down-weighting in 3 iterations, requiring $3 \times 1\,596\,489$ local fits per step, and solution of matrix equation with $47\,661 \times 47\,661$ symmetric matrix C using MINRES.

... without regularisation		**... with regularisation and pre-sigmas**	
1 hour 54 min	**total time**	**1 hour 15 min**	**total time**
471 int. iterations	**solution (MINRES)**	**112 int. iterations**	**solution (MINRES)**
12 minutes	**of matrix equation**	**3 minutes**	**of matrix equation**

2. The MILLEPEDE algorithm

Optimal solution in the least squares sense of the equation $\boldsymbol{J}\boldsymbol{x} \cong \boldsymbol{z}$, with (small) corrections $\boldsymbol{x}$ to alignment and track parameters and residuals $\boldsymbol{z}$ from measured data $\boldsymbol{y}$, is defined by the requirement

$$F(\boldsymbol{x}) = \|\boldsymbol{J}\boldsymbol{x} - \boldsymbol{z}\|^2 = \text{minimum}$$

From $\partial F/\partial \boldsymbol{x} = 0$ (with covariance matrix $\boldsymbol{V}_z$ of the data):

$$\text{Matrix equation:} \quad \left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{V}_y^{-1}\boldsymbol{J}\right)\, \boldsymbol{x} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{V}_y^{-1}\boldsymbol{z} \qquad \text{to be solved}$$
$$\boldsymbol{C}\, \boldsymbol{x} = \boldsymbol{b}$$

The solution vector $\boldsymbol{x}$ is a linear transformation of the residual vector $\boldsymbol{z}$:

$$\boldsymbol{x} = \boldsymbol{C}^{-1}\, \boldsymbol{b} = \left[\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{V}_y^{-1}\boldsymbol{J}\right)^{-1}\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{V}_y^{-1}\right)\right]\, \boldsymbol{z}$$

Solution by inversion would be limited to small matrix dimensions n , because of cpu-time $\propto n^3$.

Practicable method for the solution of the matrix equation

- **space:** make use of sparse structure of matrix: space requirement $\ll n^2/2$;
- **cpu-time:** make use of fast solution methods e.g. based of Krylov sequence like MINRES.

How to use MILLEPEDE

Input = sets of single measured data points from local fits (track, cosmics, halo muons, survey data, Laser data ...)

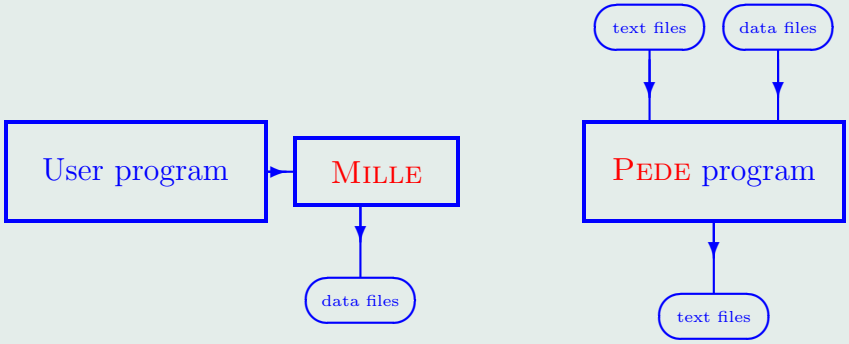
$$y_i = \underbrace{f(x_i, \boldsymbol{q}, \boldsymbol{p})}_{\text{fit function}} + \underbrace{\sum_{j=1}^{\nu} \left(\frac{\partial f}{\partial q_j}\right) \Delta q_j}_{\text{local derivatives}} + \underbrace{\sum_{\ell \in \Omega} \left(\frac{\partial f}{\partial p_\ell}\right) \Delta p_\ell}_{\text{global derivatives}} + \epsilon \qquad \epsilon \in N(0, \sigma_i^2);\ \ell = \underbrace{\text{parameter label}}_{\text{positive integer}}$$

Derivatives express the change of residual $z_i = y_i - f(x_i, \boldsymbol{q}, \boldsymbol{p})$, if q_j or p_ℓ is changed by Δq_j or Δp_ℓ.

The Jacobian $\boldsymbol{J}$ constructed from the (local and global) derivatives, and the correction vector $\boldsymbol{x}$ includes all corrections Δq_j or Δp_ℓ of local and global parameters.

(1) File with $z_i = y_i - f(x_i, \boldsymbol{q}, \boldsymbol{p})$, σ_i and all derivatives written within user program by MILLE.

Allows to repeat local fit (only last iteration) in PEDE.



(2) Data files are processed in stand-alone program PEDE, steered by text files.

The sparse matrix $\boldsymbol{C}$ of a simultaneous fit of alignment parameters (global) and track parameters (local) is a large matrix, that can be reduced to a smaller matrix for the alignment parameters only using Schur complements (no approximation!).

The matrix C^{total}, a 8 030 100 × 8 030 100 matrix (several 100 Tera Bytes) . . .

$$\boldsymbol{C}^{\text{total}} = \left(\begin{array}{cccccccccc}
x & & & & & x\,x\,x & & & x\,x\,x & & \cdots \\
 & x & & & & x\,x\,x & & & & & \cdots \\
 & & x & & & & & & x\,x\,x & & \cdots \\
 & & & x & & & x\,x\,x & & & x\,x\,x & \cdots \\
 & & & & x & & x\,x\,x & & & & \cdots \\
 & & & & & x\,x\,x & & & & & \cdots \\
 & & & & & & x\,x\,x & & & & \cdots \\
 & & & & & & & & & x\,x\,x & \cdots \\
 & & & & & & & & & x\,x\,x & \cdots \\
\hline
x & & x & & & x\,x\,x & & & & & \\
x & & x & & & x\,x\,x & & & & & \\
x & & x & & & x\,x\,x & & & & & \\
\hline
 & x\,x & & x & & & x\,x\,x & & & & \\
 & x\,x & & x & & & x\,x\,x & & & & \\
 & x\,x & & x & & & x\,x\,x & & & & \\
\hline
x & x & & & & & & x\,x\,x & & & \\
x & x & & & & & & x\,x\,x & & & \\
x & x & & & & & & x\,x\,x & & & \\
\hline
 & x & & & x\,x & & & & x\,x\,x & & \\
 & x & & & x\,x & & & & x\,x\,x & & \\
 & x & & & x\,x & & & & x\,x\,x & & \\
\hline
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \ddots
\end{array}\right)$$

$$\longrightarrow \quad \boldsymbol{C}^{\text{global}} = \left(\begin{array}{ccccccc}
x & & x & & & & \\
 & x & & & x & & \\
x & & x & & & & \\
 & & x\,x & & x\,x & x & \\
 & & x\,x & & x & & \\
 & x & & x & & & \\
 & & x\,x & & x & & \\
 & & x & & & x\,x & \\
 & & x & & & x\,x &
\end{array}\right)$$

Element $(\boldsymbol{C}^{\text{global}})_{jk} \neq 0$, if parameters j and k in same local fit.

Note: the inverse of a sparse matrix (= covariance matrix) is dense; **all parameters are correlated!**

. . . is reduced to a (sparse) 47 655 × 47 655 matrix C^{global} for the global parameters.

Optimization

Ideally one NEWTON step $\boldsymbol{x}$ from $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{b}$ is sufficient to reach the minimum, but

- solution not perfect due to round-off error in high dimension (large condition number);
- outlier treatment introduces a non-linearity.

Iteration: solve $\boldsymbol{C}\boldsymbol{d}_k = \boldsymbol{b}_k$ for $\boldsymbol{d}_k$ and line-search $\Phi(\alpha) = F(\boldsymbol{x}_k + \alpha \cdot \boldsymbol{d}_k) = \min$ w.r.t. α, with $\boldsymbol{x}_k$ and $\boldsymbol{x}_k + \boldsymbol{d}_k$ forced to satisfy the constraints with high-precision.

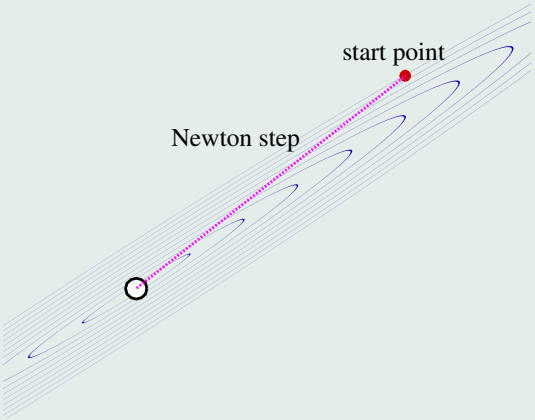
$$s = \boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{b}_k = \boldsymbol{b}_k^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{b}_k = \boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{C}\boldsymbol{d}_k \qquad \text{(EDM)}$$

$$\Phi(\alpha) \approx F_k - \alpha s + \tfrac{1}{2}\alpha^2 s$$

Sometimes large angle ($\approx 90°$) between Newton direction $\mathbf{d}_k$ and $\boldsymbol{b}_k$ (the direction of steepest descent).
Two-dim. example ($\Longrightarrow$) with

- large eigenvalue (small half-axis of ellipse), and
- small eigenvalue (large half-axis of ellipse) – linear combination of the two parameters corresponds to weakly defined mode.

$F(\boldsymbol{x})$ with large correlation



3. Weak modes I . . . and constraints

Alignment of HEP track detectors . . . based **only** on track residual minimization: *incomplete* data!

Certain linear combinations of alignment parameters (degrees of freedom) are

- **undefined:** [eigenvalues zero]
  a general linear transformation ($3+9$ parameters) will (almost) not affect the χ^2 of the local fits (3 translation-, 3 rotation-, 3 scaling-, 3 shearing-parameters)
- **weakly defined:** [eigenvalues small]
  certain non-linear deformations.

and the detector alignment is distorted.

These problems are visible in the **eigenvalue spectrum** of the matrix $\boldsymbol{C}^{\text{global}}$.

The undefined/weakly defined degrees of freedom can be

- **fixed** by equality constraints, and
- **improved** by data mixtures (tracks + survey information + Laser data + . . .).

$\Rightarrow$ diagonalization of matrix $\boldsymbol{C}^{\text{global}}$

Solution by diagonalization

$$\boldsymbol{C} = \boldsymbol{U}\,\boldsymbol{\Lambda}\,\boldsymbol{U}^{\mathrm{T}} \qquad \text{Diagonalization of symmetric matrix}$$

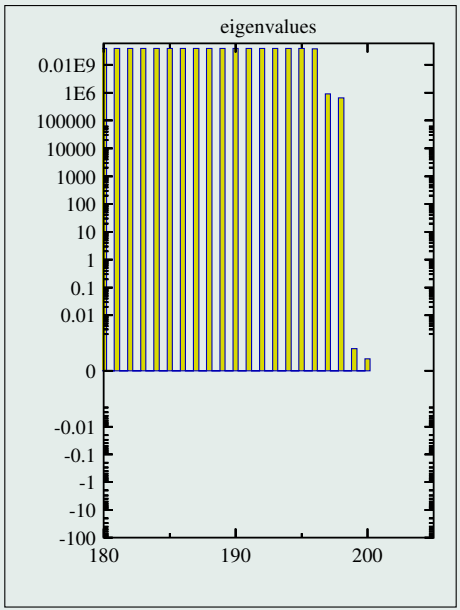
with $\boldsymbol{\Lambda}$ diagonal, $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$ with $\boldsymbol{U}\,\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}^{\mathrm{T}}\,\boldsymbol{U} = \mathbf{1}$. $\qquad \boldsymbol{C}^{-1} = \boldsymbol{U}\,\boldsymbol{\Lambda}^{-1}\,\boldsymbol{U}^{\mathrm{T}}$

$$\text{eigenvalue ordering in } \boldsymbol{\Lambda} = [\mathrm{diag}\,(\lambda_i)]: \quad \lambda_1 \;\geq\; \ldots \geq \lambda_k \geq \lambda_{k+1} \geq \ldots \lambda_n$$

The solution can be expressed by Fourier coefficients c_j (with covariance matrix $\boldsymbol{V}_c = \mathbf{1}$):

$$\boldsymbol{x} = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} c_j \boldsymbol{u}_j \qquad \text{with} \quad c_j = \frac{1}{\sqrt{\lambda_j}} \left(\boldsymbol{b}^{\mathrm{T}} \boldsymbol{u}_j\right)$$

Fourier coefficients c_j, which are insignificant (i.e. compatible with zero), should follow a normal distribution $N(0,1)$; they can make a large contribution to the solution $\boldsymbol{x}$, if the eigenvalues λ_j are small.



Simple one-dimensional alignment example ($\Rightarrow$) **without equality constraints**: $\rightsquigarrow$ 2 eigenvalue $\approx$ 0 and matrix $\boldsymbol{C}^{\mathrm{global}}$ singular!

Minimization with equality constraints

Constraint equations for m linear (**equality**) constraints described by $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$ ($\boldsymbol{A}$ has m rows)
Task: minimize $F(\boldsymbol{x})$ subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$ instead of minimize $F(\boldsymbol{x})$

Constraint equations e.g.: $\sum \Delta x_\ell = 0$; $\sum \Delta y_\ell = 0$; $\sum \Delta z_\ell = 0$; no overall rotation ...

Calculation of $\boldsymbol{x}$ with Lagrange method: introduce m multipliers $\boldsymbol{\lambda}$

$$\mathcal{L}(\boldsymbol{x}) = F(\boldsymbol{x}) + \boldsymbol{\lambda}^{\mathrm{T}} (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{c})$$

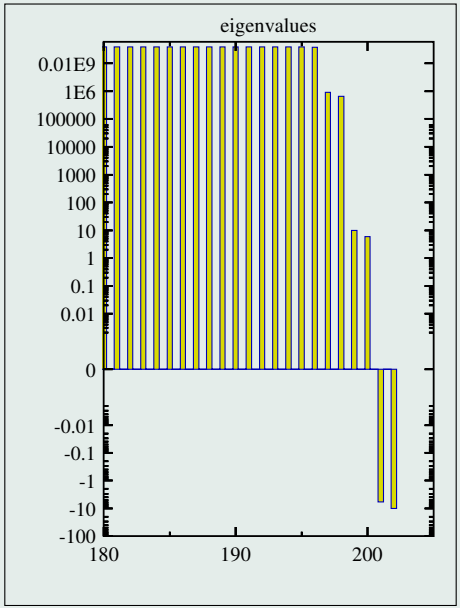
From $\partial\mathcal{L}/\partial\boldsymbol{x} = 0$ and $\partial\mathcal{L}/\partial\boldsymbol{\lambda} = 0$:

$$\boldsymbol{C} \times \begin{pmatrix} \boldsymbol{x} \\ \hline \boldsymbol{\lambda} \end{pmatrix} = \left(\begin{array}{c|c} \boldsymbol{C}^{\mathrm{global}} & \boldsymbol{A}^{\mathrm{T}} \\ \hline \boldsymbol{A} & 0 \end{array}\right) \times \begin{pmatrix} \boldsymbol{x} \\ \hline \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ \hline \boldsymbol{c} \end{pmatrix}$$

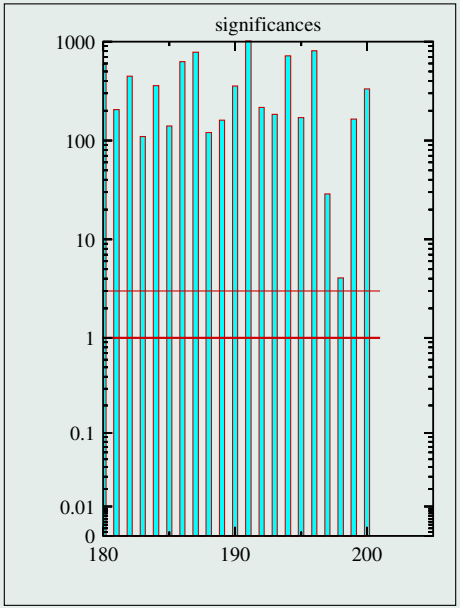
Unique solution (for sufficient constraints), matrix $\boldsymbol{C}$ not positive definite.

Simple one-dimensional alignment example ($\Rightarrow$) **with 2 equality constraints**: $\rightsquigarrow$ larger positive eigenvalues and 2 negative eigenvalues!

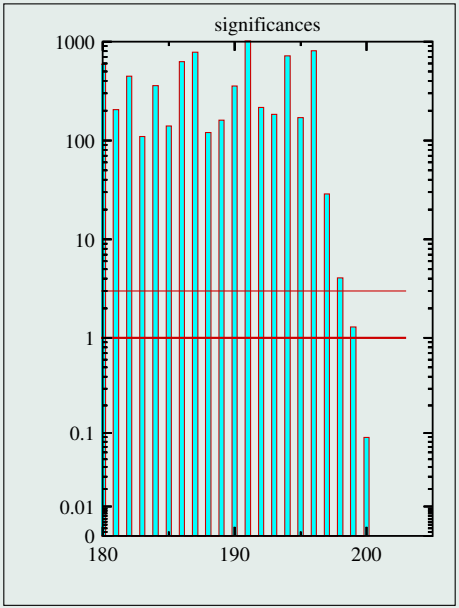
Constraints **essential** for acceptable solution.



Significances (standard deviations = 1, independent) can be defined by $|c_j|$



without constraints



with constraints

Remember: $$\boldsymbol{x} = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} c_j \boldsymbol{u}_j$$

Contribution to solution $\boldsymbol{x}$ corresponding to very small eigenvalues is c_j, divided by $1/\sqrt{\text{eigenvalue}}$.

Pre-sigmas

Small (positive) eigenvalues of matrix $\boldsymbol{C}$ correspond to a large condition number ("ill-conditioning") with an unstable solution!

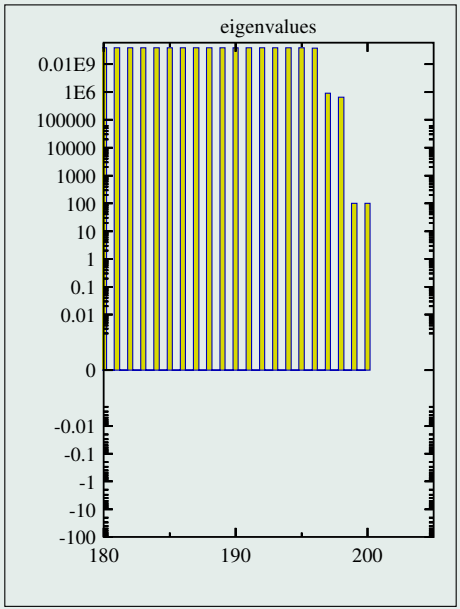
Proposal by Levenberg (1944) and Marquardt (1963):

> add to $\boldsymbol{C}^{\text{global}}$ a "suitable chosen" multiple of the unit matrix, that stabilizes the matrix, but requires more iterations (to get the same solution).*

MILLEPEDE (I and II): define, for each parameter ℓ, an a-priori accuracy called pre-sigma σ_ℓ^{pre}, which defines a stabilizing diagonal matrix $\boldsymbol{D}^{\text{pre}}$:

$$(\boldsymbol{D}^{\text{pre}})_{\ell\ell} = \frac{1}{(\sigma_\ell^{\text{pre}})^2}$$

$$\left(\boldsymbol{C}^{\text{global}}\right)\ \boldsymbol{x} = \boldsymbol{b} \qquad \Rightarrow \qquad \left(\boldsymbol{C}^{\text{global}} + \boldsymbol{D}^{\text{pre}}\right)\ \boldsymbol{x} = \boldsymbol{b}$$



Simple one-dimensional alignment example ($\Rightarrow$) with pre-sigma $\sigma^{\text{pre}} = 0.1$, without constraints:
$\rightsquigarrow$ small (and all) eigenvalues increased by $1/0.1^2 = 100$.

Constraints still **essential** for acceptable solution.

* V. B., MILFIT: A Program to Find Maxima of Likelihood Functions, Report DESY 71/18

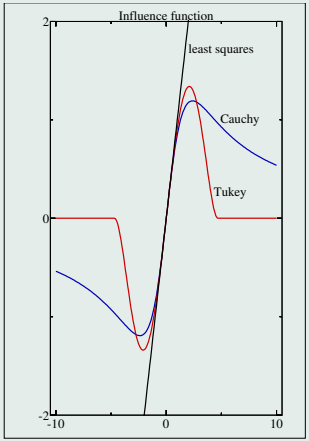
4. Outliers and line-search

Least-squares: influence of residual z is proportional to its size $|z|$. The presence of outliers in the data will deteriorate the alignment result.

Initial mis-alignment can fake outliers – no initial rejects allowed, but in MILLEPEDE *immediately after the first step, which usually makes a huge improvement.*

M-estimates. The **square** is replaced in M-estimates by a dependence with reduced influence for larger residuals (used in local fits).

$z =$ residual/std. deviation	influence function $\psi(z) = \mathrm{d}\rho(z)/\mathrm{d}z$	add. weight $\omega(z) = \psi(z)/z$
Least squares	$= z$	$= 1$
Cauchy($c = 2.3849$)	$= \dfrac{z}{1 + (z/c)^2}$	$= \dfrac{1}{1 + (z/c)^2}$
Huber $\begin{cases} \text{if } \lvert z\rvert \le c = 1.345 \\ \text{if } \lvert z\rvert > c = 1.345 \end{cases}$	$= \begin{cases} z \\ c \cdot \mathrm{sign}\,(z) \end{cases}$	$= \begin{cases} 1 \\ c/\lvert z\rvert \end{cases}$



Initially only fits with huge χ^2 rejected, later χ^2-cut is reduced to value corresponding to 3σ (should correspond to $\ll 0.17\%$ of the cases; large %-values indicate problems in e.g. track parametrization).

Line search and slopes

Original line-search requirement $\Phi(\alpha) = \min$ difficult, because $\Phi(\alpha)$ is discontinous due to outlier-downweighting and -rejection.

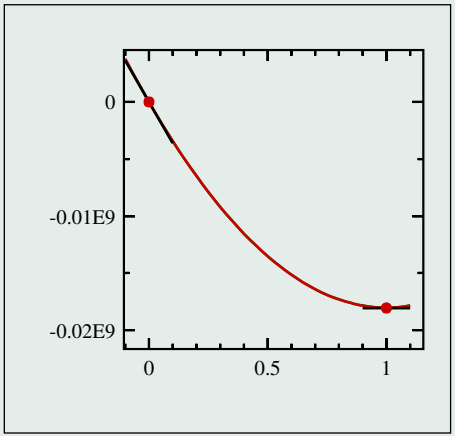
$$s = \boldsymbol{d}_k^{\mathrm{T}} \boldsymbol{b}_k = \boldsymbol{b}_k^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{b}_k = \boldsymbol{d}_k^{\mathrm{T}} \boldsymbol{C} \boldsymbol{d}_k \qquad \text{(called EDM in MINUIT)}$$

$$\Phi(\alpha) \approx F_k - \alpha s + \tfrac{1}{2}\alpha^2 s \qquad (\approx \text{parabolic})$$

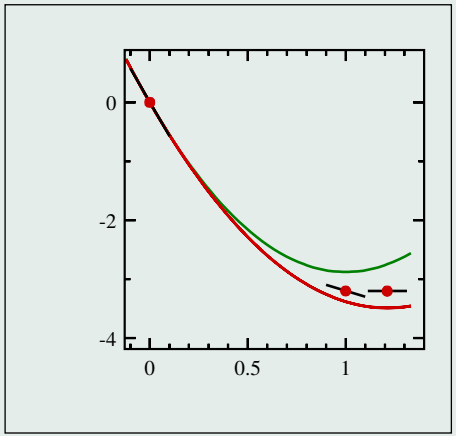
Replaced in MILLEPEDE 2009 by line-search slope requirement

$$\boxed{\Phi'(\alpha) = 0 \ ,}$$

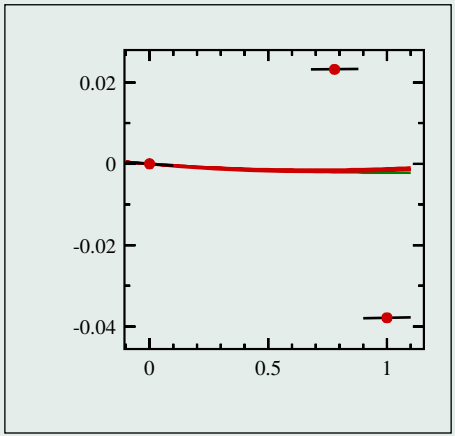
insensitive to outliers. Eventually, in difficult cases, re-definition of matrix $\boldsymbol{C}$ after outlier treatment.



Parabolic behaviour for large ΔF.



Improvement for medium ΔF.



Large fluctuations for small ΔF.

5. Solution of large systems

$$\boldsymbol{C}\,\boldsymbol{x} = \boldsymbol{b}$$

$\boldsymbol{C}$ is **symmetric and indefinite** (i.e. not positive-definite), very large and sparse.
Standard method in MILLEPEDE 2009: MINRES*) = Minimal Residual, for symmetric matrix.
Iterative: the iterate $\boldsymbol{x}_k$ minimizes

$$\|\boldsymbol{C}\,\boldsymbol{x} - \boldsymbol{b}\|_2 \qquad \text{over the set} \qquad \mathcal{S}_k = \boldsymbol{x}_0 + \text{span}\{\boldsymbol{r}_0, \boldsymbol{C}\boldsymbol{r}_0, \ldots, \boldsymbol{C}^{k-1}\boldsymbol{r}_0\}\ ,$$

where $\boldsymbol{r}_0$ = initial residual $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{C}\boldsymbol{x}_0$ (Krylov subspace, with orthogonalization).
Algorithm needs only product $\boldsymbol{C}\times$ vector (see next page); convergence speed

- depends on eigenvalue spectrum (improved by pre-sigma/regularization);
- convergence is accelerated by *preconditioning* (see after next page).

solution takes 12 or 3 min (factor $\approx 10^4$ faster than inversion) for 47 661 parameters, depending on use of pre-sigmas/regularization.
$\rightsquigarrow$ cpu-time is not a problem!

MINRES is special form of GMRES-Algorithm **) (= Generalized Minimal Residual) for non-symmetric matrices, which requires more memory space (and a restart procedure)
Other methods for large matrix equations: GMRES, BiCG, CGS, BICGSTAB, TFQMR, QMRCGSTAB, ...

*) C. C. Paige and M. A. Saunders (1975), Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12(4), pp. 617-629.

`www.stanford.edu/group/SOL/software/minres.html` Software MINRES from July 2003

**) Y. Saad and M.H. Schultz, 1986

Sparse matrix storage II

The used solution method for the matrix equation never modifies the matrix $\boldsymbol{C}$ and needs only the

$$\text{product} \qquad \boldsymbol{C} \times \textbf{vector} \Rightarrow \textbf{vector}$$

The $47\,655 \times 47\,655$ matrix C^{global} for the global parameters is sparse ...

The indexed storage scheme in MILLEPEDE for the symmetric matrix requires, in addition to space (8 Bytes) for the $N_{\text{non-zero}}$ non-zero matrix elements, also $N_{\text{non-zero}}$ integer words (4 Bytes) and is optimized for the product $\boldsymbol{C} \times$ vector (only 9 lines of code).

Sparse-matrix storage requires more time than a simple matrix array:

- Array size limited to $(2^{31} - 1)$ 32-bit words on 64-bit system $\equiv$ 8.5 Giga Bytes
  Symm. matrix $(n^2 + n)/2 > 2^{31} - 1 = 2\,147\,483\,647$ already for $n = 2^{16} = 65\,536$ – but MILLEPEDE code allows larger n, if matrix sparse.

  From total 4.1×10^9 index-pairs find set of 0.23×10^9 distinct index-pairs.
  Only 19.9 % of the off-diagonal elements are non-zero, requiring 2.7 Giga Bytes of memory, instead of 18 Giga Bytes/9 Giga Bytes of double precision words.

- During matrix generation (sums): find the location for an index pair (j, k) (using row-pointer + binary search)

Preconditioning

The convergence rate of iterative methods depends on the spectral properties of the matrix. A transformation called *Preconditioning** may improve the spectral properties:

$$\text{instead of} \quad \boldsymbol{C}\,\boldsymbol{x} = \boldsymbol{b} \qquad \text{solve} \qquad \left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{b} \qquad \text{for } \boldsymbol{x}$$

(same solution as original system, but condition number of $\left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)$ smaller). This requires an approximate solution of the equation without using too much extra memory space and computing time.

Method in MILLEPEDE 2009: constraint preconditioner, approximating symmetric $\boldsymbol{C}^{\text{global}}$ by $\boldsymbol{D}$ (diagonal elments only), but keeping the constraints unmodified:

$$\begin{pmatrix} \boldsymbol{C}^{\text{global}} & \boldsymbol{A}^{\mathrm{T}} \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \boldsymbol{D} & \boldsymbol{A}^{\mathrm{T}} \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix} \equiv \begin{pmatrix} \boldsymbol{D}^{1/2} & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{D}^{-1/2} & \boldsymbol{L} \end{pmatrix} \begin{pmatrix} \boldsymbol{D}^{1/2} & \boldsymbol{D}^{-1/2}\boldsymbol{A}^{\mathrm{T}} \\ \boldsymbol{0} & -\boldsymbol{L}^{\mathrm{T}} \end{pmatrix}$$

$$\text{Cholesky decomposition of} \qquad \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$$

The solution $\boldsymbol{x}$ from the two left/right-triangular matrices is determined in two loops.

Now standard method in MILLEPEDE, improving the speed of `MINRES`-solutions.

Only 0.1 % of total Cpu-time

* Preconditioning is an option in `MINRES`.

6. Weak modes II . . . and regularization

Is there a method to suppress the contributions of weak modes?

Regularization: control the *norm of the residuals* and, simultaneously, the *norm of the solution* $\boldsymbol{x}$.

$$F_\tau(\boldsymbol{x}) = \|\boldsymbol{J}\boldsymbol{x} - \boldsymbol{y}\|^2 + \tau \|\boldsymbol{x}\|^2 = \text{minimum}$$

Matrix equation: $(\boldsymbol{x} \Rightarrow \boldsymbol{x} + \boldsymbol{\Delta x})$ $\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{V}_y^{-1}\boldsymbol{J} + \tau \cdot \boldsymbol{1}\right)\, \boldsymbol{x} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{V}_y^{-1}\boldsymbol{y}_x - \tau\boldsymbol{x}$ to be solved for $\boldsymbol{\Delta x}$

$$\left(\boldsymbol{C}^{\text{global}} + \tau \cdot \boldsymbol{1}\right)\, \boldsymbol{x} = \boldsymbol{b}_x - \tau\boldsymbol{x}$$

The effect of regularization is clearly visible in the solution by diagonalization:

$$\boldsymbol{x} = \sum_{j=1}^{n} f_j \frac{1}{\sqrt{\lambda_j}} c_j \boldsymbol{u}_j \qquad \text{with} \quad c_j = \frac{1}{\sqrt{\lambda_j}} \left(\boldsymbol{b}^{\mathrm{T}}\boldsymbol{u}_j\right) \qquad \text{and filter factors} \quad f_j = \left(\frac{\lambda_j}{\lambda_j + \tau}\right)$$

$\rightsquigarrow$ only the (not significant) linear combinations with small eigenvalues are suppressed by the f_j.

Solution of course does not require diagonalization (and it is even faster than without regularization).

Millepede 2009: τ replaced by $\tau \cdot \boldsymbol{D}^{\text{pre}}$ to take into account the different a-priori accuracies.

Use of survey data

MILLEPEDE II allows to use different data, e.g. survey data of pixel barrel structure using photographic techniques (H. Kästli, F.Meier, PSI) in order to improve weakly defined degrees of freedom.

- Cross-hair marks on sensor surface of module (precision of distance between marks on a sensor known better than 0.5 μm).
- Each (of seven) photos covers 2 (of 8) adjoining modules, markers measured precisely (2 μm).

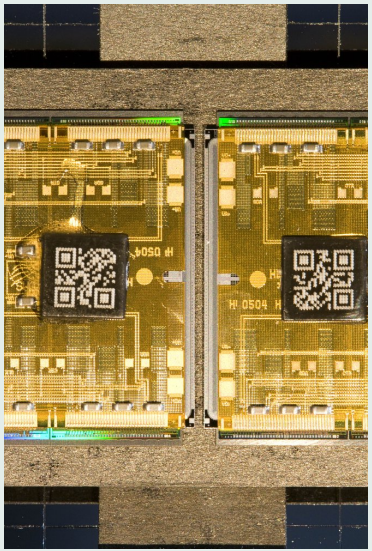
Fit transformation from global aligment parameters $\boldsymbol{x}^{\text{sensor}}$, $\boldsymbol{y}^{\text{sensor}}$ and $\boldsymbol{\gamma}^{\text{sensor}}$ to measured marker coordinates on the photos:

$$\begin{aligned} 7 \text{ photos with } 2 \times 4 \text{ measured values} &= 56 \\ \text{local fit with 4 parameters/photo} &= 28 \\ \text{global fit with 3 alignment parameters/sensor} &= 24 \end{aligned}$$

56 - 28 - 24 = 4 degrees of freedom left
$\rightsquigarrow$ stand-alone fit with 3 constraints possible!

$$\begin{aligned} \text{Accuracy in } \boldsymbol{x}^{\text{sensor}} &\pm 2\ \mu\text{m} \\ \boldsymbol{y}^{\text{sensor}} &\pm 15\ \mu\text{m} \\ \boldsymbol{\gamma}^{\text{sensor}} &\pm 0.2\ \text{mrad} \end{aligned}$$

(if used in MILLEPEDE no extra constraints necessary.)



Photograph as used in survey

⇒more

Summary

Alignment and calibration parameters with (experiment-independent) MILLEPEDE II:

- simultaneous fit of **all** global (alignment) and local parameters;
- include calibration of e.g. Lorentz angle, local values of drift velocity, coefficients for correction functions (e.g. in H1);
- include beam parameters: vertex position, beam direction;
- linear equality constraints to fix undefined degrees of freedom (translation, rotation)
- simultaneous use of all available data types: vertex tracks, cosmics (also $\boldsymbol{B} = 0$), halo muons, Laser data, survey data to fix weakly defined degrees of freedom + regularization;
- outlier rejection and down-weighting of bad single hits

Now: Support of MILLEPEDE II by the "Statistical Tools Group" of

Analysis Centre of the Helmholtz Alliance "Physics at the Tera Scale"

The global alignment parameters x^{sensor}, y^{sensor} and γ^{sensor} are determined by the fit of a transformation to the measured marker positions on the photographies.

Real marker coordinates are

$$x^{\text{marker}} = x^{\text{sensor}} - v^{\text{marker}} \cdot \cos\gamma^{\text{sensor}} + u^{\text{marker}} \cdot \sin\gamma^{\text{sensor}}$$
$$y^{\text{marker}} = y^{\text{sensor}} + v^{\text{marker}} \cdot \sin\gamma^{\text{sensor}} + u^{\text{marker}} \cdot \cos\gamma^{\text{sensor}} ,$$

where v^{marker}, u^{marker} are the precisely known coordinates of the marker on the sensor, and x^{sensor}, y^{sensor} and γ^{sensor} are the real sensor coordinates and orientation angle, which have to be determined in the fit.

Transformation from real marker coordinates to measured marker coordinates on the photography is

$$x^{\text{meas}} = a_1 + a_3 \cdot x^{\text{marker}} + a_4 \cdot y^{\text{marker}}$$
$$y^{\text{meas}} = a_2 - a_4 \cdot x^{\text{marker}} + a_3 \cdot y^{\text{marker}}$$

where the parameters $a_1 \ldots a_4$ are to be determined in the transformation fit (assuming one scaling and one rotation parameter per photo) to the photo.

Constraints: $\sum \Delta x^{\text{sensor}} = 0$ $\sum \Delta y^{\text{sensor}} = 0$ $\sum \Delta\gamma^{\text{sensor}} = 0$

. . . omitting all indices.

⇐**back**

Contents