Tracker Analysis Progress

Edward Santos
on behalf of the Tracker Group

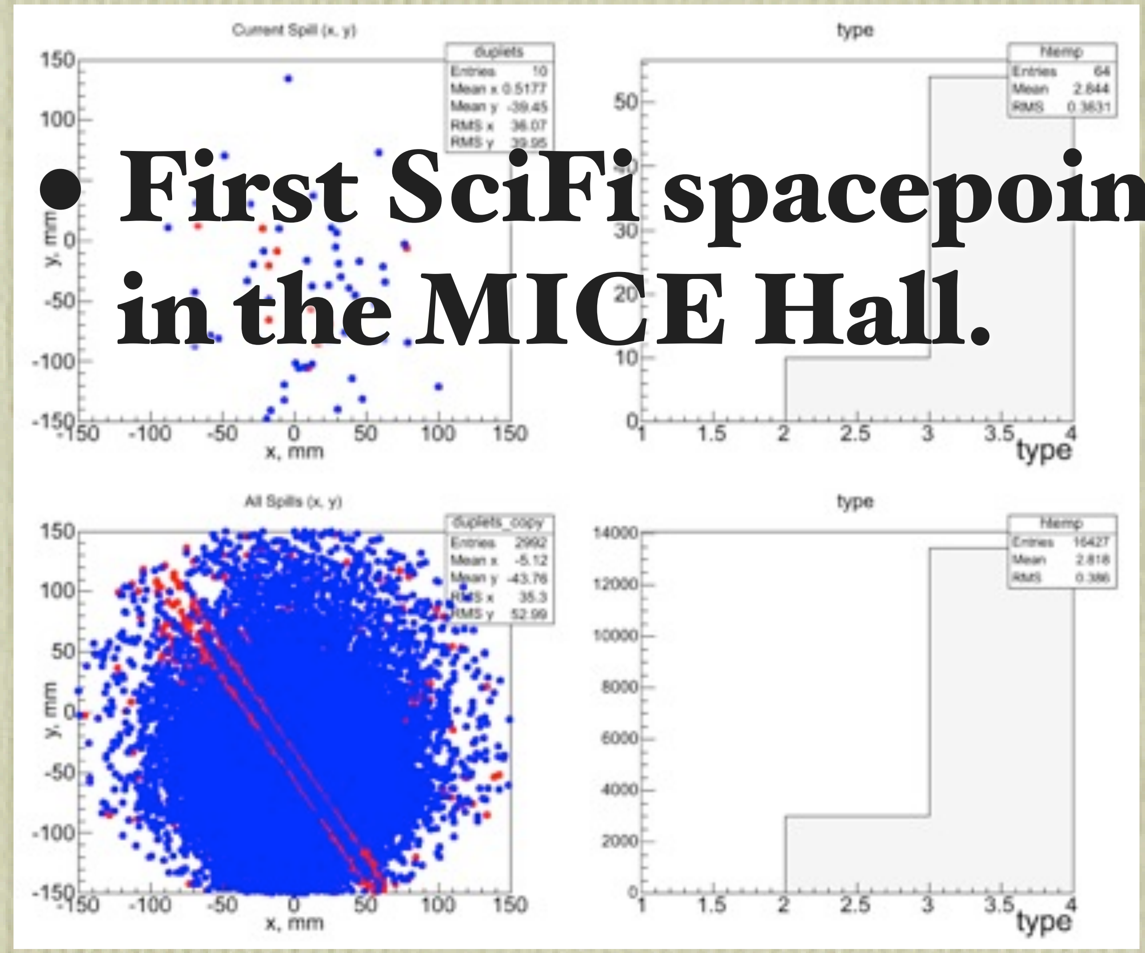
Imperial College
London

Overview

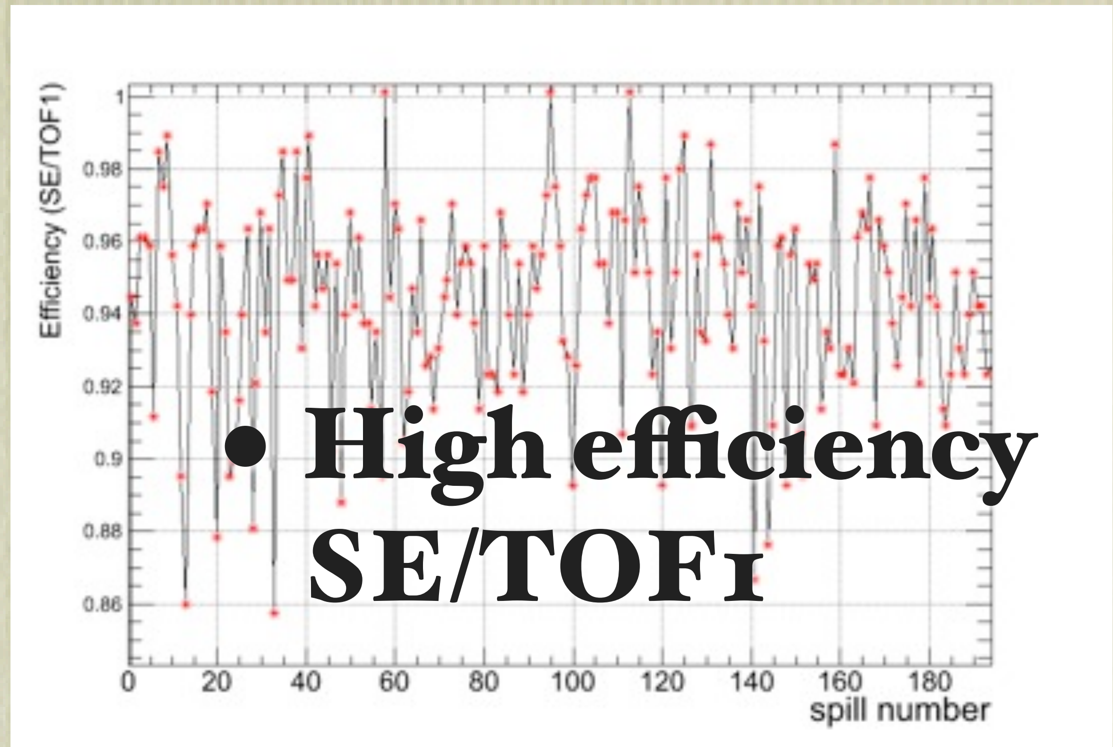
- May Run/Cosmic Run (brief)

- Track Fitting with the Kalman Filter

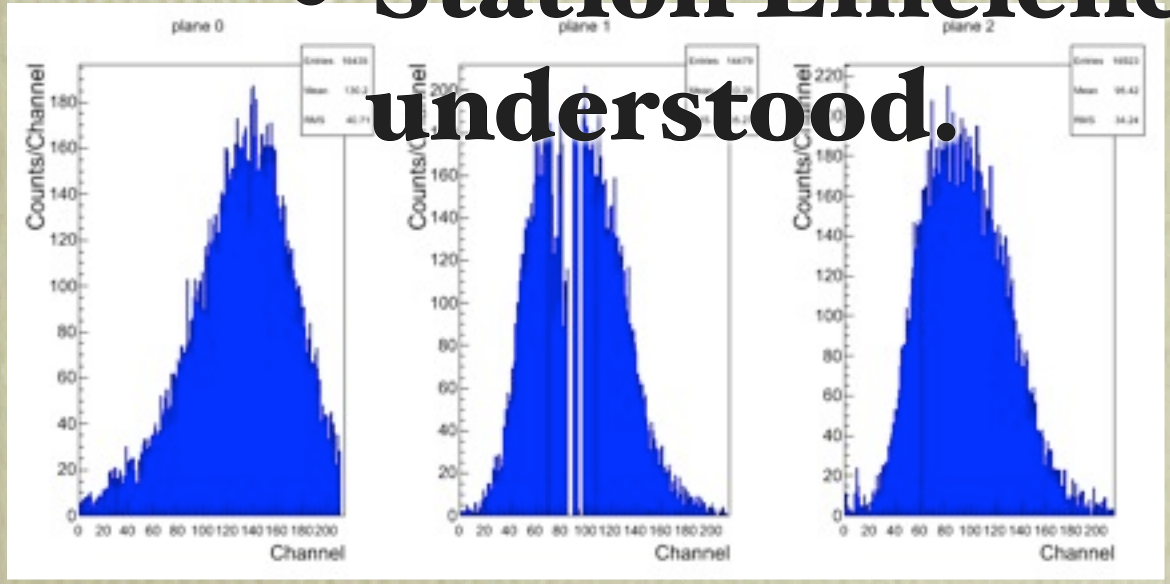
May Run



- **First SciFi spacepoints in the MICE Hall.**



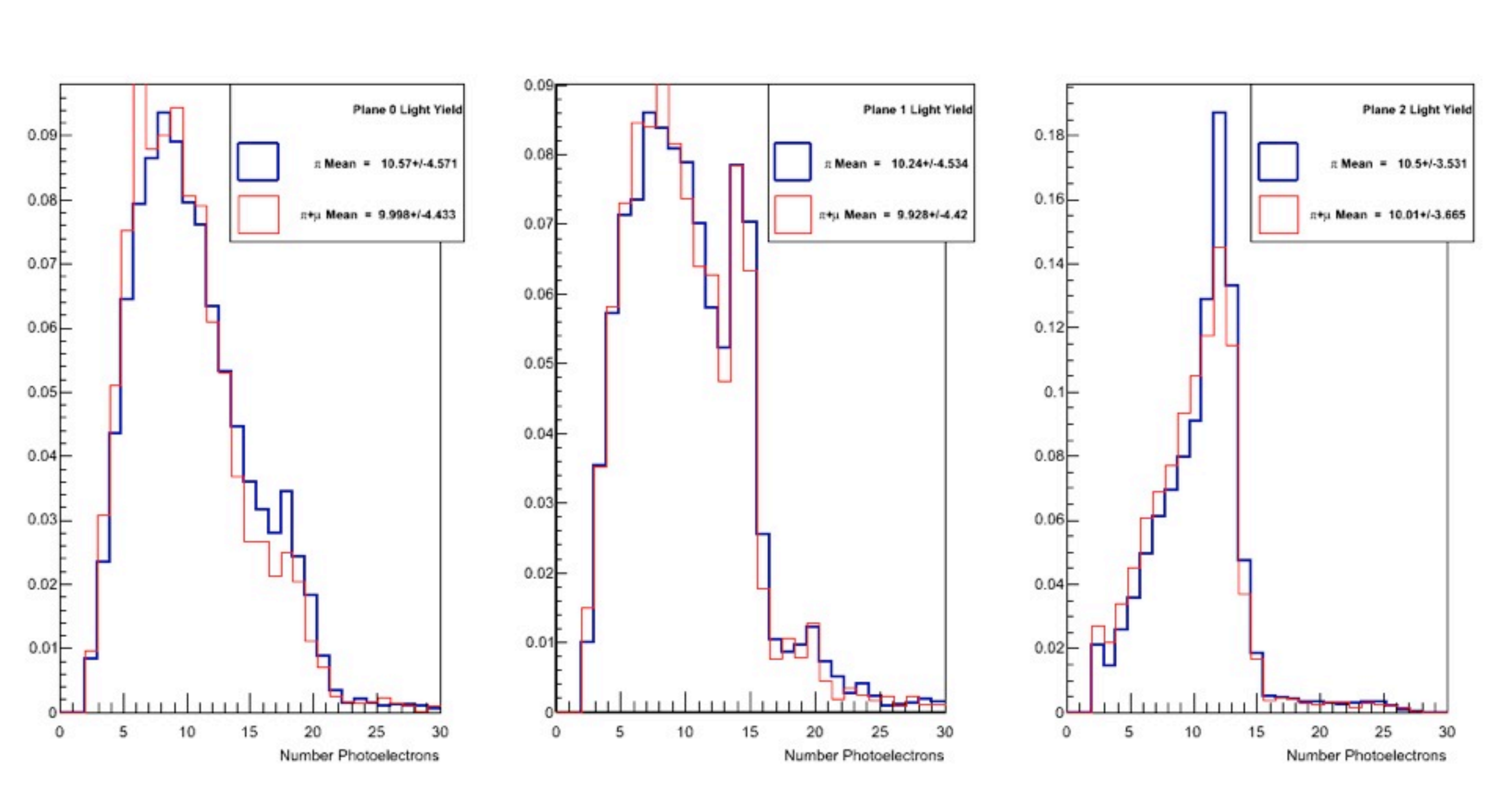
- **High efficiency SE/TOF1**



- **Station Efficiency understood.**

Light Yield

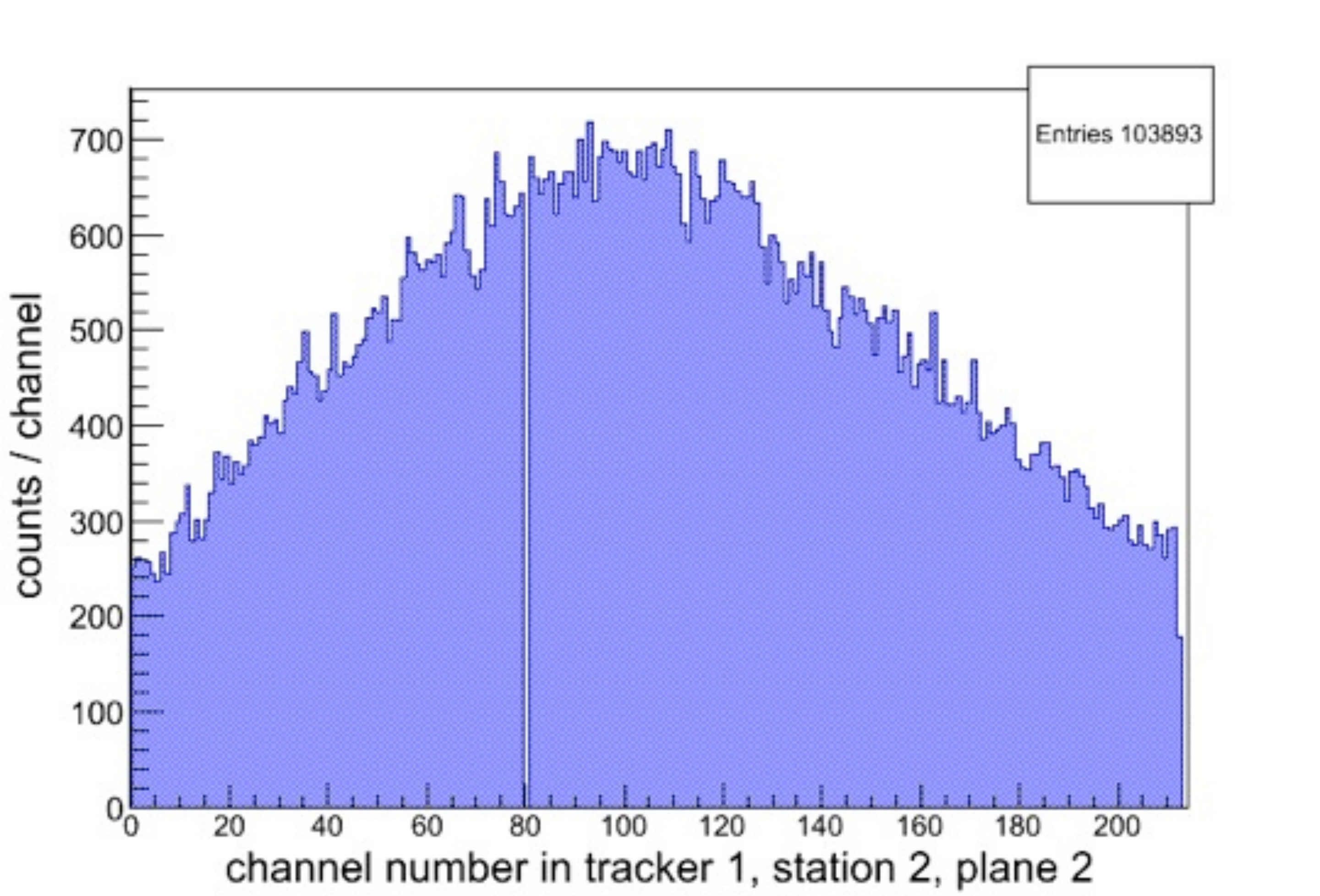
Investigate light production for different momentum settings (proposed for Chicago Students).



Cosmic Run...

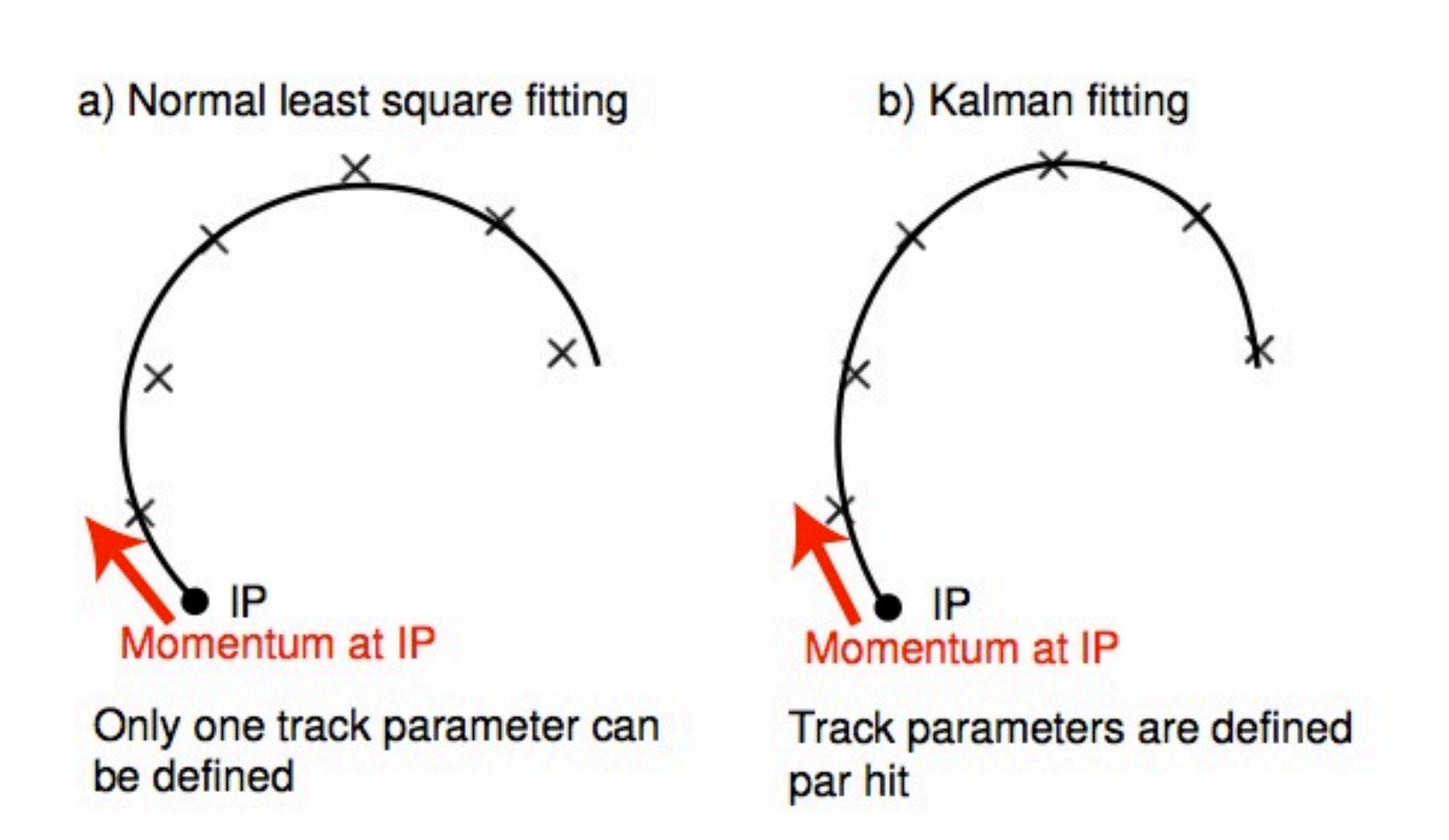
Wrap up MICE note with:

- reconstruction efficiencies,
- dead channels...



Kalman Fitting

Why?


a) Normal least square fitting

b) Kalman fitting

Only one track parameter can be defined

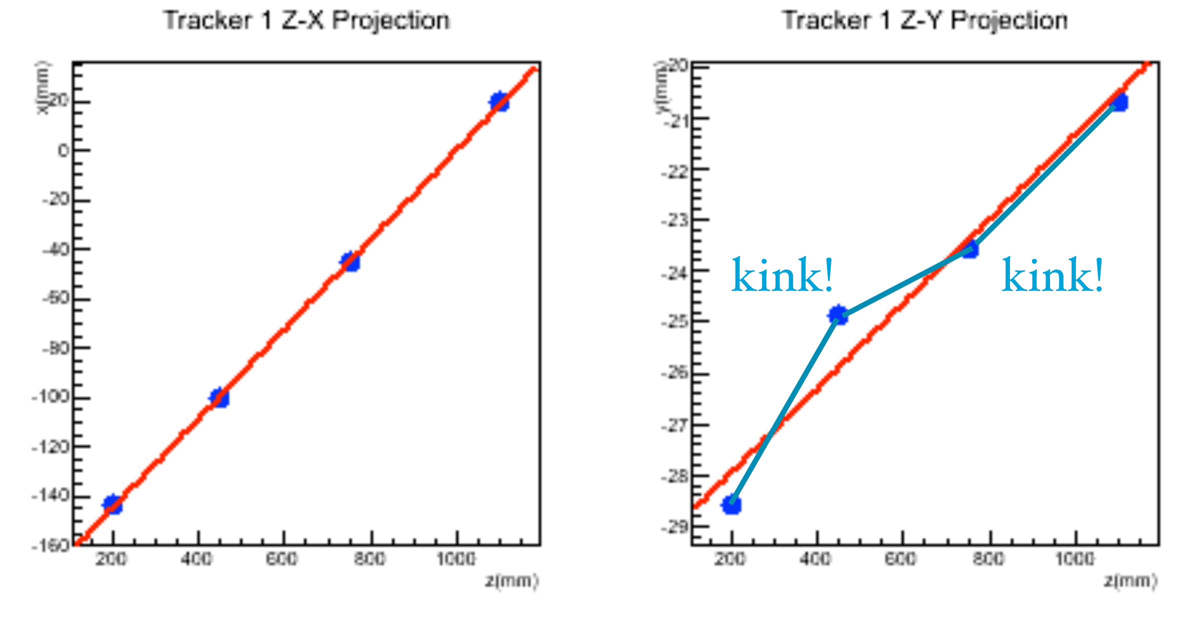
Track parameters are defined par hit

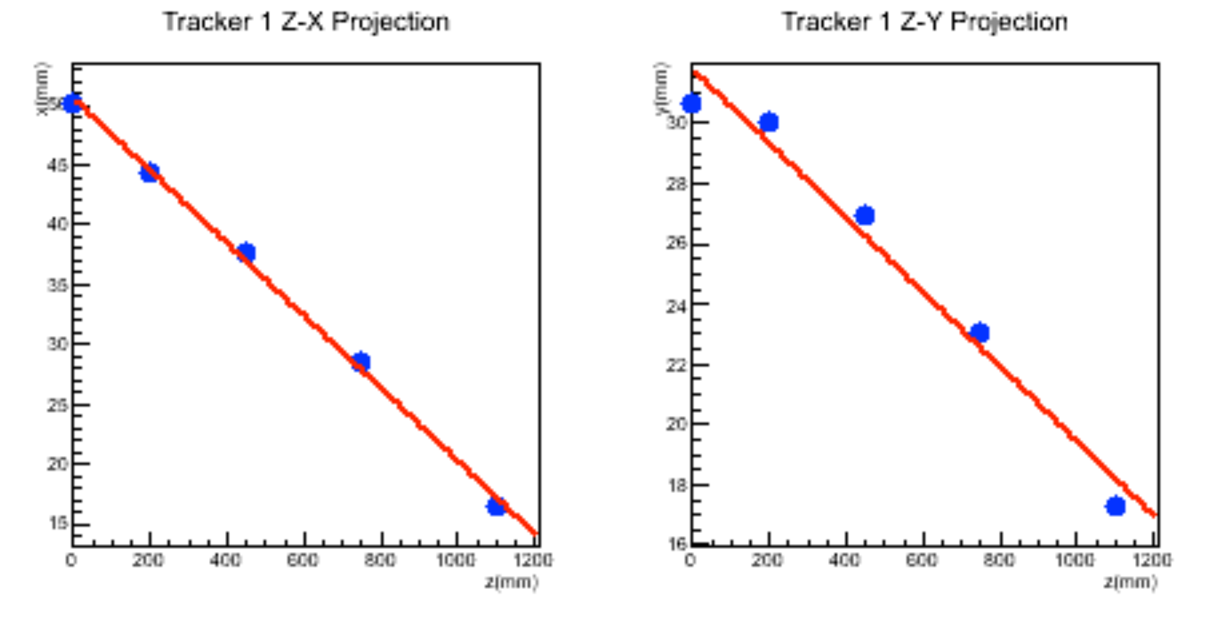
Figure 1.1: Simple χ^2 fitting versus Kalman filter based fitting.

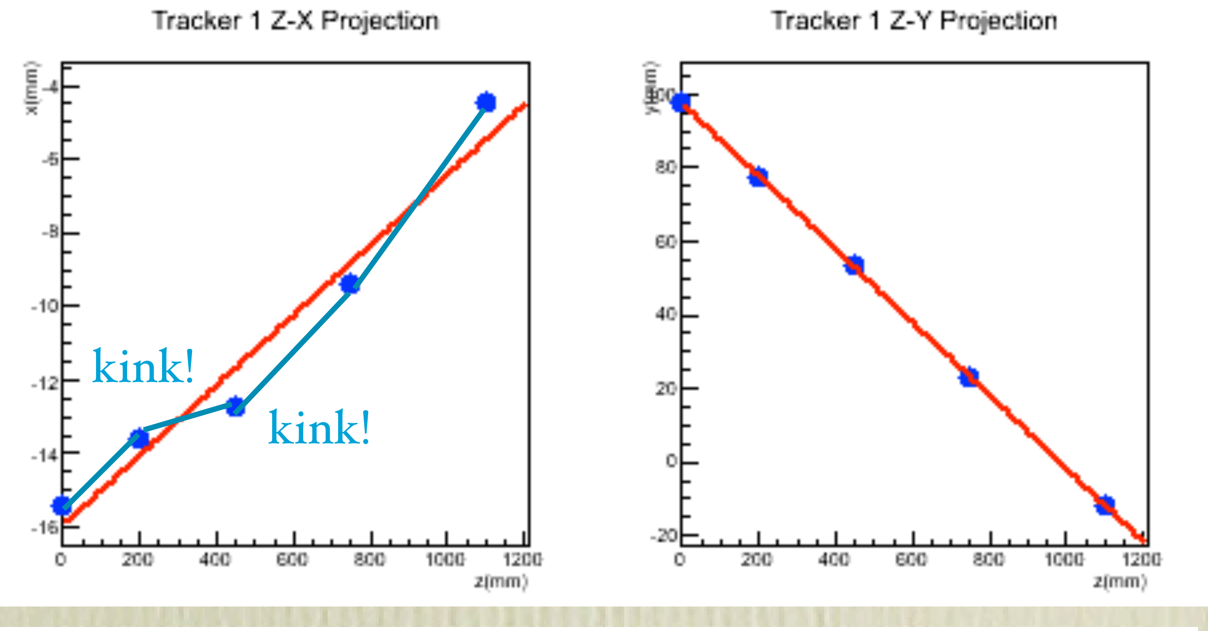
Optimal estimate of the state-vector (x, y, x', y', p) at each site.

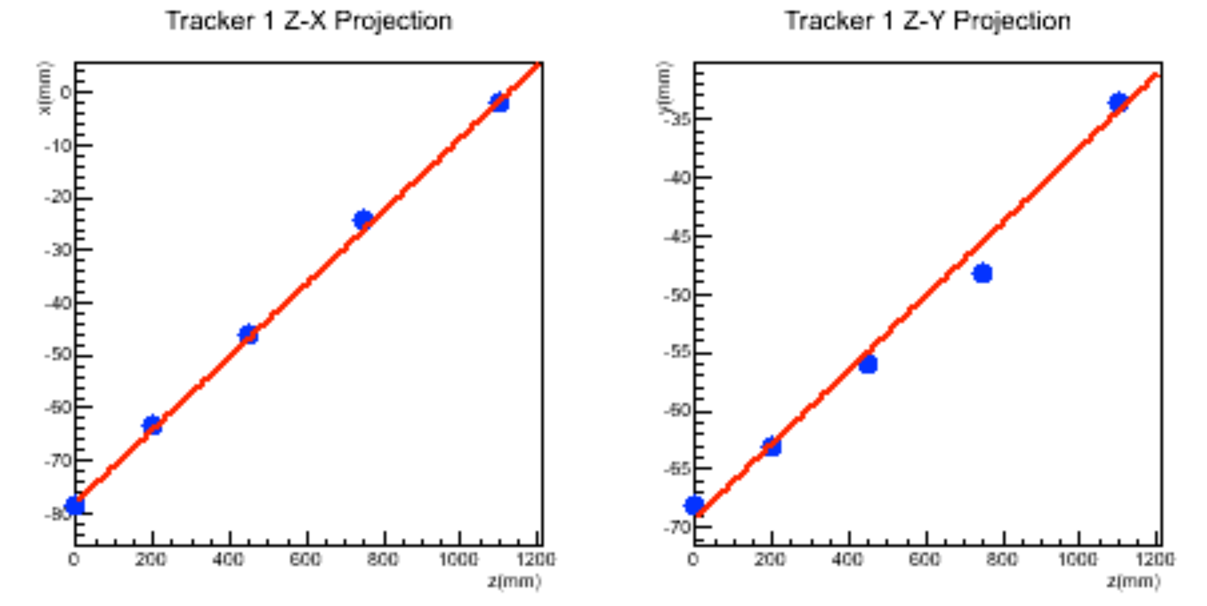
from "Extended Kalman Filter", by Keisuke Fujii

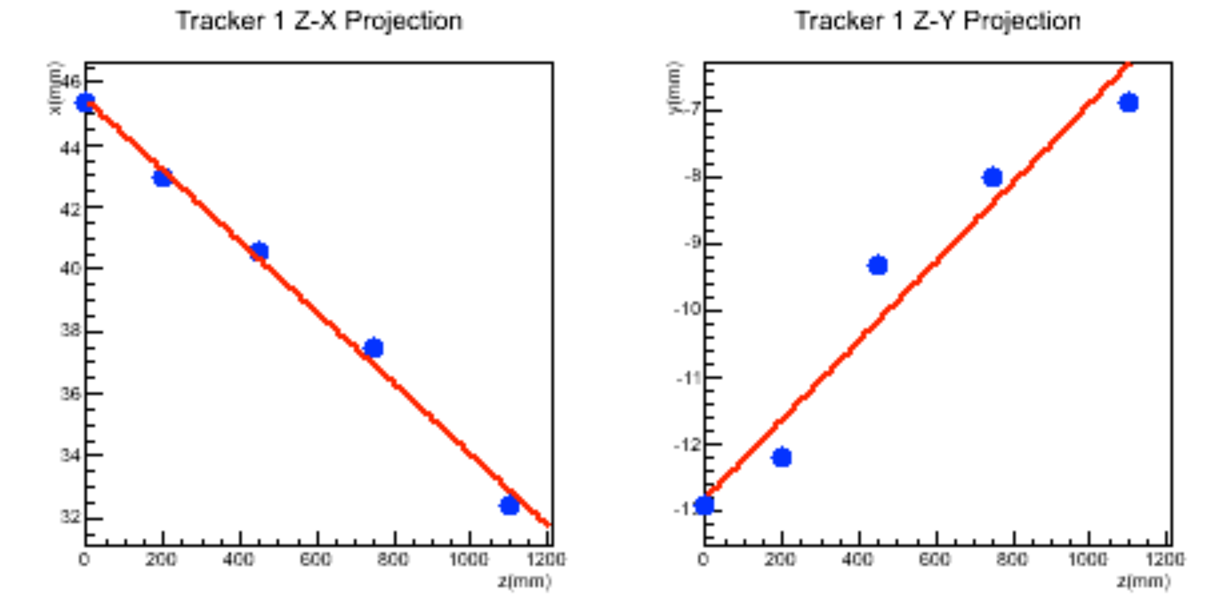
The tracker example at the Pattern Recognition stage (Adam Dobbs)







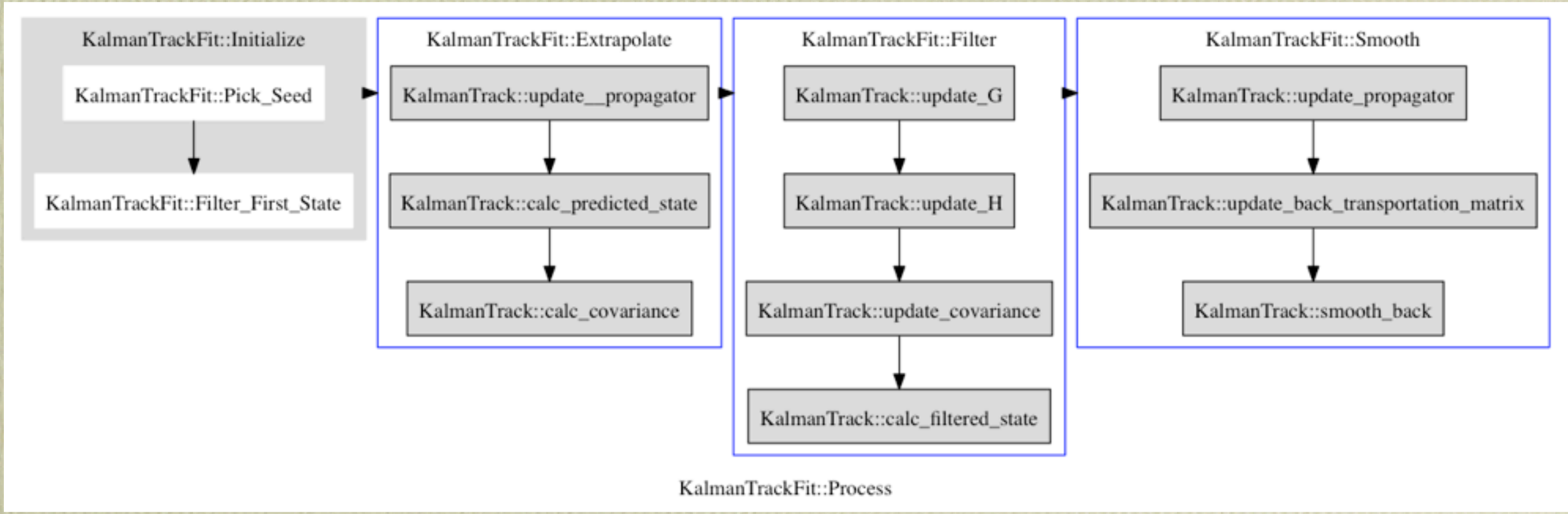




Track Fit with Kalman

- In G4MICE RecPac was used for the track fit
  - RecPac proved to be hard to understand/deal with;
  - inclusion of MCS didn't work;
  - it was slow!
- Tracker Group now owns it's own Kalman Filter routine:
  - includes MCS (Energy Loss to be added too);
  - doesn't slow down the reconstruction in a noticeable way;
  - ownership = comprehension, control.

MAUS meets Kalman



KalmanTrackFit::Initialize

KalmanTrackFit::Pick_Seed

KalmanTrackFit::Filter_First_State

KalmanTrackFit::Extrapolate

KalmanTrack::update__propagator

KalmanTrack::calc_predicted_state

KalmanTrack::calc_covariance

KalmanTrackFit::Filter

KalmanTrack::update_G

KalmanTrack::update_H

KalmanTrack::update_covariance

KalmanTrack::calc_filtered_state

KalmanTrackFit::Smooth

KalmanTrack::update_propagator

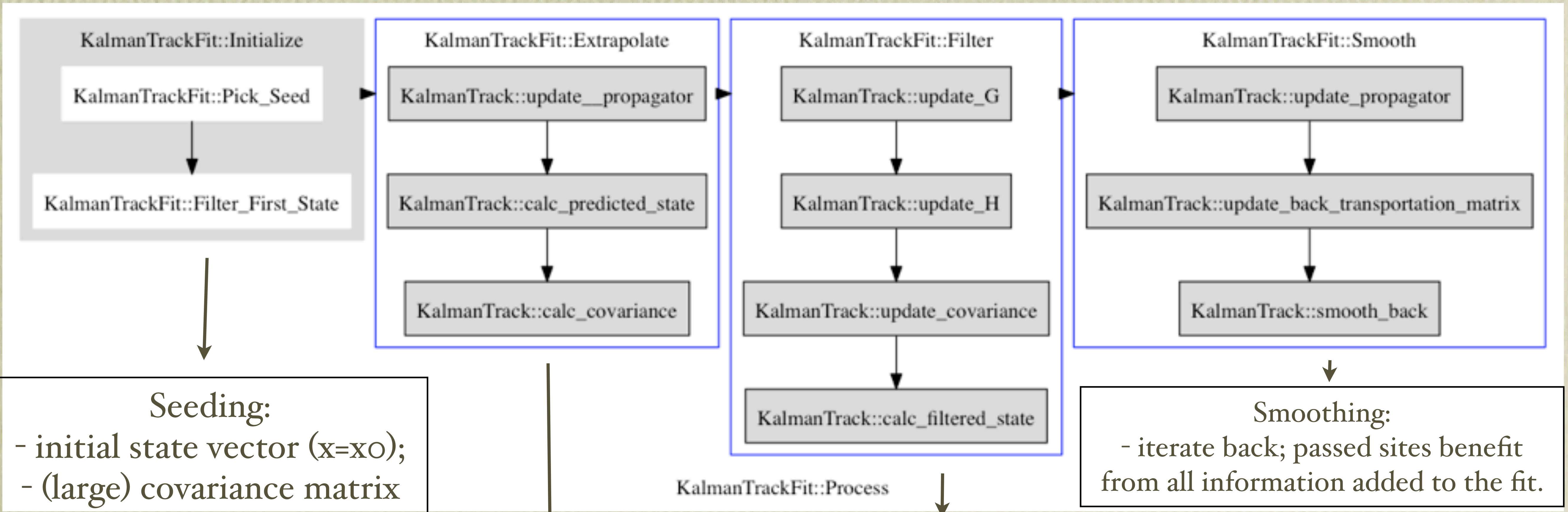
KalmanTrack::update_back_transportation_matrix

KalmanTrack::smooth_back

KalmanTrackFit::Process

3 main Kalman Classes:
KalmanTrackFit - main worker, controls workflow;
KalmanTrack - abstract class, from which StraightTrack, HelicalTrack, ... are derived.
KalmanSite - container class for site matrices (state vectors, covariance matrices, measurements...)

MAUS meets Kalman



KalmanTrackFit::Initialize

KalmanTrackFit::Pick_Seed → KalmanTrackFit::Filter_First_State

KalmanTrackFit::Extrapolate

KalmanTrack::update__propagator → KalmanTrack::calc_predicted_state → KalmanTrack::calc_covariance

KalmanTrackFit::Filter

KalmanTrack::update_G → KalmanTrack::update_H → KalmanTrack::update_covariance → KalmanTrack::calc_filtered_state

KalmanTrackFit::Smooth

KalmanTrack::update_propagator → KalmanTrack::update_back_transportation_matrix → KalmanTrack::smooth_back

KalmanTrackFit::Process

Seeding:
- initial state vector ($x=x_0$);
- (large) covariance matrix

Smoothing:
- iterate back; passed sites benefit from all information added to the fit.

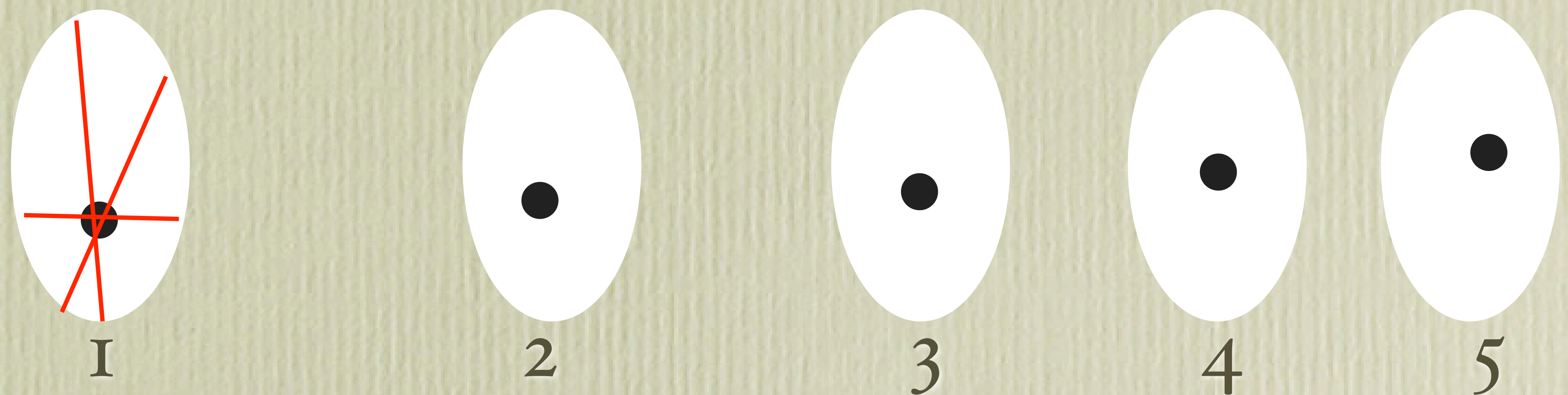
Extrapolation:
$x_1=x_0+mz$
- use deterministic equations of motion to predict state vector at the next site;
- mcs is accounted for when updating cov. matrix.

Filtering:
$x'_1 = x_1 + K*\mathrm{pull}$
- include measurement at the site.
- Kalman Gain defined by covariance matrix of projected state vector and measurement error.

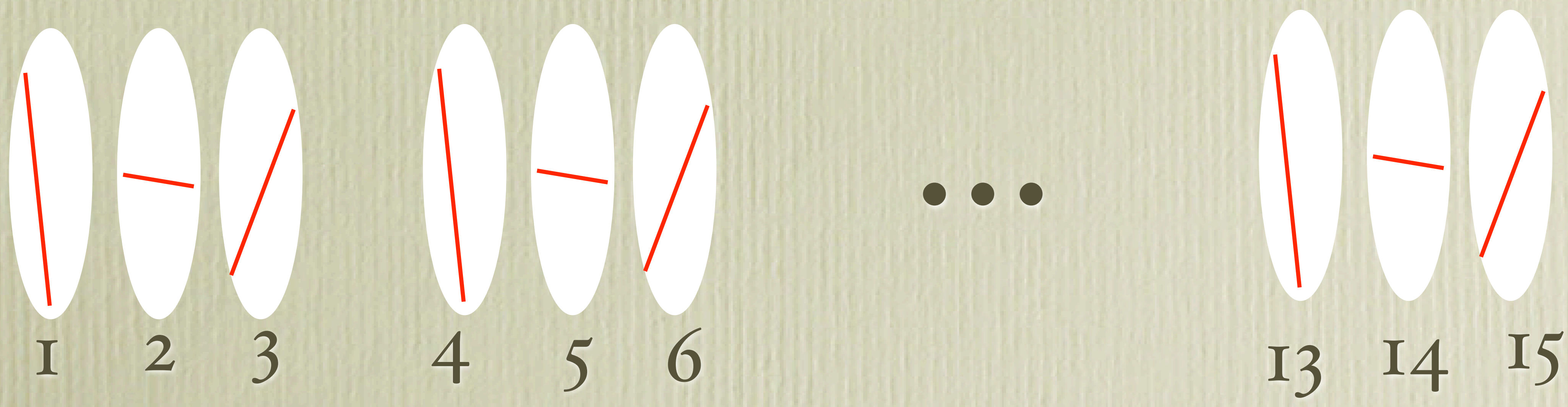
Measurement Sites

- The fitting routine doesn’t use the reconstructed spacepoints (x, y, z)...
- ... it uses the unbiased scintillating plane measurements.

Spacepoint Fit Scheme

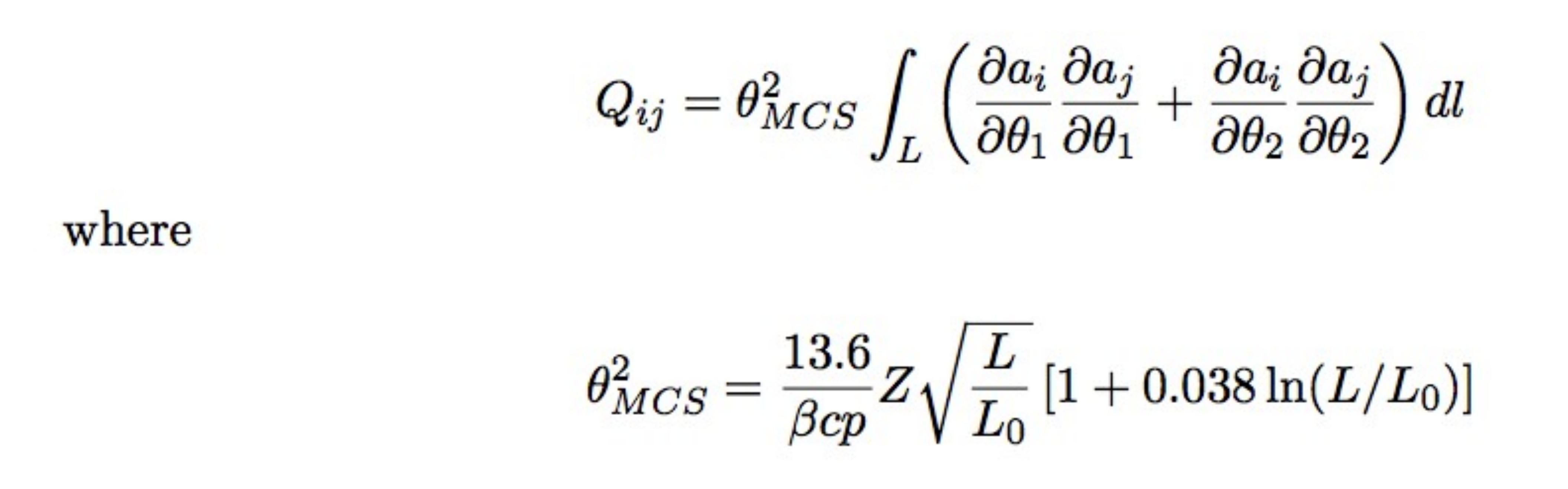


Measurement Fit Scheme



Including MCS in the Fit

- The scattering happens in the two planes perpendicular to the direction of motion. They are normally distributed around a mean value of 0.
- No thin layer approximation. Matrix elements defined by:



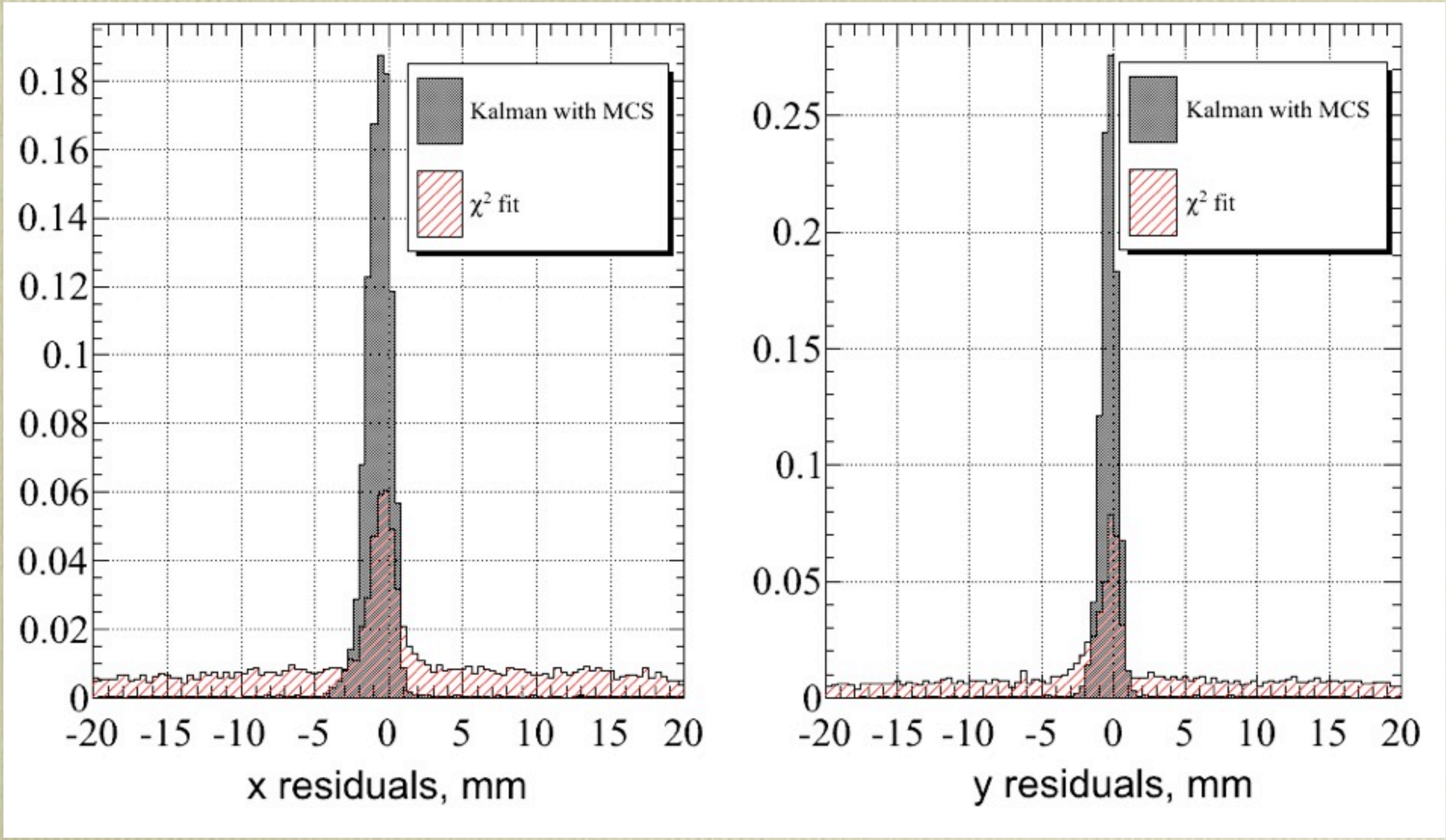
$$Q_{ij} = \theta^2_{MCS} \int_L \left(\frac{\partial a_i}{\partial \theta_1} \frac{\partial a_j}{\partial \theta_1} + \frac{\partial a_i}{\partial \theta_2} \frac{\partial a_j}{\partial \theta_2} \right) dl$$

where

$$\theta^2_{MCS} = \frac{13.6}{\beta c p} Z \sqrt{\frac{L}{L_0}} \left[1 + 0.038 \ln(L/L_0)\right]$$

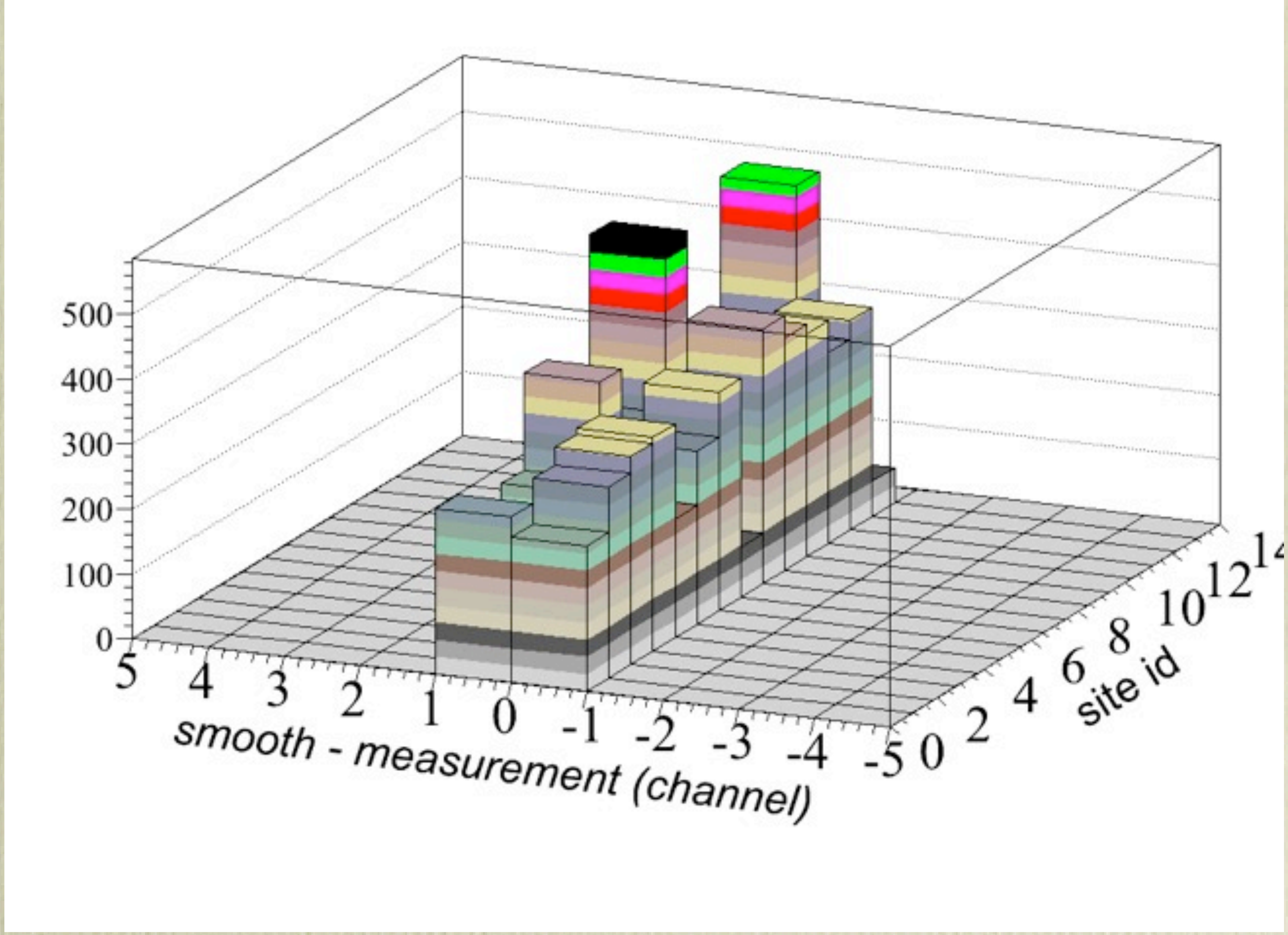
extracted from the code documentation.

Plot below: MC truth - reconstructed



Cosmic data

(remember: a full tracker has 15 sites)



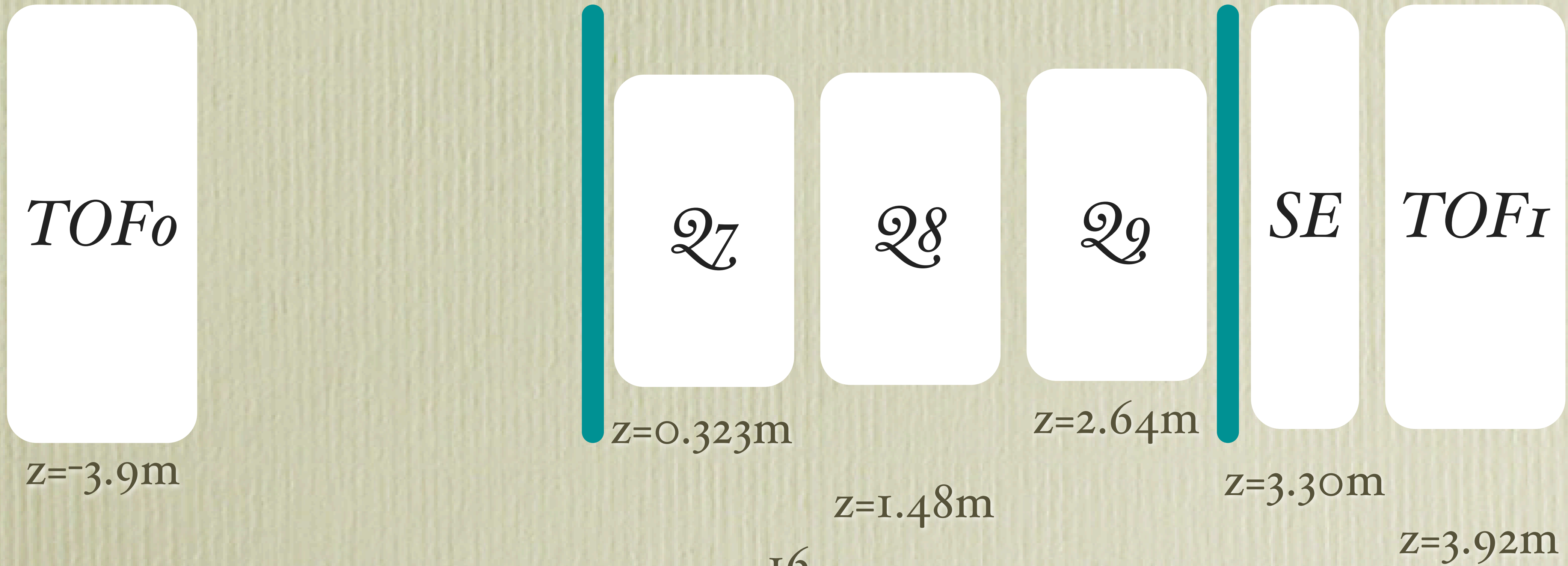
- the smoothed values agree with the measurement.
  - the kinks in the trajectory don’t ruin the fit.

Flexibility

- Can pick from different modes:
  - straight tracks;
  - helical tracks;
  - propagation in quadrupole fields;
- measurement sites can be defined to be scintillating fibre hits or slab hits.

TOF + Single Station Fit

- There is a plan and the tools for it
  - Jaroslaw Pasternak (Imperial) is on board here!



To do:

- Wrap-up Tracker Efficiency Studies in a MICE Note;
- Light Yield Studies (Chicago students);
- Kalman work:
  - add energy loss;
  - see what momentum reconstruction looks like (need helical tracks);
  - add testing and finish documentation;
  - check statistical properties.
- May Run:
  - try propagation of particles in the quadrupoles.