Large-matrix inversion with MillePede-II and application to track-based alignment.
Claus KLEINWORT, Patrick L.S. CONNOR

Introduction

Purpose
- Developed in context of track alignment.
- Required for inversion of large, sparse matrices.
- Implement various tricks of linear algebra to perform matrix inversion in block-matrix algebra & parallelize.

Control & History
- Development of MillePede in FAME-TT started in 1996 by Volker Blobel (CERN-Hamburg) [1].
- First implementation in 1996 for H1 at HERA experiment as Phase-I tracker alignment in LMK by Claus Kleinwort (DESY) [2].
- Phase-II with MillePede II since 2005 for use of large scale, the latest revision in 2010 (for LHC) with improvements mainly on memory management.
- Official code in Python-5D (most performing solving), basing implementation in C++
- Development still very active, mainly by Claus Kleinwort (2012).
- Successfully used in many HEP experiments.

Why
- To be implemented by the user according to actual problem
- MillePede-II in context of track alignment.
- Outputs residuals, dérivatives & uncertainties for local and global parameters.
- Implementation available for Fortran, C++ & Python.
- Parallelization with OpenMP (gcc4).

Mathematical principles

General formulation
- \( A \), \( m \), \( d \) describe independent measurements.
- \( \alpha \) represents the first derivative of the measurement.

A bit of block-matrix algebra
- \( \sum_{i=1}^{n} \begin{pmatrix} \alpha_i \end{pmatrix} \cdot \begin{pmatrix} A_i \end{pmatrix} \cdot \begin{pmatrix} m_i \end{pmatrix} = \begin{pmatrix} \sigma \end{pmatrix} \)
- \( n \) local & global parameters
- \( m_n \) local parameters
- \( m_g \) global parameters
- \( m_T \) global parameters

More local measurements will not significantly affect computing time

Invert large, symmetric matrices

General Broken Lines

Fast track model necessary for tracker alignment

Introduction
- Mathematically equivalent to Kalman fiber for track alignment (aligned in tracking in KEPC).
- Computes parameter, localised to local and global parameters.
- Implementation available for C++, Python, including H1 step in context of tracker alignment.

GRL's in a nutshell
- Main deviations from perfect holes (2 parameters) are due to multiple scattering in material.
- Inclination measured as the trajectory is described in 3D.
- GRLs are used in the first step to separate local parameters.
- GRLs also describe the influence of surrounding structure.

Solution
- Full inversion: \( C' \) is the inverse of \( C \).

Future
- Massive parallelisation

Acknowledgements
- Multiple contributions from the Belle-II collaboration.

What about your project?