

## Status and Prospects for Millepede II

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 and equality constraints in the fit. The Millepede II version, now on the web, uses fast methods in the non-iterative fit.



1. Alignment
2. Optimization
3. Millepede
4. Strategies and Prospects for MILLEPEDE II

**Design: experiment-independent program, not specific to alignment and tracks.**

Year	What happened?
1996	First studies at CERN (Opal)
1997	First version used in H1
1998	Used in H1 for Vertex det. and Central Jet Chamber
1999	Used with up to 4 800 parameters
2000	Millepede I on the web, last change
2001	
2002	
2003	
2004	
2005	Start of new development for large nr of parameters
2006	Test with H1 and cms data, up to 50 k parameters
2007	Millepede II on the web (25.th May)

Now used by: H1, ZEUS, HERAb, CMS, LHCb, Alice, PHENIX, STAR ...

⇒ Talk by M. Stoye: Track-based alignment of the CMS Tracker with Millepede II (includes studies on  $\chi^2$ -invariant deformations)

# The alignment challenge

---

*Mankind always sets itself only such problems as it can solve ...*  
– KARL MARX (1859)

Alignment of a LHC track detector requires  
the optimization of a large number of parameters,  
subject to a large number of equality constraints,  
from a large amount of data

... increasing numbers in the future!

Note: standard methods require  
 $\text{space} \propto n^2$        $\text{cpu-time} \propto n^3$

100 000 alignment parameters  
100 constraints  
1 000 000 tracks  
Laser data  
survey data

- Alignment determination should be possible in a reasonable time!
- What is a reasonable time?    Answer: **between 1 hour and 1 day!**    **Can this be realized?**

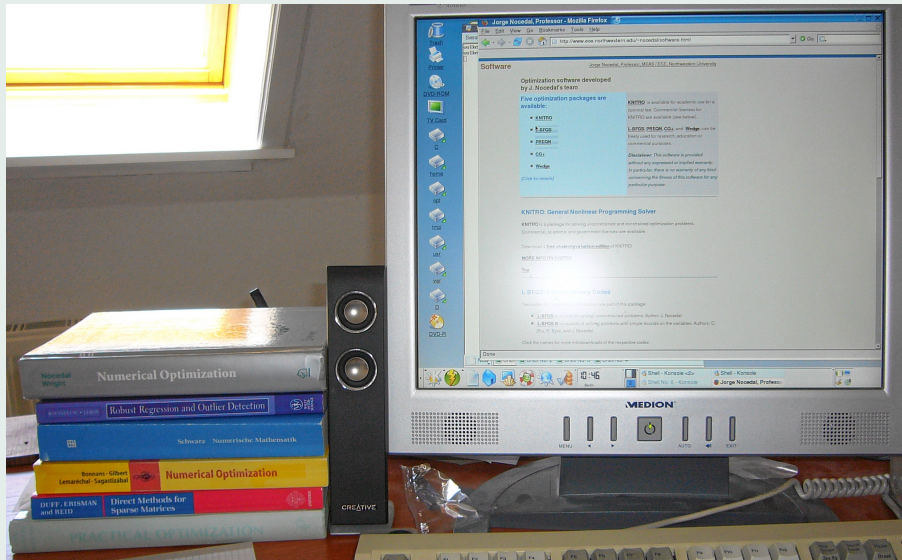
One possibility: use cluster with parallel computing of many powerful cpus ...

## ... the alternative

---

*The deep structure of reality resides in mathematical relations.*  
–PYTHAGORAS (ca. 590 – 500 B.C.)

⇒ study recent mathematical developments in the WEB and the mathematical literature!



→ references at the end of the talk.

## Translation table ...

---

HE Physics $\Rightarrow$	$\Leftarrow$ Mathematics, Statistics
$\chi^2$ -function .....	objective function (log-likelihood function) ....
constraint .....	measurement-term in objective function .....
? .....	constraint (equality, inequality constraints) ....
“linear system of equations requires inversion”	“never solve a system of equations by inversion”
“solve 4200 equations in 4200 unknowns: computational infeasible; even worse, non-linear fit won't converge” .....	“current algorithms ... for generally constrained optimization routinely solve systems in the tens and, perhaps even, hundreds of thousands of unknowns and constraints” .....

The strategy in High Energy Physics experiments is often determined by the general HEP opinions about the mathematical handling of large problems.

- Construct an **Objective Function**  $F(\mathbf{x})$  from **statistical** considerations, depending on the parameter  $n$ -vector  $\mathbf{x}$ , to be minimized.  $\mathbf{x} \in \mathcal{R}^n$  with start value  $\mathbf{x}_0$ .

- **Step 1:** construct quadratic model  $M_k(\mathbf{d}) = F_k + \nabla F_k^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{C}_k \mathbf{d}$

function value:  $F_k$       gradient:  $\nabla F_k$       sec. der. matrix (Hessian):  $\mathbf{C}_k \approx \nabla^2 F_k$  at  $\mathbf{x}_k$

- **Step 2:** determine minimum of quadratic model function:

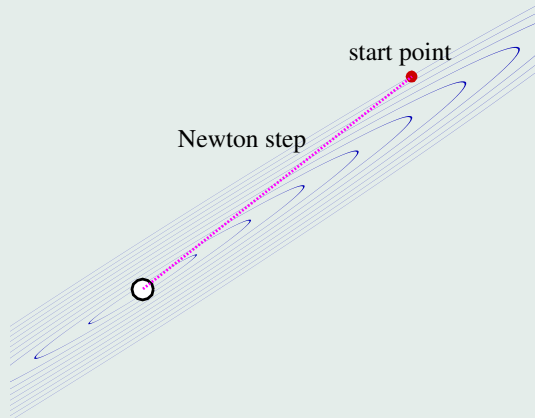
solve  $\boxed{\mathbf{C}_k \mathbf{d}_k = -\nabla F_k}$  for  $\mathbf{d}_k$  and update:  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$

- repeat steps 1 and 2 (i.e. iterate) for non-linear problems.
- **Step 3:** covariance matrix of parameter  $n$ -vector  $\mathbf{x}$  is given by inverse matrix:  $\mathbf{B} = \mathbf{C}^{-1}$

Improved by line-search  $\phi(\alpha) \equiv F(\mathbf{x}_k + \alpha \cdot \mathbf{d}_k)$  with optimization of factor  $\alpha$ .

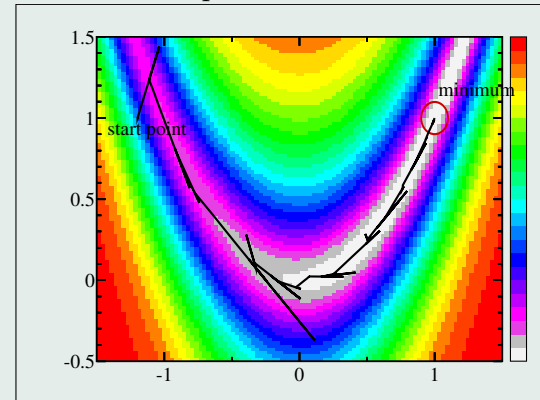
Single step only, if  $F(\mathbf{x})$  is quadratic (i.e.  $\mathbf{C} = \text{constant}$ ).

... large correlation



Quadratic function is minimized by one NEWTON step; steepest-descent would be inefficient due to high correlation.

... non-quadratic behaviour



Non-quadratic function: minimization requires to follow very narrow curved valley.

NEWTON method based on second-derivative matrix has *quadratic convergence* rate.

Simple methods like steepest-descent with only *linear convergence* rate are slow; convergence may not occur as the iteration stagnates (can be misinterpreted as indication for convergence).

## Constraints

---

*Freedom and constraint are two aspects of the same necessity.*  
–ANTOINE DE SAINT-EXUPÉRY, in La Citadelle (1948)

Constraint equations for  $m$  linear (**equality**) constraints are described by

$$\mathbf{Ax} = \mathbf{c} \quad (\mathbf{A} \text{ has } m \text{ rows})$$

Task: minimize  $F(\mathbf{x})$  subject to  $\mathbf{Ax} = \mathbf{c}$

---

Lagrange method: introduce  $m$  multipliers  $\boldsymbol{\lambda}$

$$\mathcal{L}(\mathbf{x}) = F(\mathbf{x}) + \boldsymbol{\lambda}^T (\mathbf{Ax} - \mathbf{c}) \quad \left( \begin{array}{c|c} \mathbf{C} & \mathbf{A}^T \\ \hline \mathbf{A} & 0 \end{array} \right) \left( \begin{array}{c} \mathbf{d} \\ \boldsymbol{\lambda} \end{array} \right) = \left( \begin{array}{c} -\nabla F \\ \mathbf{c} \end{array} \right)$$



Joseph-Louis Lagrange  
(1736 – 1813)

---

Matrix equation has unique solution (for sufficient constraints) **even for singular matrix  $\mathbf{C}$** , but the total matrix is no longer positive definite (has positive and negative eigenvalues).

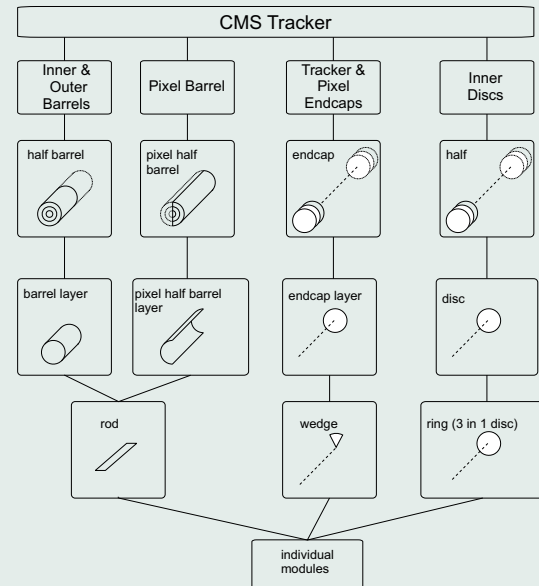
# Why constraints?

## 1. Avoid singularity of total matrix

- Overall translation + rotation is undefined in case of residual minimization!
- Singularity removed by constraints requiring zero translation + rotation

## 2. Introduction of structural constraints

- A large track detector has a sub-structure, visualized by tree structure.  $\longrightarrow$
- Set of global parameters assigned to unit, and to each sub-unit, with zero overall translation + rotation by constraints.
- Global sub-unit parameters can be fixed  $\Rightarrow$  total number of parameters is reduced.



### 3. Millepede

---

A “global” objective function  $F(\mathbf{p}, \mathbf{q})$  is constructed, which depends on the alignment corrections  $\mathbf{p}$  and all track parameters  $\mathbf{q}$  and all laser data parameters and all all other parameters

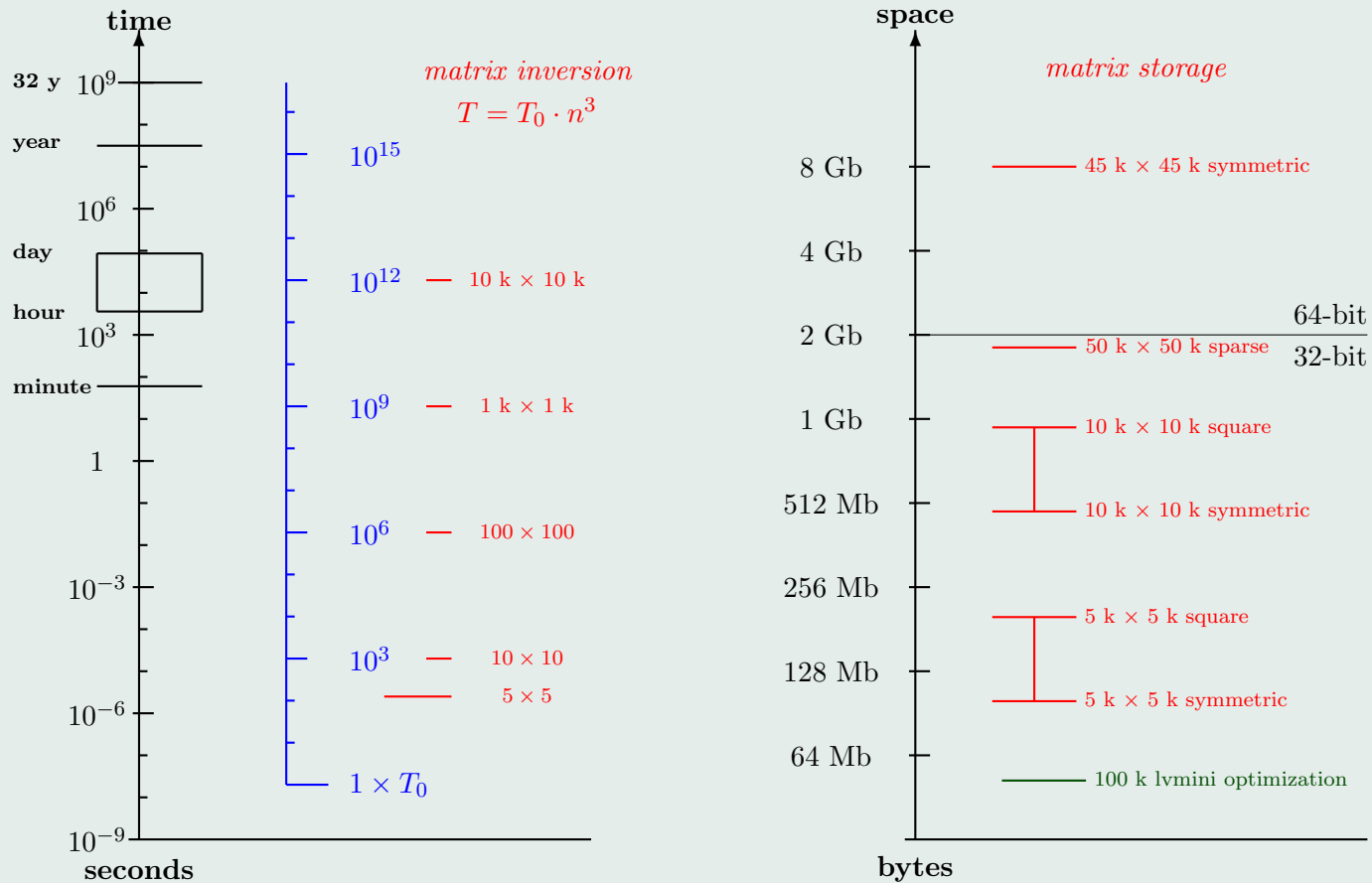
$$F(\mathbf{p}, \mathbf{q}) = \frac{1}{2} \sum_{\text{data sets}} \left[ \sum_{\text{events}} \left( \sum_{\text{tracks}} \left( \sum_{\text{hits}} \Delta_i^2 / \sigma_i^2 \right) \right) \right] + \sum [\text{terms depending on Laser data and Survey data}]$$

Essential properties of alignment algorithm:

- Simultaneous fit of all parameters  $\mathbf{p}, \mathbf{q} \dots$  (Millepede principle  $\rightarrow$  next page) in a single step, using Hessian matrix  $\mathbf{C}$ ;  
number of alignment parameters may be large (e.g. about 100 000).
- Find minimum of  $F(\mathbf{p}, \mathbf{q})$ , subject to equality constraints;  
number of constraints may be large (e.g. about 100).
- Include detailed outlier treatment:  
reject very bad data, and down-weight bad single measurements (M-estimate).  
*Note: initial deviations may be large due to misalignment!*
- Cpu-time below 1 day!



# Space-time



## Solution of matrix equation in MILLEPEDE II

---

Different methods for different conditions (number of parameters, sparsity), no single optimal method. All methods can include large (e.g. 100) number of constraints, using the Lagrange multiplier method (→ matrix not positive-definite).

Methods: matrix inversion, diagonalization, variable-band matrix decomposition, and ...

**generalized minimal residual method (GMRES):** solve  $\mathbf{C} \mathbf{x} = \mathbf{y}$  or minimize  $\|\mathbf{C} \mathbf{x} - \mathbf{y}\|_2$  (subroutine MINRES\*), symmetric matrix  $\mathbf{C}$  may be indefinite, very large and sparse).

Solution method in analogy to the method of conjugate gradients (Hestenes, Stiefel 1952). MINRES needs only the product  $\mathbf{C} \times \mathbf{vector}$ . Method is iterative, convergence speed depends on eigenvalue structure.

Convergence is accelerated by *preconditioning* (i.e. GMRES combined with another approximate solution method), which improves the eigenvalue structure.

Variable-band matrix decomposition used for preconditioning:  $5000 \times$  faster than inversion.

---

\*) 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.

## Cpu-times for alignment

---

Approximate formula for cpu-time:

$$\text{cpu-time} = T(N_{\text{tracks}}, N_{\text{parameters}}) = N_{\text{iterations}} \times \left( \alpha \cdot N_{\text{tracks}} + \beta \cdot N_{\text{parameters}}^{\gamma} \right) \quad \gamma \geq 1$$

Values of  $\alpha, \beta, \gamma, N_{\text{iterations}}$  depend on algorithm.

Experiment	type	$N_{\text{parameters}}$	$N_{\text{tracks}}$	$N_{\text{iterations}}$	$N_{\text{constraints}}$	cpu-time
D0	L, C <sup>++</sup>	6 000	0.7 Mio	70 – 100	–	1 – 3 days
BaBar Si tracker	L, C <sup>++</sup>	1 440	?	~ 100	–	< 24 hours
<b>Millepede (cms)</b> (M. Stoye study)	G, F77	50 000	3 Mio		130	1:40 hour + file times

L = local method with  $F(\mathbf{p})$  (i.e. track parameters fixed), resulting bias removed by large number of iterations. No constraints?

G = global non-iterative method with  $F(\mathbf{p}, \mathbf{q})$  (i.e. simultaneous fit of alignment and track parameters) with outlier down-weighting and -rejection.

## How to use MILLEPEDE

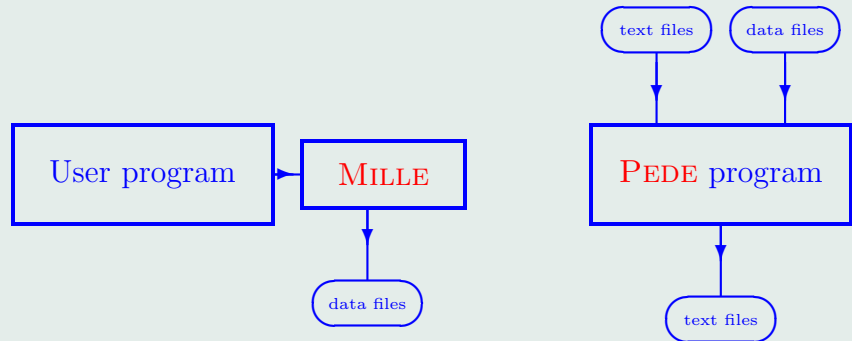
Input = sets of single measured data points from local fits (e.g. KALMAN fit with track hits):

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

Derivatives express the change of residual  $z_i = y_i - f(x_i, \mathbf{q}, \mathbf{p})$ , if  $q_j$  or  $p_\ell$  is changed by  $\Delta q_j$  or  $\Delta p_\ell$ .

(1) File with  $z_i = y_i - f(x_i, \mathbf{q}, \mathbf{p})$ ,  $\sigma_i$  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:

- select files and solution method (inversion, diagonalization, fast sparse method ...)
- information on measurement of linear combinations of global parameters (e.g. survey data)
- status of global parameters (e.g. initial values, fixed/variable, presigma)

```
Fortranfiles
!/home/albert/filealign/lhcrun1.11      ! data from first test run
/home/albert/filealign/lhcrun2.11      ! data from second run
Cfiles
/home/albert/filealign/cosmics.bin      ! cosmics
/home/albert/detalign/mydetector.txt    ! file from previous result file
/home/albert/detalign/myconstr.txt      ! test constraints

Parameter                               ! set status for selected parameters
201 0.0      0.0                        ! variable parameter (default), initial value = 0
202 1.732    -1.0                       ! fixed parameter, initial value = 1.732
204 1.23     0.020                      ! variable parameter with presigma

constraint 0.14                         ! numerical value of constraint equation
713 1.0 720 0.5                         ! pairs of parameter label and numerical factor

Measurement 10.3 0.1                   ! survey distance [713]-[714] = 10.3 +- 0.1
713 1.0 714 -1.0

method sparseGMRES 5 0.1               ! Generalized residual minimization, sparse matrix
bandwidth 6                            ! with variable-band matrix preconditioning
chisqcut 15 6                          ! chisquare cut for first and second loop
outlierdownweighting 5                 ! down-weighting in 5 local iterations
dwfractioncut 0.2                      ! reject bad records
printrecord 13 -1                      ! debug printout for record 13 and worst record
histprint                               ! print histograms
subito                                  ! exit after first step
end
```

- Single-step (non-iterative) Millepede method works for large problems.
- Fast solution of large system of equations with GMRES plus preconditioning with variable-band matrix solution.

Example: 50 000 parameters, matrix with 15 % non-zero elements, 100 constraints  $\Rightarrow$   
< 2 Gbyte matrix, 10 Minutes cpu time  
(instead of 1 month for matrix inversion with 20 Gbyte matrix)

Solution of 100 000 parameter problem should be possible on 64-bit systems with 8 Gbyte memory.

Variances of selected parameters can be calculated, if wanted.

- For smaller systems further methods, which
  - give complete covariance matrix (inversion),
  - give all eigenvalues and eigenvectors for analysis of weakly defined degrees of freedom.
- Accurate solution, even in case of round-off errors in matrix equation, with line-search algorithm.
- Large number of constraints possible (up to 1000 or more) with Lagrange formalism, plus special mathematical algorithm to reach high constraint precision.  
Undefined degrees of freedom fixed by constraints. Structural constraints can be defined.
- No problems with large amount of data, e.g several Million tracks. Measurements like survey data either in text files or in binary files (?).
- Automatic recognition of parameters (with label system) from input data.
- Detailed outlier treatment for binary data. Could be extended to data from text files.

Essential properties of an efficient alignment strategy:

- simultaneous fit of **all** parameters, no separate alignment of detector parts;
- include calibration of e.g. Lorentz angle, local values of drift velocity, correction coefficients for correction functions;
- include beam parameters: vertex position, beam direction;
- use realistic data model (for tracks, and other data ...);
- simultaneous use of **all** data types:  
tracks, two-track particles, cosmics,  $B = 0$  cosmics, halo muons, Laser data, survey data;
- linear equality constraints  
to fix undefined or weak degrees of freedom and to define parameter structure;
- outlier down-weighting (single hits) and outlier rejection.

... because **all parameters are correlated**, and isolated optimization of a subset may distort results.

## Limited memory BFGS (L-BFGS)

---

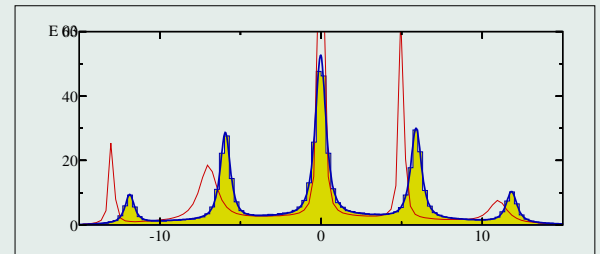
What to do, if the number of parameters is 200 000 or 500 000?

Large-scale limited memory BFGS algorithm has space requirement *proportional* to number of parameters, with e.g. only 60 Mbyte for 100 000 parameters.

Minimization package `lvmini`, using L-BFGS, developed for  $n = 2$  up to several 100 000 parameters, needs gradient  $\nabla F$  – so far no constraints possible.

280-parameter Neural Net training and  $> 100\,000$  parameter minimization under study.

Use of `lvmini` in Millepede II would require different method for constraints: elimination method under study.



`lvmini`-example of fit with 20 parameters  
Initial parameter values correspond to red line.  
Minimization requires  $\approx 100$  function evaluations.

# Prospects

---

New MILLEPEDE II      OUT NOW

Download from: [www.desy.de/~blobel/](http://www.desy.de/~blobel/) into  
fresh directory:

```
tar -xzf Mptwo.tgz
make
./pede -t
```

[New lvmini-package      OUT NOW]

Use of > 400 Mbyte requires to change 1 statement in code  
+ makefile for 64-bit system.

---

MILLEPEDE II can be used:

- Feedback welcome!
- Feedback necessary!
- Perhaps several (small) changes during the coming weeks (if feedback  $\neq 0$ ).
- Addition of
  - L-BFGS method, for even larger number of parameters?
  - histogram viewer for histogram file?

## References

- [1] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer (1999)
- [2] W.C. Davidon, *Variable metric method for minimization*, manuscript (1958), finally published SIAM J. Optimization **1** (1991) pp. 1-17.
- [3] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Mathematics of Computation **35** (1980) pp.773-782
- [4] J.J.Moré and D.J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Transactions on Mathematical Software **20** (1994), pp. 286-307
- [5] F. James and M. Roos, MINUIT, Function Minimization and Error Analysis, Reference Manual, CERN Program Library Long Writeup D506 (1994)  
F. James and M. Winkler, MINUIT Users Guide (C++ Version), CERN (2004)
- [6] Ph.E. Gill et al., *Practical Optimization*, Academic Press (1981)
- [7] J.F. Bonnans et al., *Numerical Optimization – Theoretical and Practical Aspects*, Springer (2000)
- [8] I.S. Duff et al., *Direct Methods for Sparse Matrices*, Oxford Science Publ. (1986)
- [9] H.R. Schwarz, *Numerische Mathematik*, Teubner (1993)
- [10] P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection*, Wiley (2003)

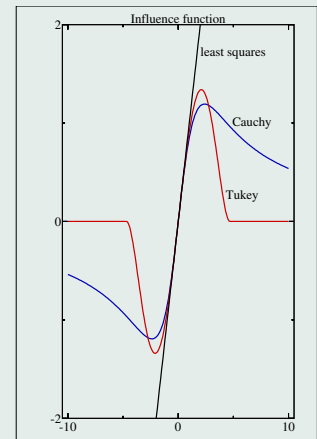
The presence of outliers in the data can deteriorate the alignment result.  
 Difficulty: wrong initial alignment parameters can fake outliers.

**Millepede I:** Large initial cut at  $\approx 10\sigma$  reduced to  $3\sigma$  in  $\approx 5$  iterations.

**Millepede II:** Same as Millepede I, in addition technique of M-estimates applied to local fits after the first iteration.

**M-estimates.** The objective function in least squares is the sum of **squares** of scaled residuals  $z$ , with **larger influence for larger residuals** (outliers). The **square** is replaced in M-estimates by a dependence with reduced influence for larger residuals.

	$\rho(z) = \ln \text{pdf}(z)$	influence function $\psi(z) = d\rho(z)/dz$	add. weight $\omega(z) = \psi(z)/z$
Least squares	$= \frac{1}{2} z^2$	$= z$	$= 1$
Cauchy ( $c = 2.3849$ )	$= \frac{c^2}{2} \ln \left( 1 + (z/c)^2 \right)$	$= \frac{z}{1 + (z/c)^2}$	$= \frac{1}{1 + (z/c)^2}$
Huber $\begin{cases} \text{if }  z  \leq c = 1.345 \\ \text{if }  z  > c = 1.345 \end{cases}$	$= \begin{cases} z^2/2 \\ c( z  - c/2) \end{cases}$	$= \begin{cases} z \\ c \cdot \text{sign}(z) \end{cases}$	$= \begin{cases} 1 \\ c/ z  \end{cases}$



## Least squares method and sparse matrix

---

Hessian  $\mathbf{C}$  may be a **sparse** matrix, with many elements  $(\mathbf{C})_{jk} \equiv 0 \dots$

Element  $(\mathbf{C})_{jk} \neq 0$ , if parameters  $j$  and  $k$  appear together at a measured data value.

$$\mathbf{C} = \begin{pmatrix} x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \end{pmatrix} \Rightarrow \mathbf{C}^{-1} = \begin{pmatrix} x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \end{pmatrix}$$

Inverse matrix  $\mathbf{C}^{-1}$  is the covariance matrix of the parameters.  $\mathbf{C}^{-1}$  is a **dense** matrix with all elements  $(\mathbf{C}^{-1})_{jk} \neq 0$ :

all pairs  $(j, k)$  of parameters have correlation  $\neq 0$ .

## Millepede simultaneous fit

---

$$\begin{pmatrix} \sum_k \mathbf{C}_k & \dots & \mathbf{H}_k^{\text{global-local}} & \dots \\ \hline \vdots & \ddots & 0 & 0 \\ \hline (\mathbf{H}_k^{\text{global-local}})^T & 0 & \mathbf{C}_k^{\text{local}} & 0 \\ \hline \vdots & 0 & 0 & \ddots \end{pmatrix} \times \begin{pmatrix} \Delta \mathbf{p}^{\text{global}} \\ \hline \vdots \\ \hline \Delta \mathbf{q}_k^{\text{local}} \\ \hline \vdots \end{pmatrix} = \begin{pmatrix} \sum_k \mathbf{b}_k^{\text{global}} \\ \hline \vdots \\ \hline \mathbf{b}_k^{\text{local}} \\ \hline \vdots \end{pmatrix}$$

The Millepede principle: transfer of the local information to the global Hessian  $\mathbf{C}$

$$\mathbf{C}_{\text{global}} = \sum_k \mathbf{C}_k - \sum_k \mathbf{H}_k \mathbf{C}_k^{-1} \mathbf{H}_k^T \quad (\text{“Schur complement”})$$

(transfer of the local information to the global Hessian  $\mathbf{C}$ .)

$$\begin{pmatrix} \mathbf{C}_{\text{global}} \end{pmatrix} \times \begin{pmatrix} \Delta \mathbf{p}^{\text{global}} \end{pmatrix} = \begin{pmatrix} \sum_k \mathbf{b}_k^{\text{global}} \end{pmatrix}$$



## Elements of the covariance matrix with MINRES

---

The inverse of matrix  $\mathbf{C}$  is the covariance matrix  $\mathbf{V}$  of the alignment parameters. This is available with matrix inversion and diagonalization, but **not** with MINRES.

Method to compute *some* elements of  $\mathbf{V}$  with MINRES:

Solution of matrix equation  $\mathbf{C}\mathbf{V}=\mathbf{1}$  (right hand-side is unit matrix)

for  $\mathbf{V}$  would give the complete covariance matrix  $\mathbf{V}$  and

Solution of matrix equation  $\mathbf{C}\mathbf{v}_j=\mathbf{e}_j$  (right hand-side is  $j$ -th column of unit matrix)

for  $\mathbf{v}_j$  will give on  $j$ -th column of the covariance matrix  $\mathbf{V}$ .

Elements of covariance matrix are determined by hits statistics and by geometry.

## Feasible parameters

---

A vector  $\mathbf{x}$  compatible with constraint equations  $\mathbf{Ax} - \mathbf{c} = 0$  is called a *feasible* vector.

Round-off errors can introduce small deviations:  $\mathbf{Ax} - \mathbf{c} = \boldsymbol{\varepsilon}$ .

In order to **force feasibility** a minimum-norm correction  $\Delta\mathbf{x}$  with  $\min \|\Delta\mathbf{x}\|_2$  is calculated by

$$\Delta\mathbf{x} = -\mathbf{A}^T (\mathbf{AA}^T)^{-1} \boldsymbol{\varepsilon}$$

in each iteration.

The product  $\mathbf{AA}^T$  is a square  $m$ -by- $m$  non-singular matrix for sufficient constraints.

# Contents

<b>1. Alignment</b>	<b>2</b>	Millepede simultaneous fit . . . . .	24
The alignment challenge . . . . .	3	Variable-band matrix . . . . .	25
... the alternative . . . . .	4	Elements of the covariance matrix with MINRES	26
Translation table ... . . . .	5	Feasible parameters . . . . .	27
<b>2. Optimization</b>	<b>6</b>	<b>Table of contents</b>	<b>28</b>
Minimization . . . . .	7		
Constraints . . . . .	8		
Why constraints? . . . . .	9		
<b>3. Millepede</b>	<b>10</b>		
Simultaneous fit . . . . .	11		
Space-time . . . . .	12		
Solution of matrix equation in MILLEPEDE II . .	13		
Cpu-times for alignment . . . . .	14		
How to use MILLEPEDE . . . . .	15		
MILLEPEDE II Keywords . . . . .	16		
<b>4. Strategies and Prospects for Millepede II</b>	<b>17</b>		
Alignment strategies . . . . .	18		
Limited memory BFGS (L-BFGS) . . . . .	19		
Prospects . . . . .	20		
<b>References</b>	<b>21</b>		
<b>If there are these questions ...</b>	<b>22</b>		
Outliers ... . . . .	22		
Least squares method and sparse matrix . . . . .	23		