General Broken Lines (GBL) for track fitting and alignment

C. Kleinwort - DESY

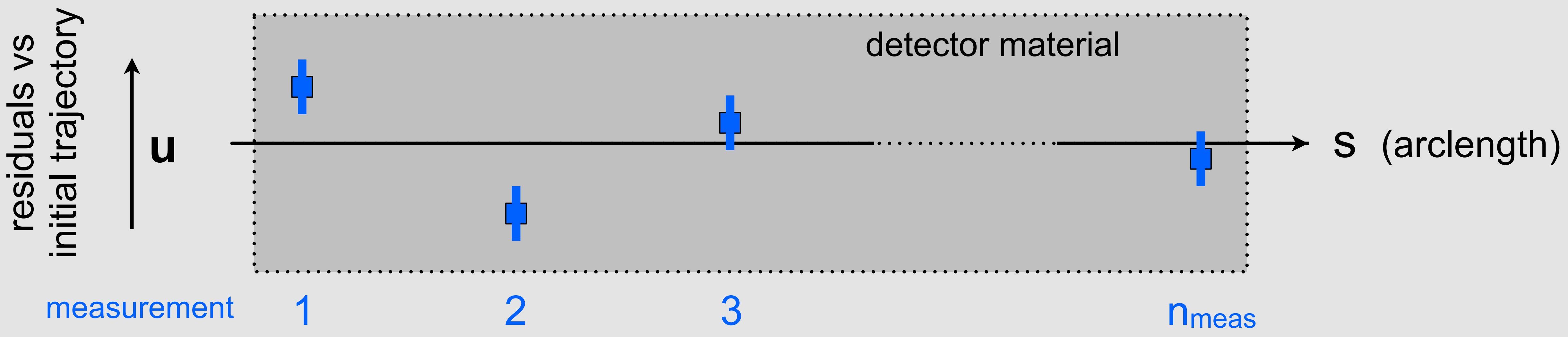
EUTelescope Workshop 26.03.13

Overview

- General Broken Lines NIM A, 673 (2012), 107-110
  - Based on original broken lines by V. Blobel (UHH)
  - Concepts
    - Definition, construction of trajectory
    - Local track parameters, implementation of fit
    - Comparison with Kalman filter
- GBL software package, use cases
    - Simple track fitting
    - Tracker alignment with Millepede-II
- Summary

Definition

- Trajectory based on 'general broken lines'
  - Track refit to add with local offsets $\mathbf{u}_i(s_i)$ the description of multiple scattering to an initial trajectory ('seed') based on the propagation in a magnetic field (and energy loss)

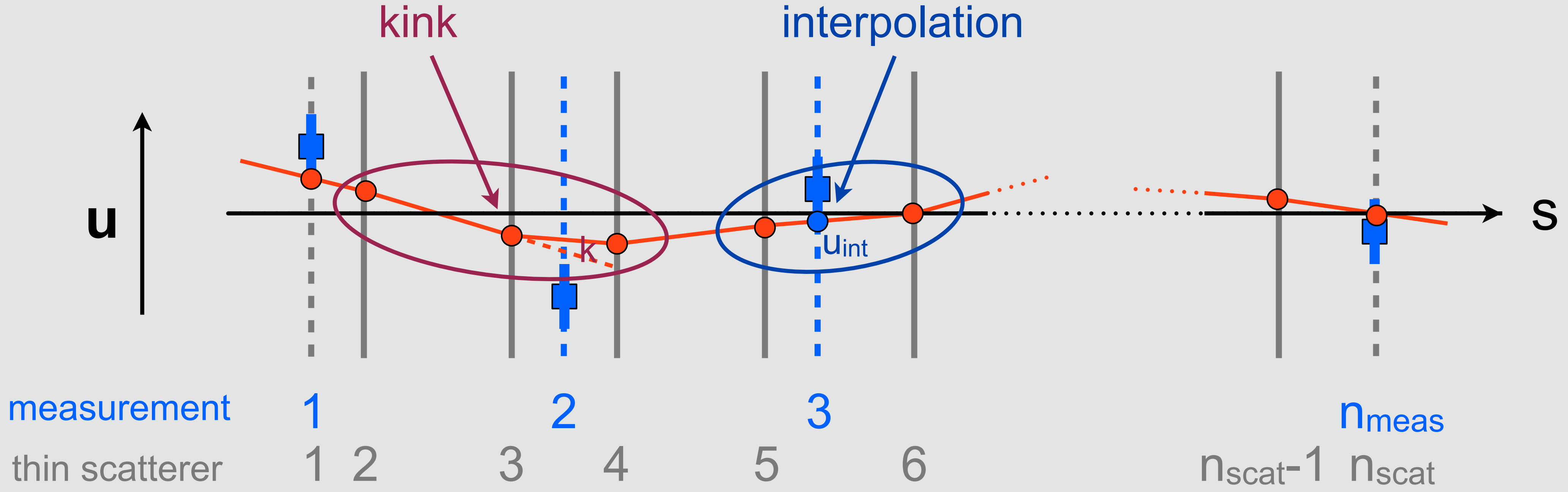


Construction (I)

- Trajectory based on 'general broken lines'
  - Refit for set of (n_{meas}) measurements **m** on a track based on sequence of (n_{scat}) thin scatterers
    - Material between adjacent measurements is in general a thick scatterer, represented by (up to) 2 thin scatterers with similar mean and RMS of material
    - Dummy thin scatterers at first and last measurement
    - Offsets **u** in local system (u_1,u_2,w) at each scatterer
    - Measurements can coincide with a scatterer or are described by interpolation of adjacent scatterers
  - List of points (meas., scat.) ordered in arc-length

Construction (II)

- Prediction $\mathbf{u}_{int}$ at measurement **m** from interpolation of adjacent scatterers (➜ residuals $\mathbf{r}_m = \mathbf{m} - \partial\mathbf{m}/\partial\mathbf{u} \cdot \mathbf{u}_{int}$)
- Triplets of consecutive scatterers define kinks **k**
  - 2D multiple scattering angles at central scatterers
    - Expectation value zero, variance $\mathbf{V}_k$ according to central scatterer
  - Additional n_{scat}-2 2D residuals $\mathbf{r}_k = \mathbf{k}$ (+$\mathbf{k}_0$ if iterating)



Construction (III)

- Track (fit) parameters
  - One common ‘curvature’ correction
  - n_{scat} 2D (small) offsets **u**, $n_{scat} \leq 2n_{meas}$
- Seeding
  - Internally from (fit of) same measurements
  - Externally from (fit of) independent measurements

Local track parameters (I)

- Local track parameters $\mathbf{p}_i$
  - Defined (by user) in orthonormal system (u_1,u_2,w) at each point 'i'
    - Offsets $\mathbf{u}=(u_1,u_2)$ in local system
    - Direction: angles or slopes (e.g. $\mathbf{u}'=\partial\mathbf{u}/\partial w$) or ..
    - Curvature: q/p or ..
  - Use (q/p, $\mathbf{u}'$, $\mathbf{u}$) in the following
  - Technical constraint
    - Multiple scattering covariance matrix must be diagonal (at least one u_i perpendicular to track direction)

Local track parameters (II)

- Curvilinear system (x_T, y_T, z_T) well suited
  - Constructed from flight direction **T** (from seed) at point: **$W=Z_T=T$, $U_1=X_T=Z\times T/|Z\times T|$, $U_2=Y_T=T\times X_T$**
  - Curvilinear track parameters $\left(\frac{q}{p},\lambda,\phi,x_\perp,y_\perp\right)$
- Refit determines at each point corrections to the local track parameters (from the seed)

★ Propagation on initial trajectory needs jacobian for transformation of local parameters

- $\mathbf{T}_i^+ = \partial\mathbf{p}_i/\partial\mathbf{p}_{i-1}$ (, $\mathbf{T}_i^- = \partial\mathbf{p}_i/\partial\mathbf{p}_{i+1} = (\mathbf{T}_{i+1}^+)^{-1}$)

Implementation (I)

- Linear least squares fit of **x**=(Δq/p,…,$\mathbf{u}_j$,…)
  - Minimize $$\chi^2(\Delta\tfrac{q}{p},\mathbf{u}_1\ldots\mathbf{u}_{n_{scat}})=\sum_{i=1}^{n_{meas}}(\mathbf{m}_i-\mathbf{H}_i\cdot\mathbf{p}_{\text{int},i})^t\,\mathbf{V}_{meas,i}^{-1}(\mathbf{m}_i-\mathbf{H}_i\cdot\mathbf{p}_{\text{int},i})$$

(1) $$+\sum_{i=2}^{n_{scat}-1}\left(\mathbf{k}_i+\mathbf{k}_{0,i}\right)^t\mathbf{V}_{k,i}^{-1}\left(\mathbf{k}_i+\mathbf{k}_{0,i}\right)\quad\left(+\Delta\mathbf{p}_{seed}^t\mathbf{V}_{seed}^{-1}\Delta\mathbf{p}_{seed}\ \text{for ext. seed}\right)$$

$$\mathbf{p}_{\text{int}}=\left(\Delta\tfrac{q}{p},\mathbf{u}'_{\text{int}},\mathbf{u}_{\text{int}}\right),\quad\mathbf{H}=\frac{\partial\mathbf{m}}{\partial\mathbf{p}_{\text{int}}}$$ (up to 5D measurement **m**)

  - Need local derivatives $\partial\mathbf{p}_{\text{int},i}/\partial\mathbf{x}$, $\partial\mathbf{k}_i/\partial\mathbf{x}$ to get corresponding linear equation system **A·x=b**
    - $\mathbf{u}'_{\text{int},i}$, $\mathbf{u}_{\text{int},i}$, $\mathbf{k}_i$ depend only on few (≤3) adjacent $\mathbf{u}_j$
    - Matrix **A** has band structure with band width m≤5 and
    - from general Δq/p dependence (full) border of size b=1

Implementation (II)

- Bordered Band Matrix

$$A_{ij} = 0 \text{ for } \min(i,j) > b \wedge |i-j| > m$$

$$\left(\begin{array}{c|ccccccc} d & b & b & b & b & b & b & b \\ \hline b & d & m & m & 0 & 0 & 0 & 0 \\ b & m & d & m & m & 0 & 0 & 0 \\ b & m & m & d & m & m & 0 & 0 \\ b & 0 & m & m & d & m & m & 0 \\ b & 0 & 0 & m & m & d & m & m \\ b & 0 & 0 & 0 & m & m & d & m \\ b & 0 & 0 & 0 & 0 & m & m & d \end{array}\right)$$

$\mathbf{A_u}$

  - Fast (band) solution by root free Cholesky decomposition
    - $\mathbf{A_u}=\mathbf{LDL}^{\dagger}$ (**L** triangular band, **D** diagonal)
    - fwd/bwd substitution $\mathbf{Lz}=\mathbf{b_u}$, $\mathbf{L^{\dagger}x_u}=\mathbf{D^{-1}z}$
  - Full solution by block mat. algebra
  - Effort to calculate
    - Solution **x** (→ $\Delta\mathbf{p}_i$): $\sim n_{par}\cdot(m+b)^2$
    - Bordered band part of $\mathbf{A}^{-1}$ (→ cov($\mathbf{p}_i$)): $\sim n_{par}\cdot(m+b)^2$
    - Full $\mathbf{A}^{-1}$ (→ Millepede): $\sim n_{par}^2\cdot(m+b)$
    - Inversion would be $\sim n_{par}^3$
  - For track fit ($\Delta\mathbf{p}_i$, cov($\mathbf{p}_i$)) time linear in n_{meas}

Implementation (III)

- Local track parameters $\mathbf{p} = (\frac{q}{p}, \mathbf{u}', \mathbf{u}),\ \ \mathbf{u} = (u_1, u_2),\ \ \mathbf{u}' = \frac{\partial \mathbf{u}}{\partial w}$
- Propagation of offset Δu using local linearization
  - with initial offset Δu_0, slope $\Delta u'_0$, curvature $\Delta q/p_0$:

(2) $$\Delta \mathbf{u} = \frac{\partial \mathbf{u}}{\partial \mathbf{u}_0} \Delta \mathbf{u}_0 + \frac{\partial \mathbf{u}}{\partial \mathbf{u}'_0} \Delta \mathbf{u}'_0 + \frac{\partial \mathbf{u}}{\partial \frac{q}{p_0}} \Delta \frac{q}{p_0} = \mathbf{J} \Delta \mathbf{u}_0 + \mathbf{S} \Delta \mathbf{u}'_0 + \mathbf{d} \Delta \frac{q}{p_0}$$

$(\mathbf{d}, \mathbf{S}, \mathbf{J}) = \frac{\partial \mathbf{u}}{\partial \mathbf{p}_0}$ taken from jacobian $\frac{\partial \mathbf{p}}{\partial \mathbf{p}_0}$

- Solve for slope $\Delta u'_0$:

(3) $$\Delta \mathbf{u}'_0 = \mathbf{S}^{-1} \left(\Delta \mathbf{u} - \mathbf{J} \Delta \mathbf{u}_0 - \mathbf{d} \Delta \frac{q}{p_0} \right)$$

Implementation (IV)

- With triplet $\mathbf{u}_-$, $\mathbf{u}_0$, $\mathbf{u}_+$ of offsets: $(\Delta\mathbf{u}_0 \rightarrow \mathbf{u}_0,\ \ \Delta\mathbf{u} \rightarrow \mathbf{u}_\pm)$

(4) $$\mathbf{u}_+ = \mathbf{J}_+\mathbf{u}_0 + \mathbf{S}_+\mathbf{u}'_+ + \mathbf{d}_+\Delta\tfrac{q}{p},\ \ \mathbf{u}'_{0(+)} = \mathbf{W}_+\left(\mathbf{u}_+ - \mathbf{J}_+\mathbf{u}_0 - \mathbf{d}_+\Delta\tfrac{q}{p}\right),\ \ \mathbf{W}_+ = \mathbf{S}_+^{-1}$$

(5) $$\mathbf{u}_- = \mathbf{J}_-\mathbf{u}_0 + \mathbf{S}_-\mathbf{u}'_- + \mathbf{d}_-\Delta\tfrac{q}{p},\ \ \mathbf{u}'_{0(-)} = \mathbf{W}_-\left(\mathbf{J}_-\mathbf{u}_0 - \mathbf{u}_- + \mathbf{d}_-\Delta\tfrac{q}{p}\right),\ \ \mathbf{W}_- = -\mathbf{S}_-^{-1}$$

- Kink $\mathbf{k}$ at $\mathbf{u}_0$

(6) $$\mathbf{k} = \mathbf{u}'_{0(+)} - \mathbf{u}'_{0(-)} = \mathbf{W}_+\mathbf{u}_+ - \left(\mathbf{W}_+\mathbf{J}_+ + \mathbf{W}_-\mathbf{J}_-\right)\mathbf{u}_0 + \mathbf{W}_-\mathbf{u}_- - \left(\mathbf{W}_+\mathbf{d}_+ + \mathbf{W}_-\mathbf{d}_-\right)\Delta\tfrac{q}{p}$$

- Interpolation
  - solve (6) for $\mathbf{u}_{int}=\mathbf{u}_0$ with $\mathbf{k}\equiv 0$, (4 or 5) using $u_0 = u_{int}$

(7) $$\mathbf{u}_{\text{int}} = \mathbf{N}\left(\mathbf{W}_+\mathbf{u}_+ + \mathbf{W}_-\mathbf{u}_-\right) - \mathbf{N}\left(\mathbf{W}_+\mathbf{d}_+ + \mathbf{W}_-\mathbf{d}_-\right)\Delta\tfrac{q}{p},\ \ \ \mathbf{N} = \left(\mathbf{W}_+\mathbf{J}_+ + \mathbf{W}_-\mathbf{J}_-\right)^{-1}$$

(8) $$\mathbf{u}'_{\text{int}} = \mathbf{W}_-\mathbf{J}_-\mathbf{N}\mathbf{W}_+\mathbf{u}_+ - \mathbf{W}_+\mathbf{J}_+\mathbf{N}\mathbf{W}_-\mathbf{u}_- - \left(\mathbf{W}_-\mathbf{J}_-\mathbf{N}\mathbf{W}_+\mathbf{d}_+ - \mathbf{W}_+\mathbf{J}_+\mathbf{N}\mathbf{W}_-\mathbf{d}_-\right)\Delta\tfrac{q}{p}$$

Broken lines vs Kalman filter (I)

- General Broken Lines with
  - One measurement only
  - External seed

$$\chi^2(\mathbf{p}_{bl}) = \mathbf{r}_1(\mathbf{p}_{bl})^t \mathbf{V}_{meas,1}^{-1} \mathbf{r}_1(\mathbf{p}_{bl}) + \mathbf{p}_{bl}^t \mathbf{V}_{seed}^{-1} \mathbf{p}_{bl}$$

- Normal equations

(9)

$$\left(\mathbf{V}_{seed}^{-1} + \mathbf{H}_1^t \mathbf{V}_{meas,1}^{-1} \mathbf{H}_1\right)\mathbf{p}_{bl} = \mathbf{H}_1^t \mathbf{V}_{meas,1}^{-1} \mathbf{r}_1,$$

$$\mathbf{H}_1 = \left(\frac{\partial \mathbf{r}_1}{\partial \mathbf{p}_{bl}}\right)$$

- Solution

(10)

$$\mathbf{p}_{bl} = \mathbf{V}_{bl}\left(\mathbf{H}_1^t \mathbf{V}_{meas,1}^{-1} \mathbf{r}_1\right)$$

$$\mathbf{V}_{bl} = \left(\mathbf{V}_{seed}^{-1} + \mathbf{H}_1^t \mathbf{V}_{meas,1}^{-1} \mathbf{H}_1\right)^{-1}$$

=

Kalman filtering (weighted mean formalism)

Fit parameter corrections: prediction $x_k^{k-1} = 0$

$$x_k = \mathbf{C}_k\left[\left(\mathbf{C}_k^{k-1}\right)^{-1} x_k^{k-1} + \mathbf{H}_k^t \mathbf{V}_k^{-1} m_k\right]$$

$$\mathbf{C}_k = \left[\left(\mathbf{C}_k^{k-1}\right)^{-1} + \mathbf{H}_k^t \mathbf{V}_k^{-1} \mathbf{H}_k\right]^{-1}$$

R. Frühwirth NIM A262(1987) 444-450, eqn (8b)

Broken lines vs Kalman filter (II)

- Track fitting
  - Kalman filter is externally seeded General Broken Lines fit with single (additional) measurement
  - General Broken Lines fit is optionally seedless Kalman filter adding all measurements in one filtering step
- Millepede
  - Simultaneous fit of **all** measurements as local fit
  - Can't use consecutive Kalman filter, need GBL

GBL software package

- Available from DESY SVN server
- Implementations in C++, Python, fortran
- Contains interface to Millepede-II
  - Write trajectories to MPII binary file
- For application examples use:
  - C++ version (V01-15-00, doxygen documentation)
    - Simple (track) and complex (decay) trajectories available
  - Curvilinear system for all points

$$\mathbf{U}_1 = \mathbf{X}_\perp = \begin{pmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{pmatrix}, \quad \mathbf{U}_2 = \mathbf{Y}_\perp = \begin{pmatrix} -\cos\varphi\sin\lambda \\ -\sin\varphi\sin\lambda \\ \cos\lambda \end{pmatrix}, \quad \mathbf{W} = \mathbf{Z}_\perp = \mathbf{T} = \begin{pmatrix} \cos\varphi\cos\lambda \\ \sin\varphi\cos\lambda \\ \sin\lambda \end{pmatrix}$$

Track fitting (I)

- For all points on trajectory
  - Create `GblPoint` with
    - Propagation jacobian $\mathbf{T}^{+}$ from previous point (**1** for first)
  - Optionally add 2D measurement **m** with
    - Projection matrix $\mathbf{P}=\partial\mathbf{m}/\partial\mathbf{u}$
      - with measurement directions $\mathbf{M}_i$: $\mathbf{P}^{-1} = \frac{\partial \mathbf{u}}{\partial \mathbf{m}} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix}^t \cdot \begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \end{pmatrix}$
    - Residual vector $\mathbf{r}_m$ (measurement - prediction)
    - Precision matrix $\mathbf{V}_m^{-1}$
      - can be singular (e.g. for 1D measurements)
      - internally diagonalized if necessary

Track fitting (II)

- Optionally add scatterer with
  - Initial kinks $\mathbf{k}_0$, usually zero
  - Diagonal of precision matrix $\mathbf{V}_k^{-1}$, $\quad \mathbf{V}_k = \begin{pmatrix} \theta_0^2 & 0 \\ 0 & \theta_0^2 \end{pmatrix}$
- Add point to list (`std::vector`)

★ GblTrajectory

- Created from list of `GblPoints` (and ext. seed)
- Fitted with `fit` method
  - M-estimators available for outlier down-weighting
- Optionally retrieve track parameter corrections or residuals and errors for `GblPoints`

Tracker alignment (I)

- Global derivatives $\partial r/\partial \mathbf{g}$
  - Variation of measurement residuals with global (=alignment) parameters
  - Needed by MPII to determine alignment par.
  - Have to be added to `GblPoints` with measurements
    - Labels: positive integers identifying parameters
    - Values: (non-zero) derivatives
  - Example: planar detector as rigid body
    - 3 displacements, 3 rotations,
    - $\mathbf{g}=(\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma)$, labels = 1 .. 6

Tracker alignment (II)

- Depend on prediction **p**, track slope **t** and (plane) normal **n**

$$\frac{\partial \mathbf{r}}{\partial \mathbf{g}} = \frac{\partial \mathbf{r}}{\partial \tilde{\mathbf{m}}}\frac{\partial \tilde{\mathbf{m}}}{\partial \mathbf{g}}, \quad \frac{\partial \mathbf{r}}{\partial \tilde{\mathbf{m}}} = \mathbf{1} - \frac{\mathbf{t}\cdot\mathbf{n}^t}{\mathbf{t}\cdot\mathbf{n}} = \left(\delta_{ij} - \frac{t_i \cdot n_j}{\mathbf{t}\cdot\mathbf{n}}\right) \qquad \frac{\partial \tilde{\mathbf{m}}}{\partial \mathbf{g}} = \begin{pmatrix} 1 & 0 & 0 & 0 & -z_p & y_p \\ 0 & 1 & 0 & z_p & 0 & -x_p \\ 0 & 0 & 1 & -y_p & x_p & 0 \end{pmatrix}$$

- Local system (u,v,w) in plane:
  w = **n**, $w_p = 0$ (corresponds to z_p)

$$\frac{\partial \mathbf{r}}{\partial \tilde{\mathbf{m}}} = \begin{pmatrix} 1 & 0 & -\frac{\partial u}{\partial w} \\ 0 & 1 & -\frac{\partial v}{\partial w} \\ 0 & 0 & 0 \end{pmatrix}$$

- MPII binary file
  - Input data for Millepede-II
  - Fitted `GblTrajectory` can be written directly with method `milleOut`

Summary

- General Broken Lines
  - Constructed from list of measurements and scatterers connected by propagation jacobians
  - Fast fitting ($\mathbf{x} \sim n_{scat}$, (full) $\mathbf{V} \sim n_{scat}^2$)
  - Well suited as local (track) fit for Millepede-II
- GBL maintained by Terascale Alliance
  - Implemented in C++, Python, fortran
  - Includes interface to Millepede-II

Backup

Global derivatives, alternative fits

Global derivatives (I)

- Global derivatives (for planar detectors)
  - Measurement
    - For a measurement **m** with a prediction **p** (x_p,y_p,z_p) from the track model the effects of displacements (Δx, Δy, Δz) and small rotations (α,β,γ) (around axes at origin) are in first order:

$$\tilde{\mathbf{m}} = \mathbf{m} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ z_p \\ -y_p \end{pmatrix} + \beta \begin{pmatrix} -z_p \\ 0 \\ x_p \end{pmatrix} + \gamma \begin{pmatrix} y_p \\ -x_p \\ 0 \end{pmatrix}$$

Global derivatives (II)

- Prediction
  - Linearizing the track model at the intersection point with the (nominal) measurement plane, the prediction **p** depends on the position $\mathbf{x}_i$ and track direction **t** at that point:

$$\mathbf{p}(\Delta s) = \mathbf{x}_i + \mathbf{t} \cdot \Delta s$$

  - With the normal **n** to the measurement plane the intersection of of the linearized track with the distorted measurement is given by:

$$0 = (\tilde{\mathbf{m}} - \mathbf{p}) \cdot \mathbf{n} = (\tilde{\mathbf{m}} - \mathbf{x}_i) \cdot \mathbf{n} - \mathbf{t} \cdot \mathbf{n} \cdot \Delta s \text{ or } \Delta s = \frac{(\tilde{\mathbf{m}} - \mathbf{x}_i) \cdot \mathbf{n}}{\mathbf{t} \cdot \mathbf{n}}$$

Global derivatives (III)

- Residuals
  - The residual **r** at the intersection of the linearized track with the distorted measurement is:

$$\mathbf{r} = \tilde{\mathbf{m}} - \mathbf{p} = \tilde{\mathbf{m}} - \mathbf{x}_i - \mathbf{t} \cdot \Delta s = \tilde{\mathbf{m}} - \mathbf{x}_i - \mathbf{t}\frac{(\tilde{\mathbf{m}} - \mathbf{x}_i) \cdot \mathbf{n}}{\mathbf{t} \cdot \mathbf{n}}$$

- Derivatives
  - The derivatives of the residual versus the displacements and rotations as global parameters **g** are:

$$\frac{\partial \mathbf{r}}{\partial \mathbf{g}} = \frac{\partial \mathbf{r}}{\partial \tilde{\mathbf{m}}} \frac{\partial \tilde{\mathbf{m}}}{\partial \mathbf{g}}, \quad \frac{\partial \mathbf{r}}{\partial \tilde{\mathbf{m}}} = \mathbf{1} - \frac{\mathbf{t} \cdot \mathbf{n}^t}{\mathbf{t} \cdot \mathbf{n}} = \left(\delta_{ij} - \frac{t_i \cdot n_j}{\mathbf{t} \cdot \mathbf{n}} \right)$$

$$\frac{\partial \tilde{\mathbf{m}}}{\partial \mathbf{g}} = \begin{pmatrix} 1 & 0 & 0 & 0 & -z_p & y_p \\ 0 & 1 & 0 & z_p & 0 & -x_p \\ 0 & 0 & 1 & -y_p & x_p & 0 \end{pmatrix}$$

- Local system (u,v,w) in plane
  - **w** = **n**, $w_p = 0$ (corresponds to z_p)

$$\frac{\partial \mathbf{r}}{\partial \tilde{\mathbf{m}}} = \begin{pmatrix} 1 & 0 & -\frac{\partial u}{\partial w} \\ 0 & 1 & -\frac{\partial v}{\partial w} \\ 0 & 0 & 0 \end{pmatrix}$$

Alternative (linear least squares) fits (I)

- Same trajectory $\mathbf{u}(s)$, different parameters $\mathbf{x}$
  - Use multiple scattering kinks $\mathbf{k}_i$ ("BreakPoints")

$$\mathbf{u}_{i+1} = \frac{\partial \mathbf{u}_{i+1}}{\partial \mathbf{u}_i}\mathbf{u}_i + \frac{\partial \mathbf{u}_{i+1}}{\partial \mathbf{u}'_i}\left(\mathbf{u}'_i + \mathbf{k}_i\right) + \frac{\partial \mathbf{u}_{i+1}}{\partial \frac{q}{p_i}}\Delta\frac{q}{p_i}, \quad \mathbf{u}'_{i+1} = \frac{\partial \mathbf{u}'_{i+1}}{\partial \mathbf{u}'_i}\left(\mathbf{u}'_i + \mathbf{k}_i\right) + \frac{\partial \mathbf{u}'_{i+1}}{\partial \frac{q}{p_i}}\Delta\frac{q}{p_i}$$

Change of direction at each (thin) scatterer: $\mathbf{u}' \rightarrow \mathbf{u}'+\mathbf{k}$

- $\mathbf{x}=(\Delta q/p_1, \mathbf{u}'_1, \mathbf{u}_1, \mathbf{k}_1, .., \mathbf{k}_{nscat-1})$
- All scatterers in front of measurement $\mathbf{m}_i$ contribute to prediction $\mathbf{u}_{int,i}$

$$\frac{\partial \mathbf{u}_{\text{int},i}}{\partial \mathbf{k}_j} \approx \max(0, s_i - s_j)\cdot\mathbf{1}$$

- Matrix A of linear equation system is full matrix
- Effort for solution ~ n_{par}^3

Alternative fits (II)

- Same trajectory **u**(s), different parameters **x**
  - Put multiple scattering into covariance matrix, **k**≡0

$$\mathbf{V}_r = \mathbf{V}_m + \left(\frac{\partial \mathbf{r}}{\partial \mathbf{k}}\right) \mathbf{V}_k \left(\frac{\partial \mathbf{r}}{\partial \mathbf{k}}\right)^t, \quad \mathbf{r} = \mathbf{m} - \mathbf{P} \cdot \mathbf{u}_{\mathrm{int}}, \quad \chi^2(\mathbf{x}) = (\mathbf{r})^t\, \mathbf{V}_r^{-1} (\mathbf{r})$$

    - $\mathbf{x}=(\Delta q/p_1, \mathbf{u}'_1, \mathbf{u}_1)$, $n_{par}=5$
    - $\mathbf{V}_r$ is full matrix of size n_{meas},
      residuals **r** are correlated, not usable with MillePede
    - Need $\mathbf{V}_r^{-1}$, effort for solution at least ~ n_{meas}^3