Kalman Filter, magnetic field mapping, etc

- **Kalman Filter using the weight/information formalism**
  - simpler and more intuitive: avoiding problems with big values
  - tool to predict the errors of reconstructed track (cf Renato's work)
  - extension to prediction of sensitivity to individual measurements
  - tool to make projections or changes of geometry
  - possible simplifications (« what matters » principle)

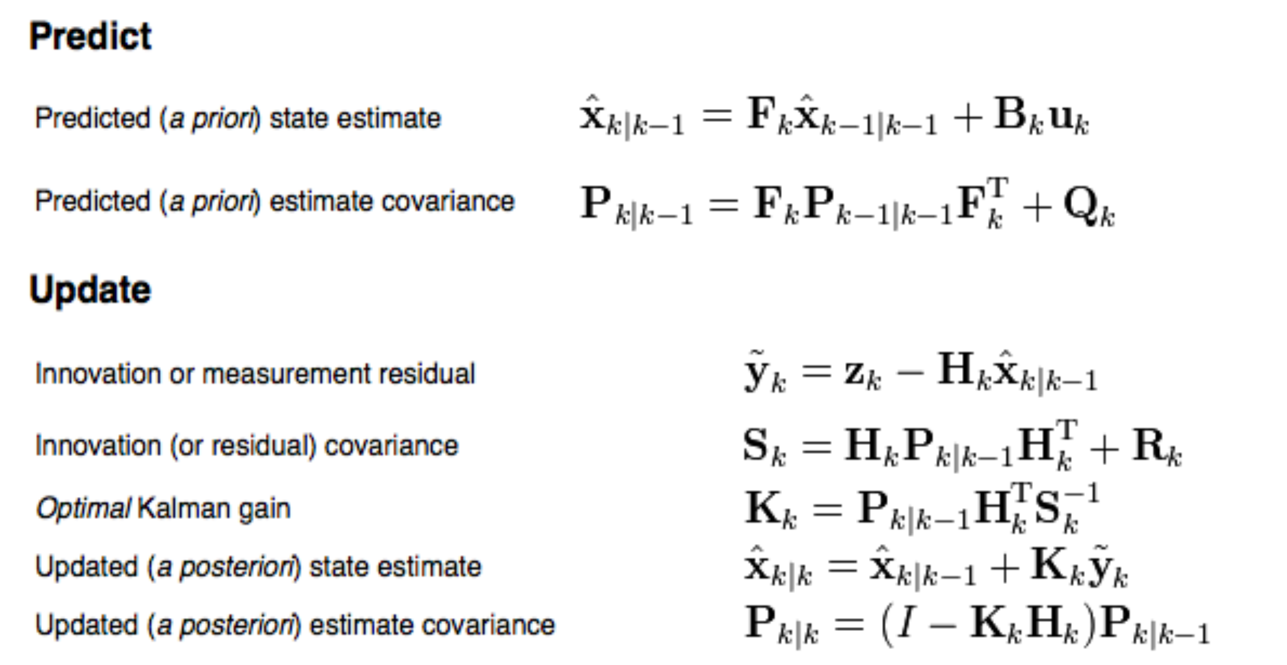
- **Magnetic field map**:
  - using triplets of polynomials for Bx,By,Bz obeying Maxwell equations
  - completing the measurements with « peripheric » permanent probes ?

- **Correlations between magnetic corrections and alignment**
  - evidence
  - possible solution with a joint fit of align. param. and magnetic correction ?
  - correction a posteriori (complement to the corrections « à la Needham »)

basic tool for track fitting :

Kalman Filter (*progressive* fitting method)

found in many textbooks… (here : Wikipedia)



Predict

Predicted (*a priori*) state estimate $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$

Predicted (*a priori*) estimate covariance $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$

Update

Innovation or measurement residual $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$

Innovation (or residual) covariance $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$

Optimal Kalman gain $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$

Updated (*a posteriori*) state estimate $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

Updated (*a posteriori*) estimate covariance $\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$

+ even more complicated expression for the “smoothing”

we will present something equivalent (and hopefully more intuitive !) and try to go further

gaussians in nD space

$G(\mathbf{x}) = K \exp\left(-\Sigma\, W_{ij}\, (x_i-\mu_i)\, (x_j-\mu_j)/2\right)$ $K^2 = \det(W)/(2\pi)^n$

covariance matrix $C = W^{-1}$

combining gaussians:

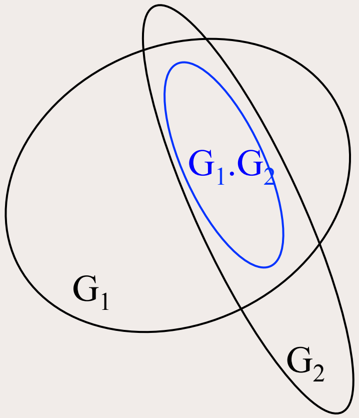
product: $(\boldsymbol{\mu}_1,W_1) \,.\, (\boldsymbol{\mu}_2,W_2) \rightarrow (W_1+W_2)^{-1}.(W_1\boldsymbol{\mu}_1+W_2\boldsymbol{\mu}_2)$, W_1+W_2

(combining *independent* informations: addition of **weight** matrices)

the new center is a « barycenter with matricial weights »

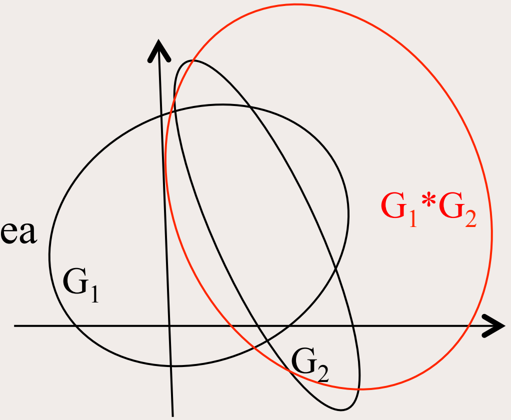
convolution: $(\boldsymbol{\mu}_1,W_1) * (\boldsymbol{\mu}_2,W_2) \rightarrow \boldsymbol{\mu}_1+\boldsymbol{\mu}_2$, $(W_1^{-1}+W_2^{-1})^{-1}$

(combining *independent* biases: addition of **covariance** matrices)



1σ contours

quantitatively:
information = 1/area
(1/volume in nD)



weight matrix (“information”) formalism

- state vector p (actually: *deviation from reference trajectory* $\delta x, \delta y, \delta t_x, \delta t_y, \delta(q/p)$)
- weight matrix $W = C^{-1}$ (if rank = 5 ; W may have rank < 5)
- propagation: $p_{propag} = D.p$ $\quad W_{propag} = (D^{-1})^T.W.(D^{-1})$ (D : jacobian matrix)
  $(W.p)_{propag} = (D^{-1})^T.(W.p)$
- noise (mult. scatt.) : $W' = (W^{-1}+S)^{-1} = (1+W.S)^{-1}.W$
- adding information : $(W_{pred}+W_{meas}).p_{upd} = W_{pred}.p_{pred} + W_{meas}.p_{meas}$
  $W_{upd} = W_{pred} + W_{meas}$ $\quad (W.p)_{upd} = (W.p)_{pred} + (W.p)_{meas}$

advantages:

- the meaning of operations is fully intuitive (e.g. addition of independent informations)
- all operations may be done whatever the rank of the W matrices

no need to « regularize » covariance matrices when beginning the Filter.

never need to solve a « singular » system: e.g., computing an interpolation/ extrapolation, updating a χ^2, etc are requested only with « complete » states

- the « noise » step may be simplified if the matrix S is reduced to (t_x, t_y) terms, e.g. $S = \mathrm{diag}(0, 0, \varepsilon_x^2, \varepsilon_y^2, 0)$ in the small angle approximation

- if W has rank < 5, the « barycenter » is degenerate: no problem !
- the « smoother is just a local interpolation: combination of a forward and a backward filter and a forward one, both up to this point

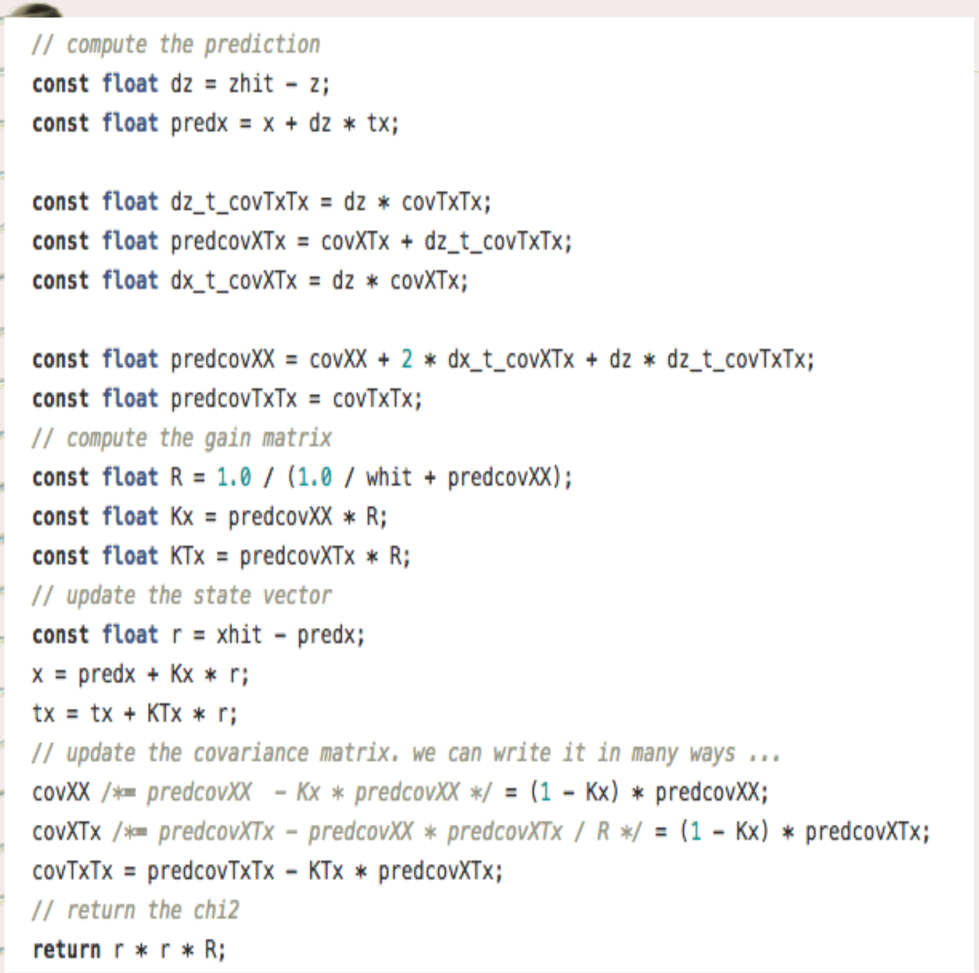
a problem with the standard Kalman Filter

(work with Dorothea)

trying to implement the Kalman Filter included in PrPixelTracking (Velo) in single precision on a GPU:

- *discrepancies between the GPU and the CPU results, and between them and the weight/information algorithm, when applied to the same data*
- *more precisely: the discrepancies (on fitted position/slope, covariance matrix, chi2) decrease with the number of points in the track*
- *agreement between all versions in double precision, and between single and double with the weight algorithm*
- *the discrepancies increase with the initial value given to cov(Tx,Tx) and cov(Ty,Ty) at the beginning of the loop on points*

the origin of the problem



```
// compute the prediction
const float dz = zhit - z;
const float predx = x + dz * tx;

const float dz_t_covTxTx = dz * covTxTx;
const float predcovXTx = covXTx + dz_t_covTxTx;
const float dx_t_covXTx = dz * covXTx;

const float predcovXX = covXX + 2 * dx_t_covXTx + dz * dz_t_covTxTx;
const float predcovTxTx = covTxTx;
// compute the gain matrix
const float R = 1.0 / (1.0 / whit + predcovXX);
const float Kx = predcovXX * R;
const float KTx = predcovXTx * R;
// update the state vector
const float r = xhit - predx;
x = predx + Kx * r;
tx = tx + KTx * r;
// update the covariance matrix. we can write it in many ways ...
covXX /*= predcovXX  - Kx * predcovXX */ = (1 - Kx) * predcovXX;
covXTx /*= predcovXTx - predcovXX * predcovXTx / R */ = (1 - Kx) * predcovXTx;
covTxTx = predcovTxTx - KTx * predcovXTx;
// return the chi2
return r * r * R;
```

at first point $C_{xx}= \sigma^2$, $C_{TxTx}=$ **Big**
(in this code: **Big** *= 1)*
the loop (pred, upd, noise) begins at the ***second*** point with a ***nearly singular*** predicted covariance :
$C'_{xx}= \sigma^2+\mathbf{Big}^2\Delta z^2$
$C'_{xTx}= \mathbf{Big}\,\Delta z$, $C'_{TxTx}= \mathbf{Big}$
the « gain » business mixes **Big** and real quantities ➔ rounding errors !

here: making **Big** ➔ ∞ in the results after updating at point 2:
$x = x_2$ $\quad Tx = (x_2\text{-}x_1)/\Delta z$
$\text{cov} = (\sigma^2, \sigma^2/\Delta z, 2\sigma^2/\Delta z^2)$
$\chi^2 = 0$
the KF machinery was useless !

simple solution

with the first two points: simple straight line fit, without noise linear system, with $w_k = 1/\sigma_k^2$:

$$\underset{\text{weight matrix}}{\begin{bmatrix} \Sigma w_k & \Sigma w_k z_k \\ \Sigma w_k z_k & \Sigma w_k z_k^2 \end{bmatrix}} \underset{\text{state}}{\begin{bmatrix} X \\ Tx \end{bmatrix}} = \underset{\text{information vector}}{\begin{bmatrix} \Sigma w_k x_k^{meas} \\ \Sigma w_k z_k x_k^{meas} \end{bmatrix}}$$

this is exactly equivalent to the limit obtained with **Big** $\to \infty$

then: begin the KF machinery **after** including the **second** point

- add the noise
- propagate to next point (prediction)
- add the next point and increment χ^2

etc…

no precision problems in the next steps (the covariance matrices does not include artificial terms)

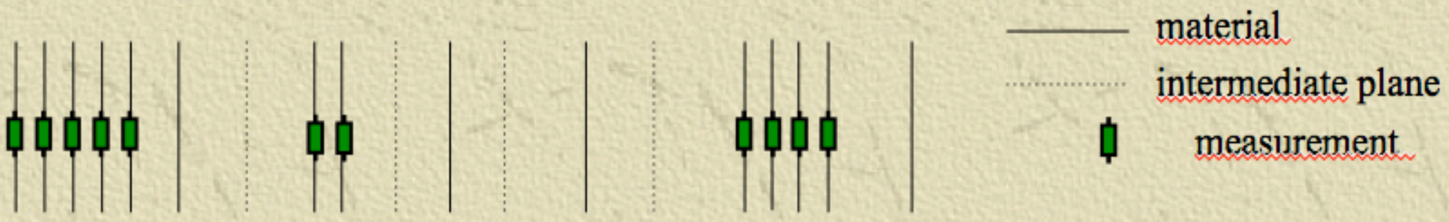
the standard machinery may be used safely with everything in 'float'

fast estimation of errors of track fit and sensitivity to individual measurements (without MC data)

standalone code with simplified geometric model (cf presentation of 2015, June 2)

- set of z-planes: position, thickness; if measurement: nature(x,y,stereo) and error
- uniform field along y between two planes; piecewise parabolic model in zx plane
- small $|t_x|$ and $|t_y|$ along the trajectory

(the framework could easily accept extensions of the last two conditions)



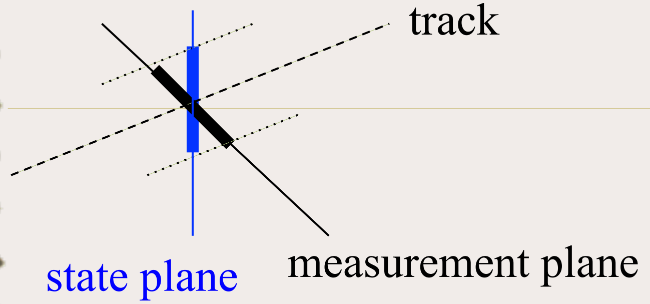
principle of the computation: perform a « stateless » Kalman Filter, doing the operations (forward +backward+interpolations) on *matrices*, not on state vector

➔ evaluate the covariance matrix on $(x,y,t_x,t_x,q/p)$ at each point

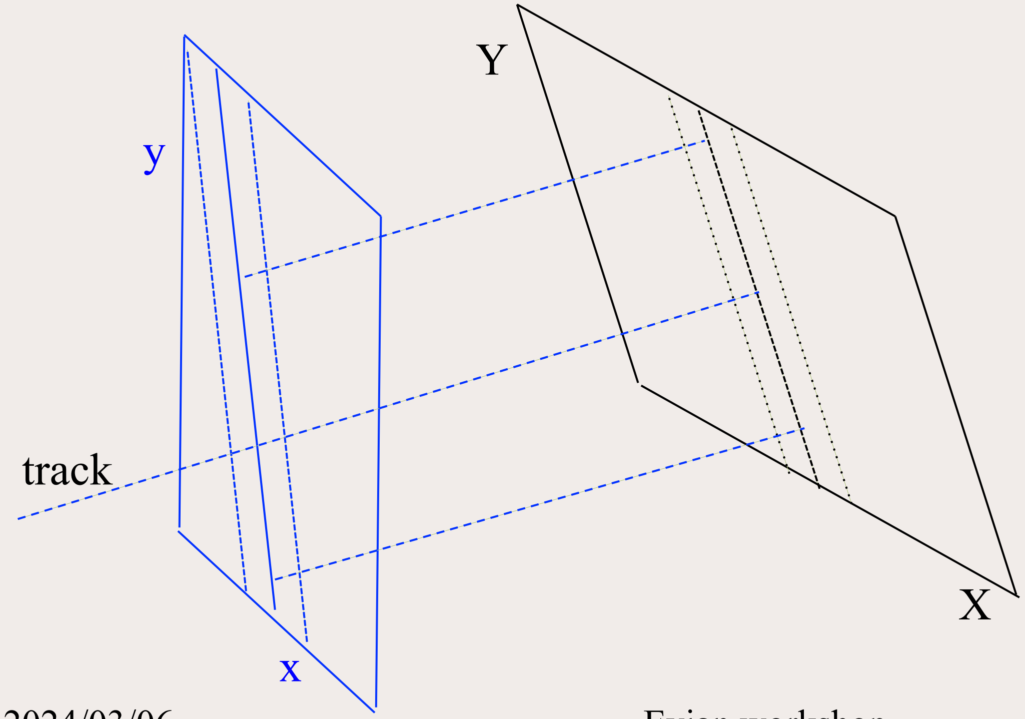
no need for explicit measured values in the planes

new feature: all KF operations are *linear* transformations of *linear* functions of the measurements ➔ on can compute also the ***sensitivity*** of the fitted quantities (including interpolations) to each individual measurement, hence the impact of accidental or systematic errors (e.g. misalignment)

projecting and changing geometry



to be included in the KF, the measurement error in an oblique plane is projected onto the state plane along the local track direction



a measurement (X,σ_X) in an oblique detector is projected onto the state plane along the local track direction as a virtual measurement of $\alpha x+\beta y$ in the state plane. α, β and the projected error depend on the geometry of both the planes and the track

sensitivity to individual measurements

linear approximation around the reference trajectory:
the KF is a squence of linear operations on the state vector
each measurement contributes linearly to the fitted sate

➔ at any step, the fitted parameters (deviations from reference) depend linearly on the measurements previously included

➔ one can compute a « matrix of sensitivity » of parameters to measurements

possible applications:

- estimate « what matters » for a given physical purpose
- sensitivity of the fitted parameters to misalignments

fitting an magnetic field map with triplets of polynomials (B_x, B_y, B_z) satisfying the Maxwell equations

- H.Wind (master of the sixties)
  polynomials classified by degrees and parities in x,y,z (J. of Comput. Phys.(1968)
  combinations of products of trigo/hyperbolic functions (NIM A 89 (1968)

- Another construction of polynomials based on spherical harmonics

div(B) = 0 and curl(**B**) = **0** is equivalent to:

B = **grad**(Φ) with Φ *harmonic*

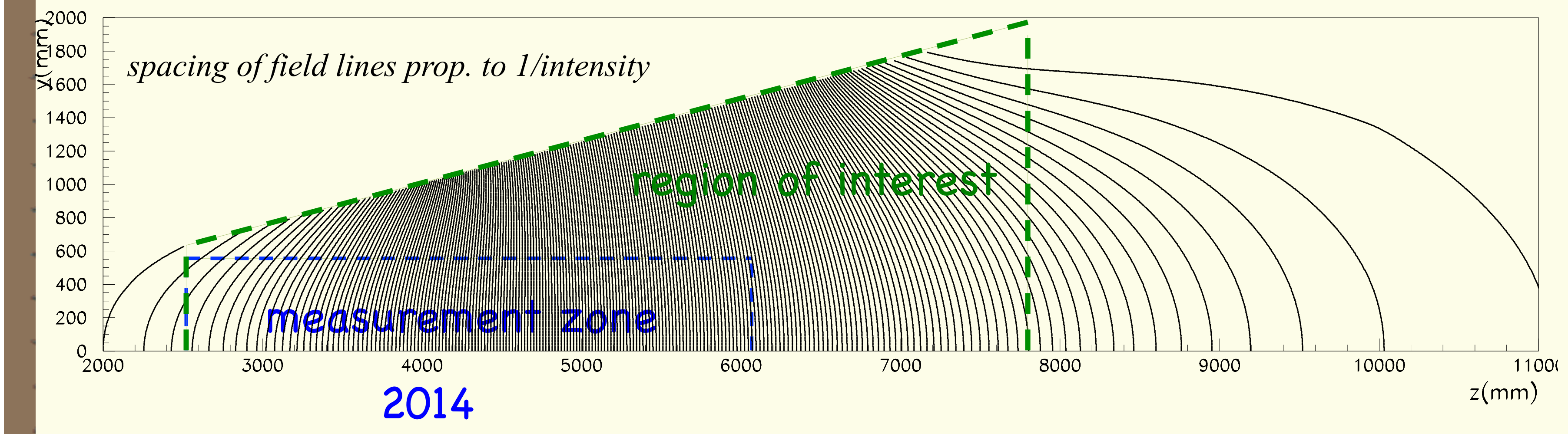
$r^l Y_{lm}(\theta,\varphi)$ is harmonic and polynomial of degree l in x,y,z coordinates

➔ taking the real and imaginary parts gives a solution with defined parities in x,y,z (useful to constrain the solution to symmetries of the system)

more advanced: use large degrees for the « regular » components (expected symmetries), and low degrees for « irregular » ones (perturbations supposed to be small)

note: the LHCb field is too complex to be globally fitted with a reasonable degree

magnetic field map measurements in 2010 and 2014



the measurement zone does not fully cover the region of interest:

extrapolation procedure in both LHCb-INT-2012-012 and LHCb-INT-2015-034:

global displacement (translation+rotation) + scaling factor

fits well to the data within the measurement zone, but no guarantee to match the remaining space

within technical constraints: can we extend the zone at large z (up to ~8000) ?

another problem when computing the new map on the grid**:** abnormal fluctuations for large |y/z| (especially on the top edge); due to interpolation procedure ?

how to obtain a more exhaustive evaluation ?

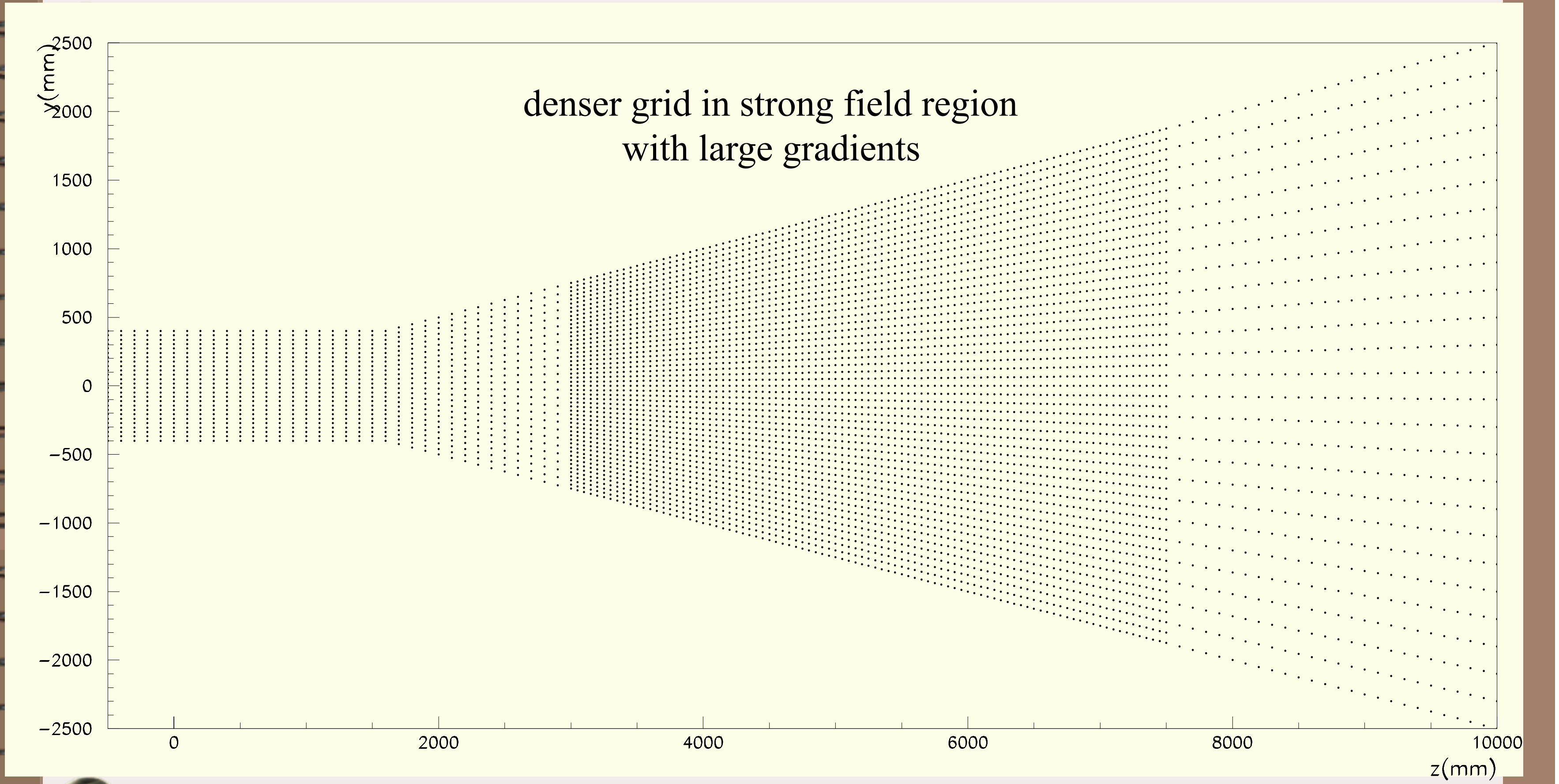
- the « regular » components may be computed from the description of the magnet
- the measurements suggest that there are perturbations (with left/right and top/ down asymmetries)
- it is impossible to make an exhaustive description of all potentially magnetic materials in the environment

if the (small) irregular component is due to remote elements, it is probably smooth within the geometrical domain of the tracking detectors: it could be described by low degree Maxwell-compatible polynomials

a set of Hall probes around this domain could give an input for such a fit
another possible advantage: providing a « slow control » of the field (long term evolution and reactions to changes of polarity)

building a « projective » map ?

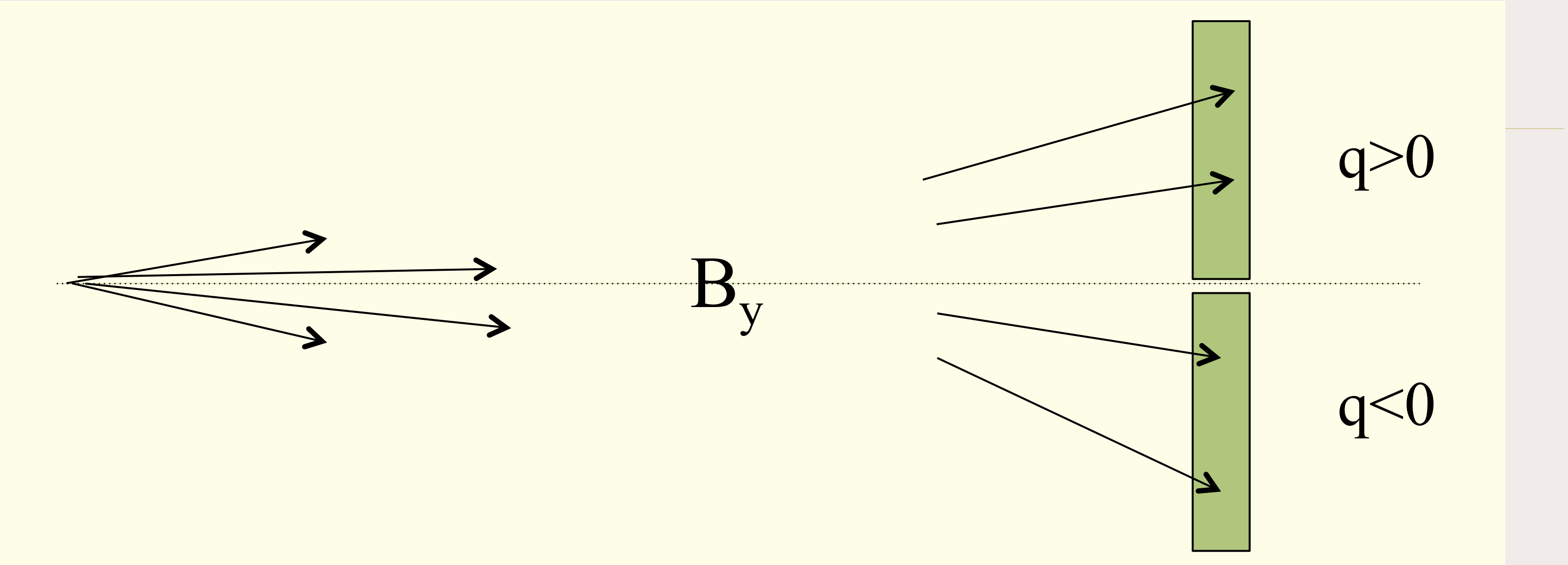
(cf talk of june 2016)



denser grid in strong field region with large gradients

for any point along physical trajectories: interpolation from points inside the acceptance
no pollution from external points, no artificial fluctuations

an example of correlation between field and alignment



- changing B_y is partially compensated by opposite displacements of the half chambers along x axis
- if no separate degrees of freedom: partial compensation by a translation along z axis
- more generally: there are correlations between field distortions and geometrical displacements
- *the alignment parameters may depend on the selected momenta*

how to disentangle the sources of deviations ?

a “possible” mathematical solution

P.B. NIM A 902 (2018) 33-44

principle:
fitting at the same time field corrections (e.g. coefficients of Maxwell-compatible polynomials) and alignment parameters on a large set of tracks with various momenta and trajectories to disentangle the dependences

toy model: two blocks of detectors (upstream/downstream) with 6 relative alignment parameters (translation+rotation)

good results, but did not work when applied on real data (more complex internal alignment needed in each block ?)

less demanding: extended correction of momentum scale

correction of momenta *a posteriori* to account for

1) field map deviations
2) misalignments

point 1) is addressed in 024 JINST 19 p02008 (à la Needham)

principle: for a given direction, a field discrepancy results in a modification of the momentum scale, reflected in the invariant mass of X ➔ m^+ $m-$ decays (assuming that both daughters are roughly the same domain in (t_x,t_y))

point 2) possible extension: consider the *momentum balance* between m^+ and $m-$

ingredient: a misalignment results in a shift on q/p: p is replaced by $p+\varepsilon qp^2$
where ε is the result of all misalignments along the (t_x,t_y) line

simplified computation for massless daughters (similar qualitative result with masses)

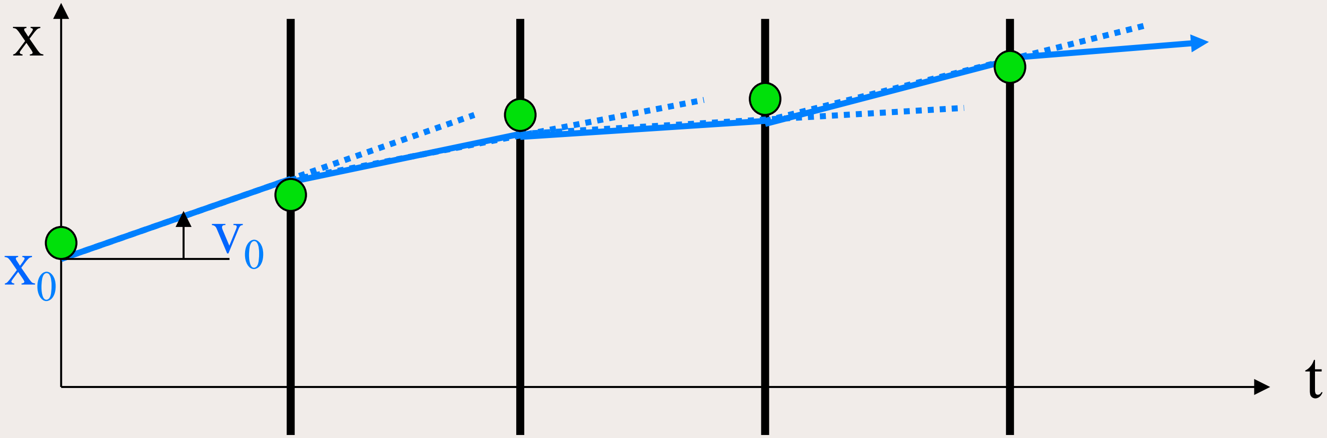
$m_X{}^2 = (p_1+p_1)^2-(\mathbf{p}_1+\mathbf{p}_2)^2 = 2p_1p_2(1-\cos(\mathbf{p}_1.\mathbf{p}_2))$

$(p_1+\varepsilon p_1{}^2)\ (p_2-\varepsilon p_2{}^2) = p_1p_2\ (1+\varepsilon(p_1-p_2))$

the shift is proportional to p_1-p_2

proposition: for a direction (t_x,t_y), evaluate the dependence on p_1-p_2 in addition to a scale factor, and introduce a correction including this dependence

linear problem with 2 parameters (movement with "noisy" speed)



initial conditions: x_0, v_0 (to be estimated)

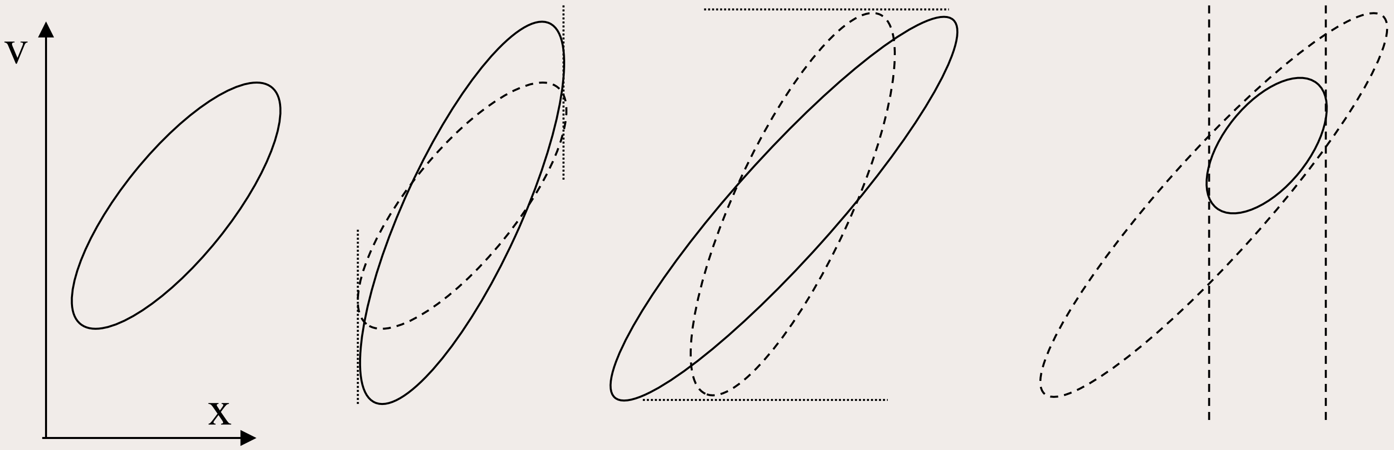
at each time step Δt :

- measurement of x (error ε_k , variance σ^2)
- random variation ζ_k of v_k (variance ρ^2)
- displacement $v_k.\Delta t$

$x_k^{mes} = x_0 + (v_0 + \zeta_1)\,\Delta t + (v_0 + \zeta_1 + \zeta_2)\,\Delta t + \ldots + \varepsilon_k$

→ *correlation (x_k^{mes}, x_j^{mes}) through the ζ_i*

progressive fit: one step “on-the fly” in the (x,v) plane



previous measurements:
state vector $p_k(x_k,v_k)$
cov. matrix $C_{k\backslash}$
weight matrix $W_k=C_k^{-1}$

noise on v
$C_k^n = C_k+N$
$$N = \begin{pmatrix} 0 & 0 \\ 0 & \rho^2 \end{pmatrix}$$

degraded information
$W_k^n = (C_k^n)^{-1}$

propagation
$p'_k\ (x_k-v_k\Delta t,\ v_k)$
$C' = D.C_k^n.D^t$
$$D = \begin{pmatrix} 1 & -\Delta t \\ 0 & 1 \end{pmatrix}$$
$W' = (D^t)^{-1}.W_k^n.D$

combination with a new measurement p_{k-1}
$W' = (C')^{-1}$, W_{k-1} $[= (C_{k-1})^{-1}\ ?]$
$(W' + W_{k-1})\ p'' = W'\ p'_k + W_{k-1}p_{k-1}$
$C''(p'') = (W' + W_n)^{-1}$
information is gained

parameterized propagation

idea: instead of using RK extrapolation for every track, precompute formulae to get a faster execution

principle:

- chose a few reference surfaces that will contain « nodes » of the Kalman Filter.
- to go from the initial surface Σ_i to the final one Σ_f , express the state vector $\mathbf{S}_f$ on Σ_f through analytical of tabulated functions of the components of the state vector $\mathbf{S}_i$ on Σ_i

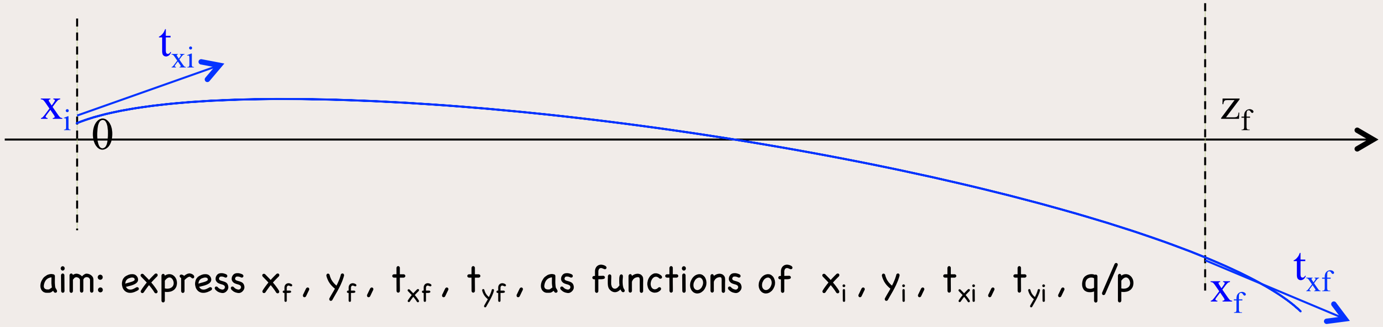
guiding criteria

- at infinite momentum, the trajectory is a straight line
- so, we can try an expansion in powers of q/p of $\Delta\mathbf{S}_f$, the difference between $\mathbf{S}_f$ and the straight line extrapolation
- the precision should be small compared to the other sources of error (mainly multiple scattering)
- the phase space may be reduced for trajectories close to the origin (particles for physics analysis)

first example in the « endcap » description (x, y, t_x, t_y, q/p at fixed z): propagate ***from*** z_i***=0 to*** z_f

- t_x and t_y are bounded by the acceptance ;

- x_i and y_i are small, so terms at first order in x_i,y_i are sufficient



aim: express x_f , y_f , t_{xf} , t_{yf} , as functions of x_i , y_i , t_{xi} , t_{yi} , q/p