University of Bologna & INFN

Kalman Filter 4 FOOT first results

Matteo Franchini

16 February 2017

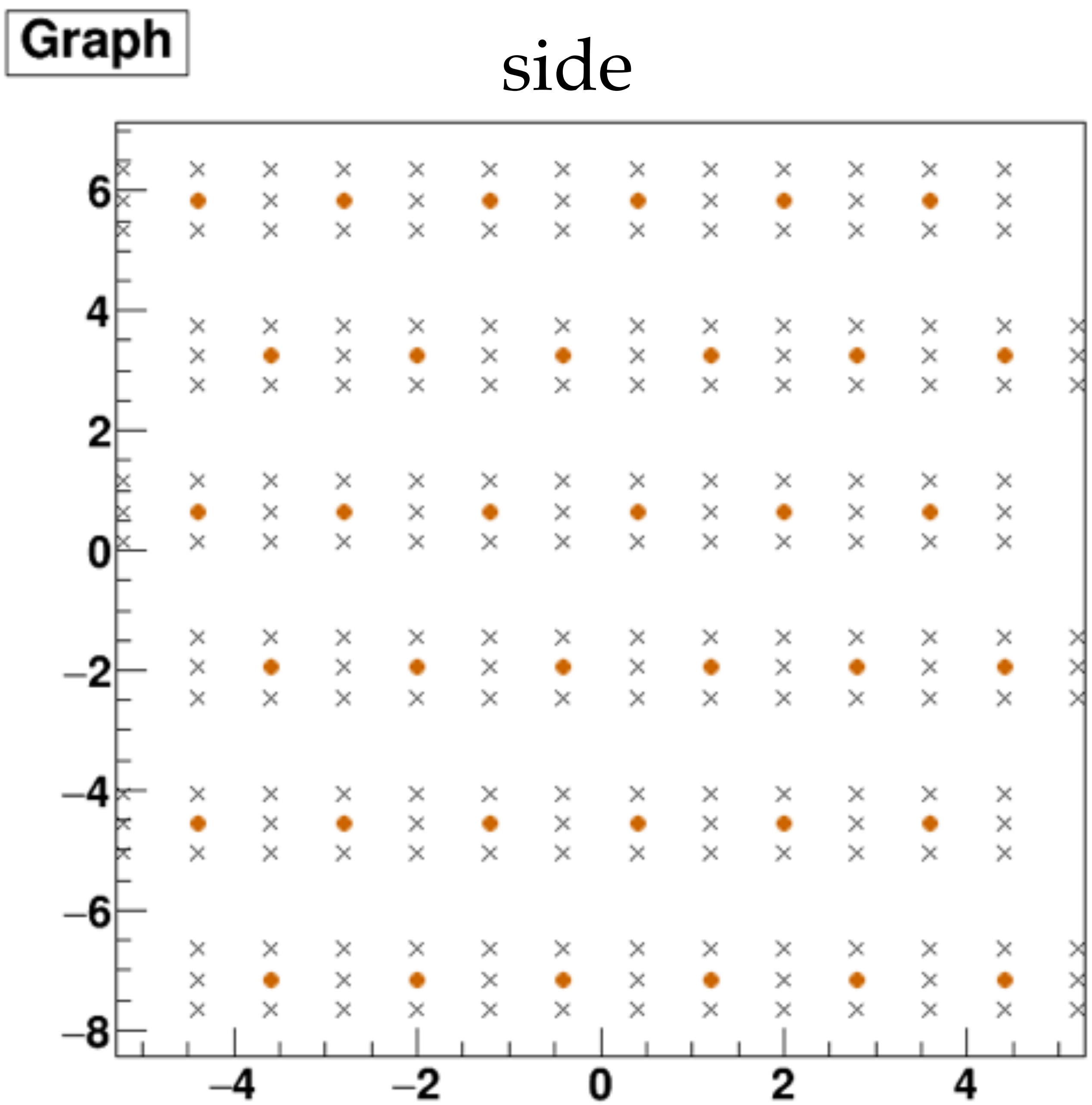
matteo.franchini@cern.ch

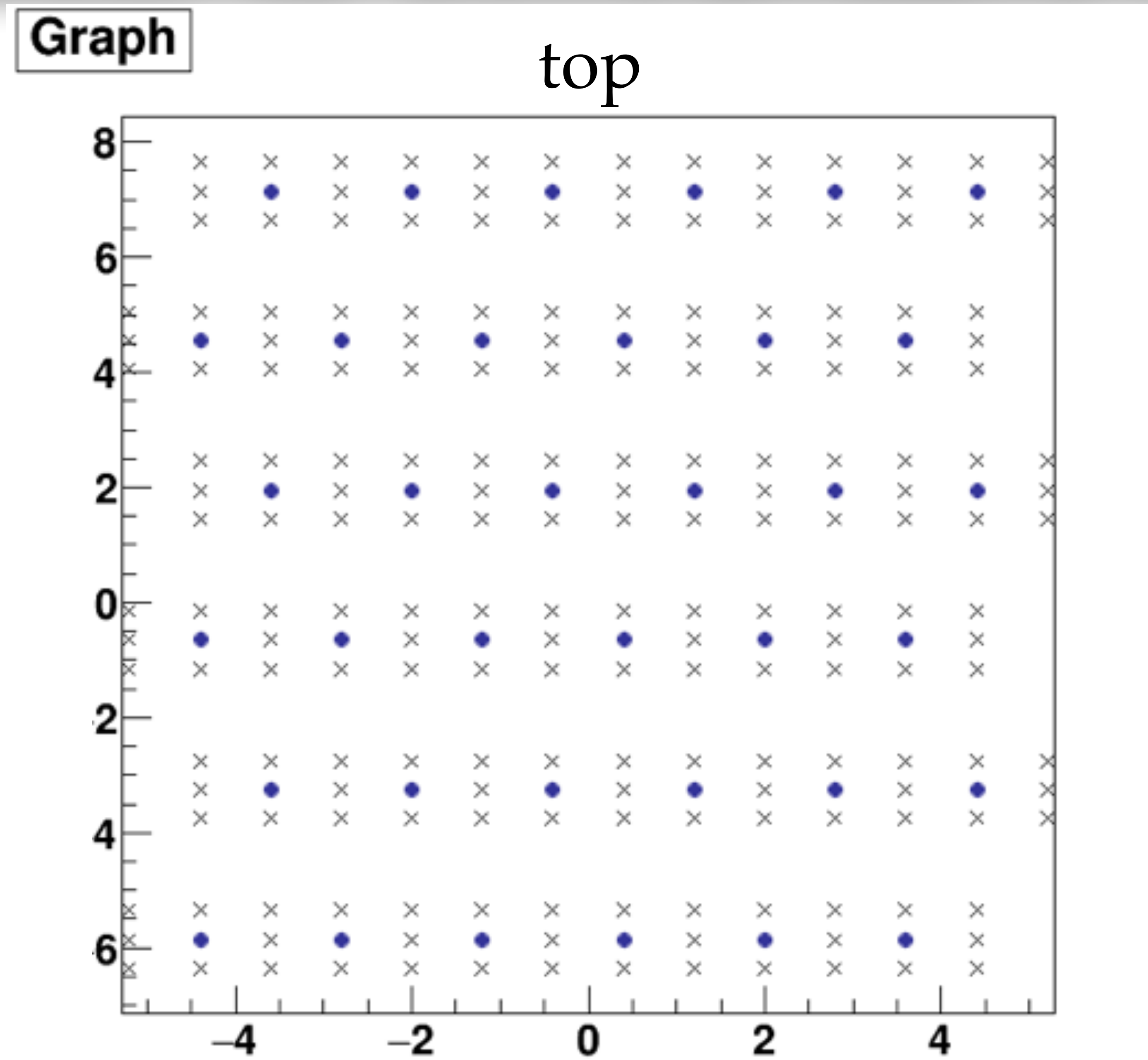
Intro

- Use Kalman Filter method to combine hits form different detectors into a precise track reconstruction (momentum!).
- Goal: momentum resolution < 4%!
- To be used as a test of the configuration of FOOT (geo + performance).
- Main problem: harmonise multiple sub-detector inputs.

Technical step until now

- Using the Reconstruction framework, including GenFit (thank Alessio).
- Starting from Drift Chamber package:
  - geometry implementation from the one in FLUKA's input files.
  - mirroring Beam Monitor (thanks Yun).
  - code restyling (FIRST->FOOT, improve in/out, stability, class for wires, …)
  - reading event info from ntuple: tests+fixes, add truth matching to hits, fix local/global coordinates transformations(managed by sub detector itself by now).
- C++ class to interface with GenFit and make the Kalman (core code!) + store tracks.
  - takes hits from different sub-detectors, transmutes in GenFit-like inputs and interpret the fit results.
  - Use global material and geometry (from each sub-detector) [to be finalised]
- Implementation of the Magnetic Field [last minute!].

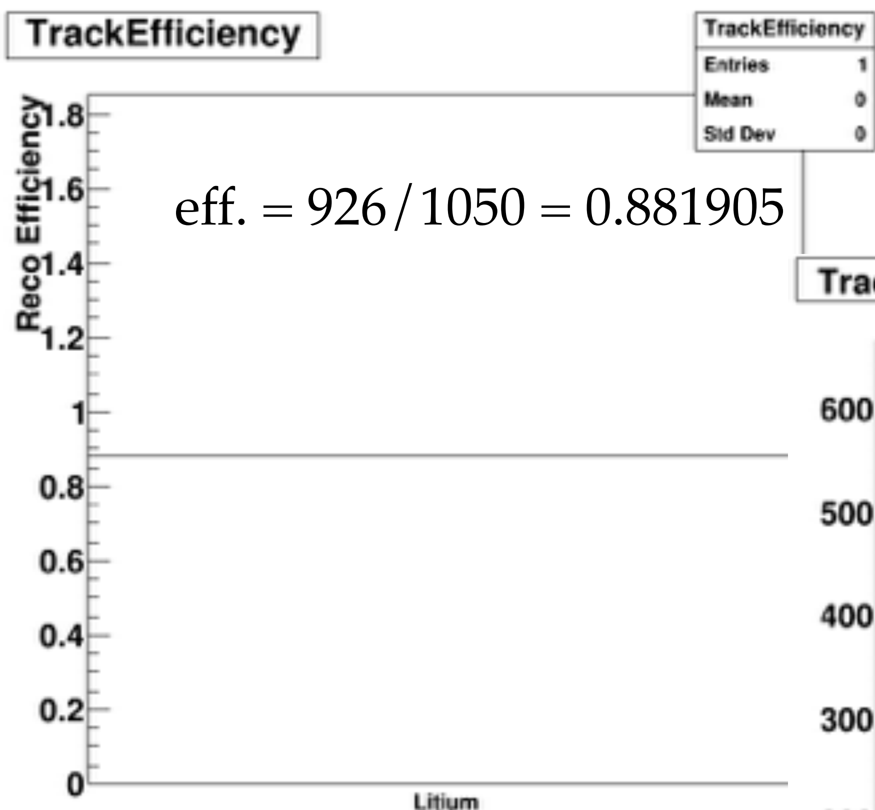

Graph
side

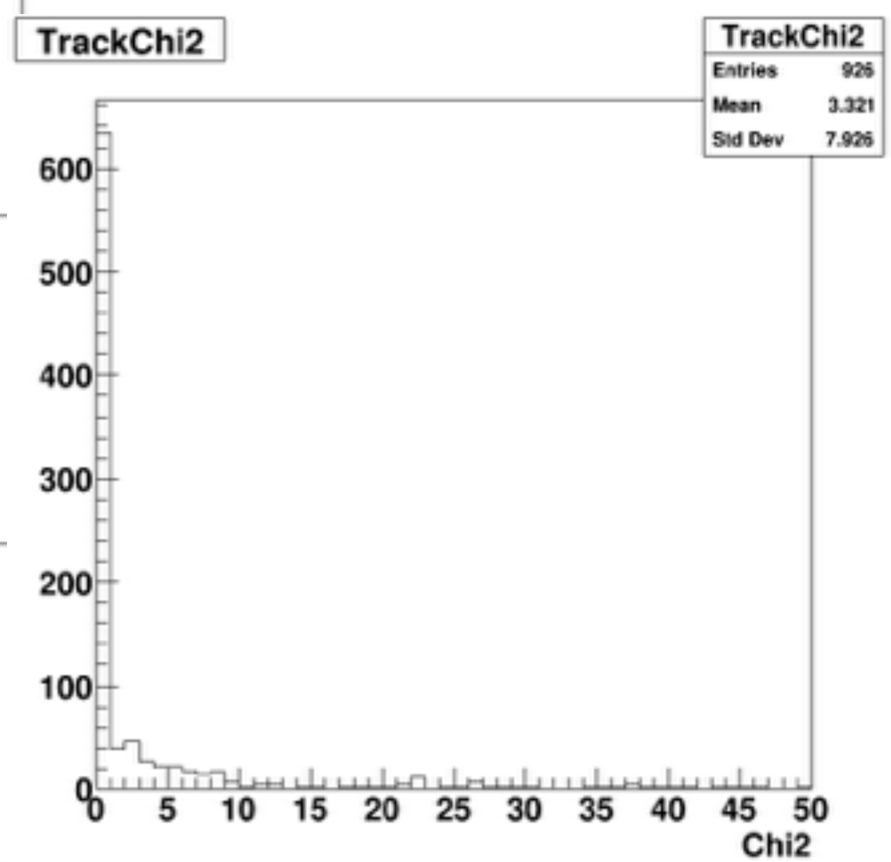

Graph
top

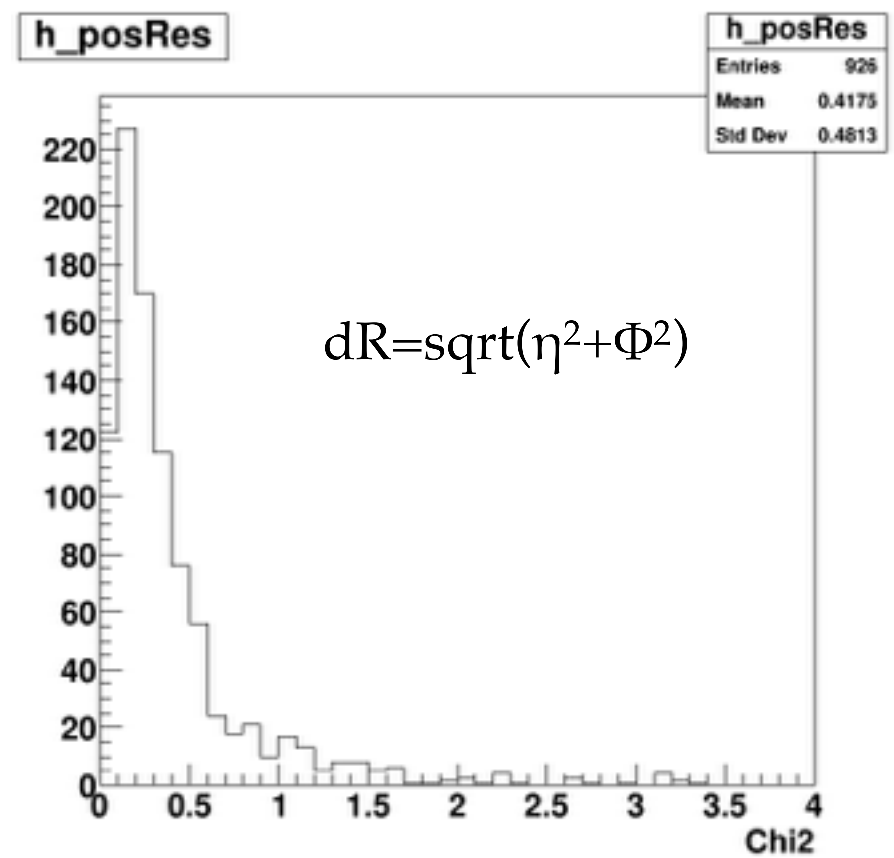
Preliminary results 1

B field OFF

- **Input file:** 12C_C2H4_noMag_highThres.root;
- **Lithium** hits ONLY;
- **Drift Chamber** ONLY;
- no-DC materials still missing (air bubble).
- **Seed:** (0,0,30)cm, pz=2.4GeV, proton.
- **tot nEvents:** 17511

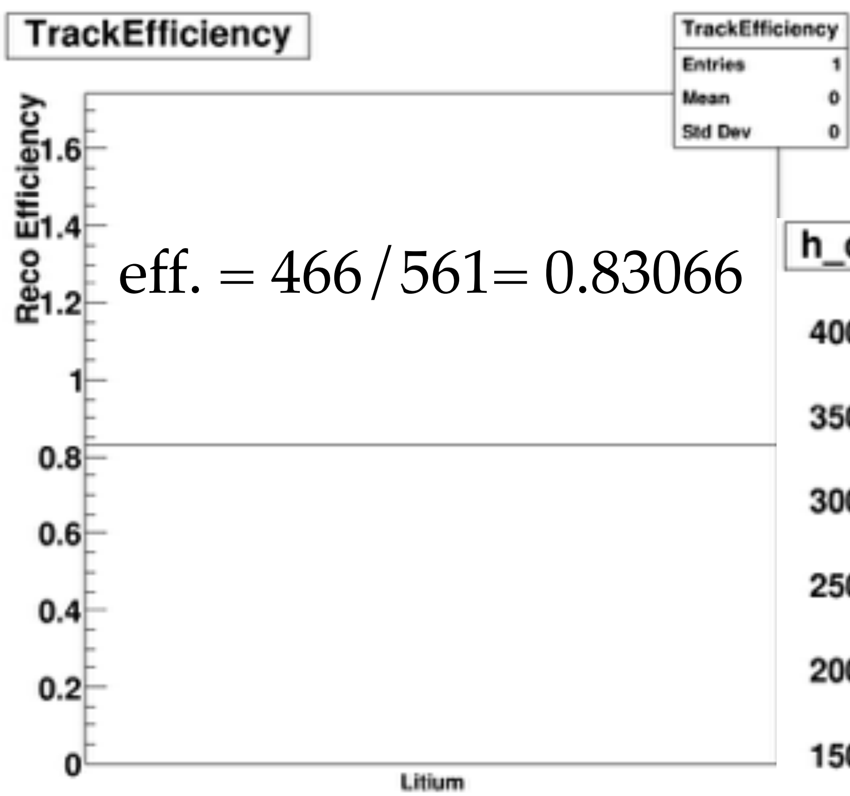


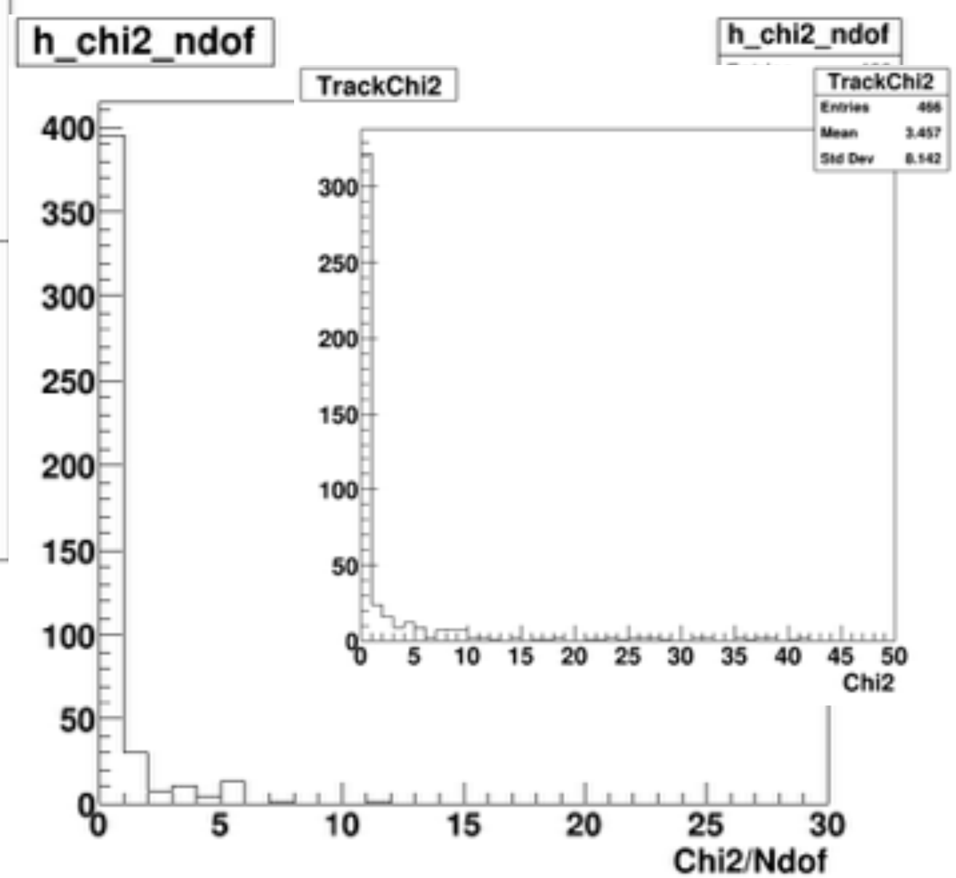


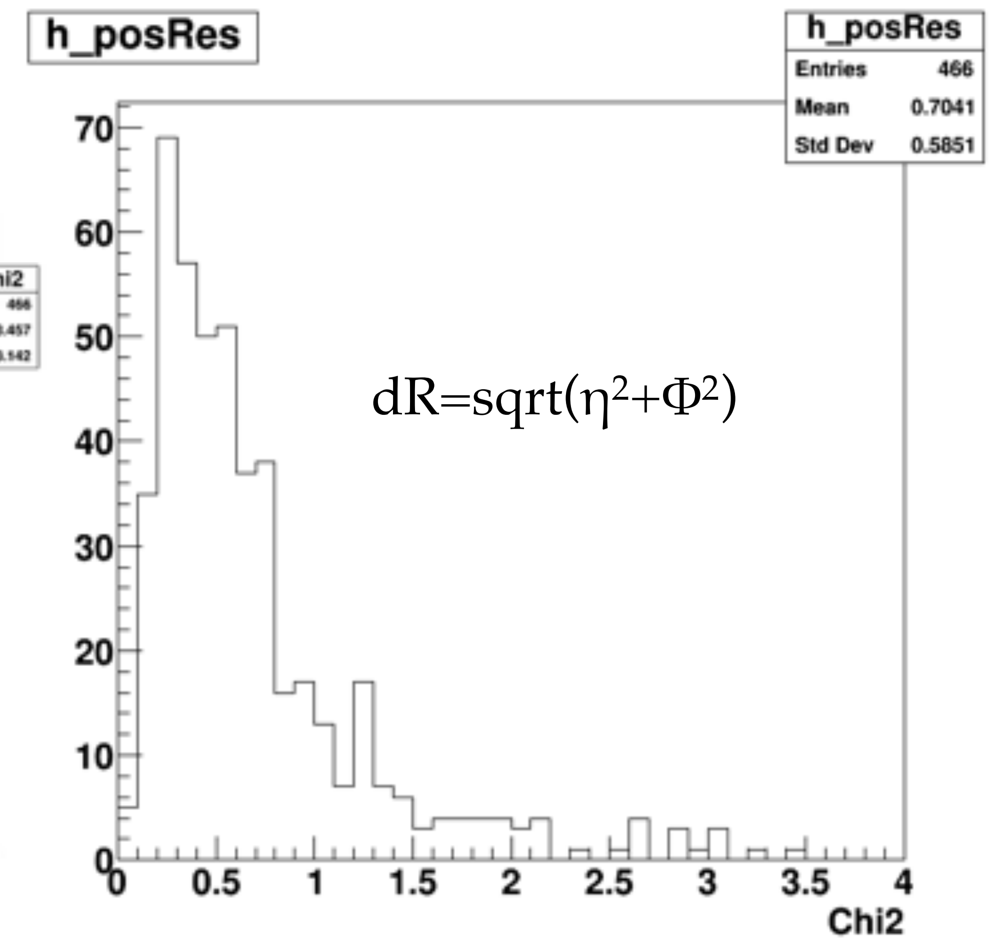


B field ON

- **Input file:** 12C_C2H4_Mag_highThres.root;
- **Lithium** hits <u>ONLY</u>;
- **Drift Chamber** <u>ONLY</u>;
- no-DC materials still missing (air bubble).
- **Seed:** (0,0,30)cm, pz=2.4GeV, proton.
- **tot nEvents:** 9715







Next Steps

- **CHECKS!!!** and tuning of the fitter (seed, mass, measurement handling, …)
- Add pixel detector, Vertex + Internal Tracker. [ongoing]
- Implement all the material, at lest between target and DC. [ongoing]
- Hit selection strategy before the KF. [dreamworld stage]
- Testing different Foot configurations:
  - vacuum "inside" magnets;
  - no Internal Tracker;

Back-up

Kalman Filter

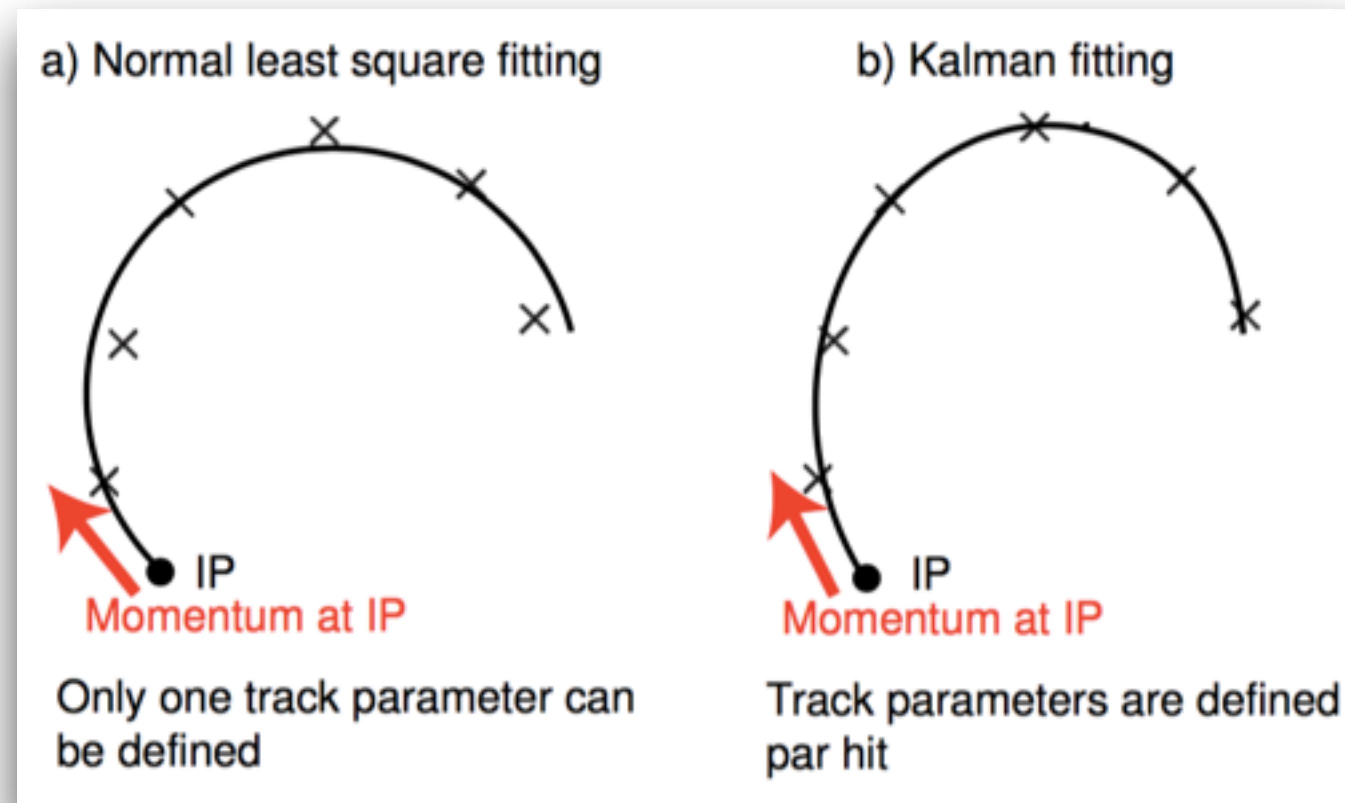
- R.E.Kalman proposed an iterative method to estimate the states of a dynamic system starting from a series of measurement points on N surfaces.

- Initially used to calculate the trajectory of ballistic missiles. Later introduced in particle physics (1984).

- Precise as a global χ^2 fitting;

- ***Fast***;

- Best ***track parameter*** found ***for each hit!!!***



"A new approach to linear filtering and prediction problems" Trans. ASME J. Basic Eng. 82 (1960), 35

Kalman in pills

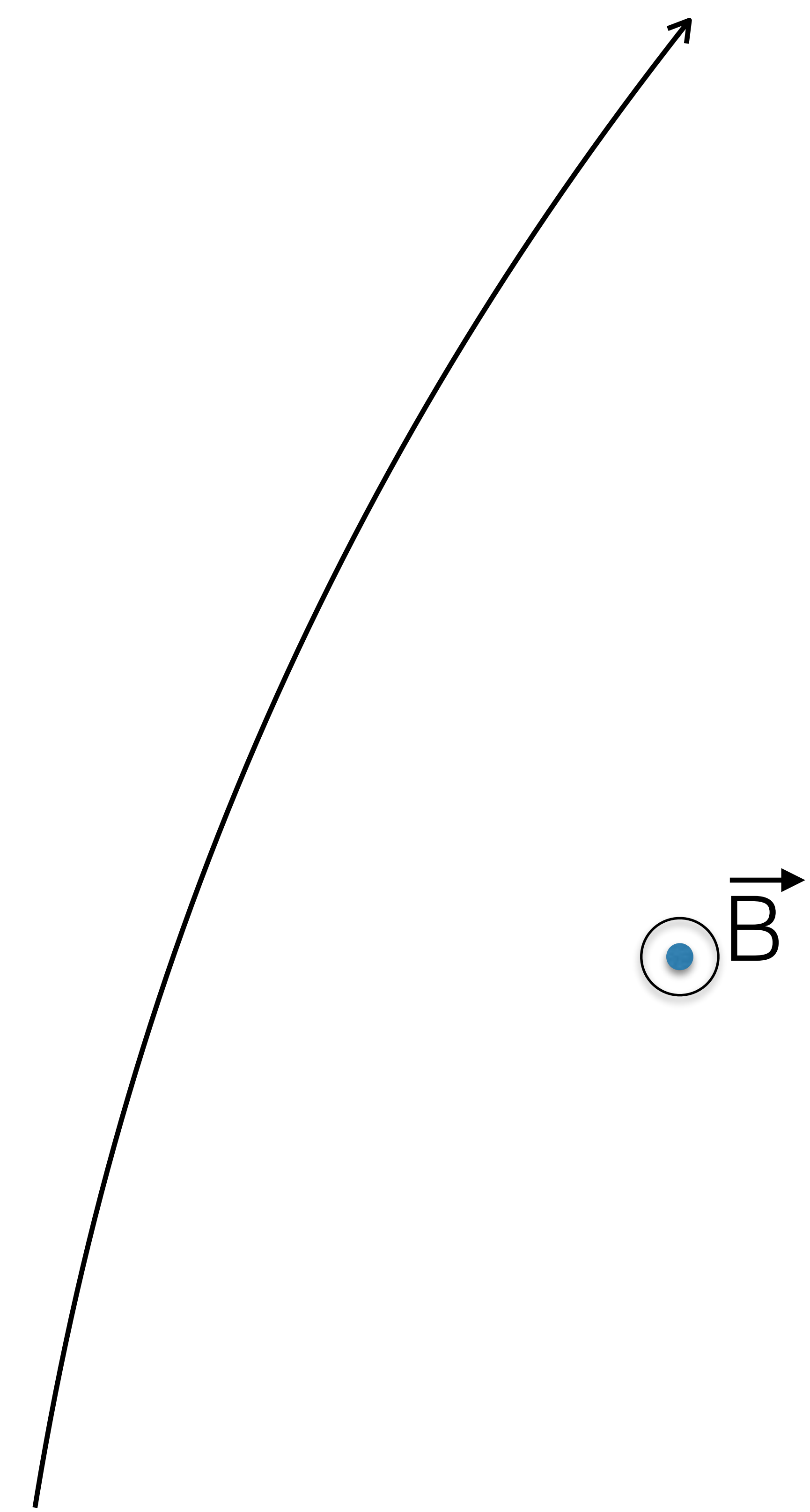
1. Take an **ideal particle** in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.

2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).

3. Propagate the first hit to the next layer. Propagator Matrix F.

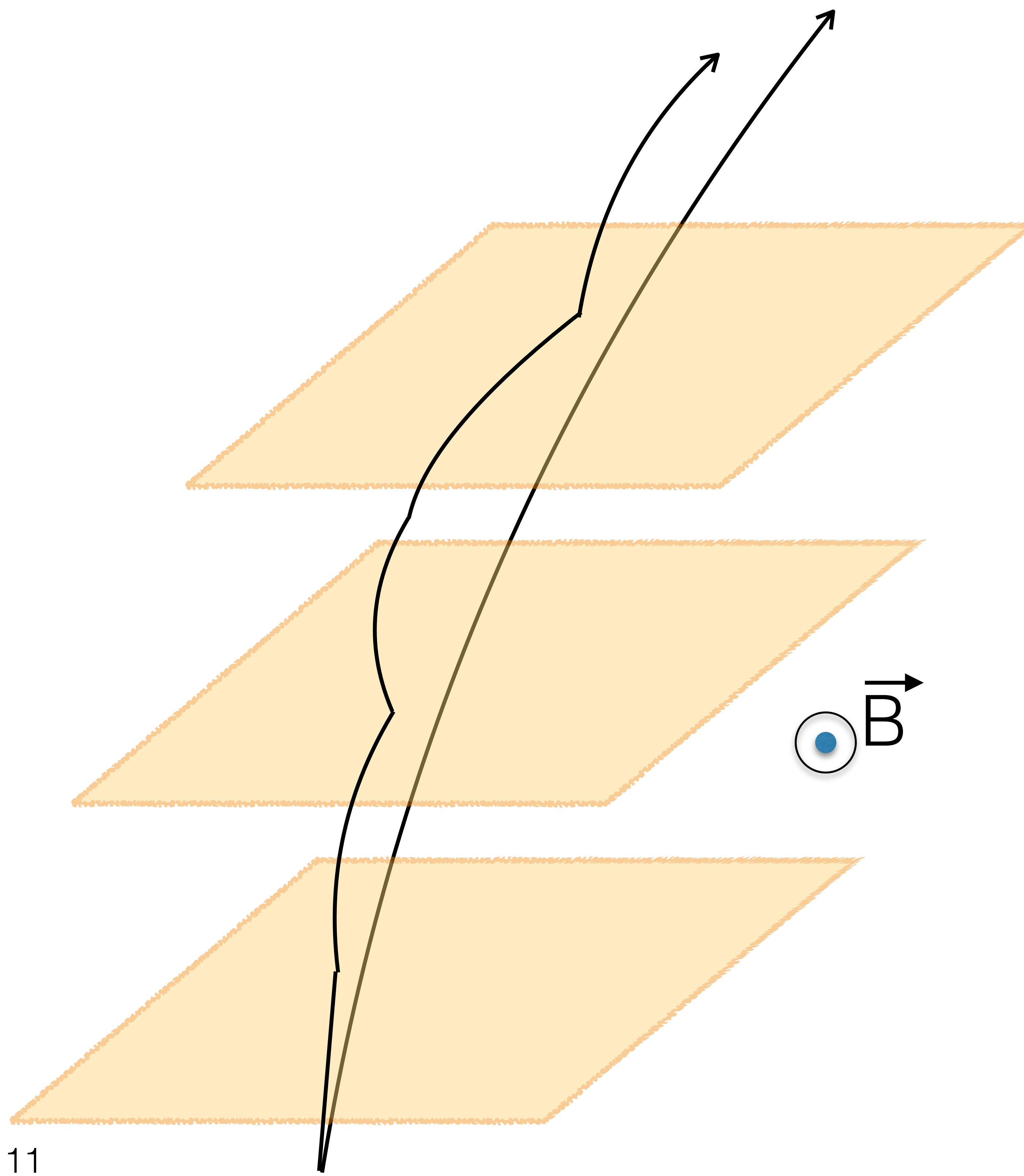
4. Find the best compromise between the propagated point and the closest hit on the 2nd layer. Use a Chi2 and a Projection Matrix H.

5. Iterate 3 and 4 for the next leayers.



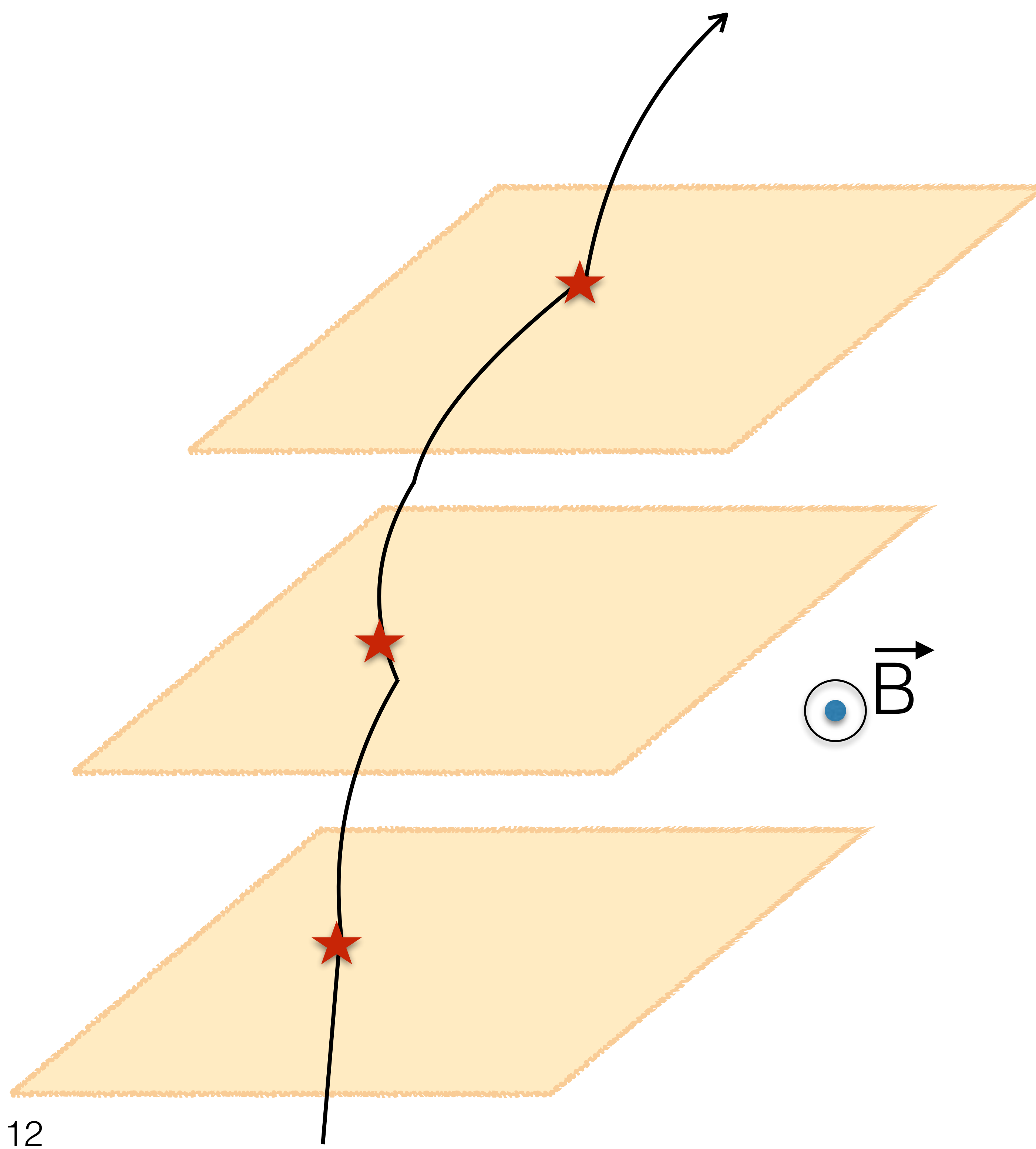
Kalman in pills

1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to **M.S.** and **energy loss**.
2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).
3. Propagate the first hit to the next layer. Propagator Matrix F.
4. Find the best compromise between the propagated point and the closest hit on the 2nd layer. Use a Chi2 and a Projection Matrix H.
5. Iterate 3 and 4 for the next leayers.



Kalman in pills

1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.
2. We'll see some **measurement hits** on the detector layers (considering finite detector uncertainty).
3. Propagate the first hit to the next layer. Propagator Matrix F.
4. Find the best compromise between the propagated point and the closest hit on the 2^{nd} layer. Use a Chi2 and a Projection Matrix H.
5. Iterate 3 and 4 for the next leayers.



Kalman in pills

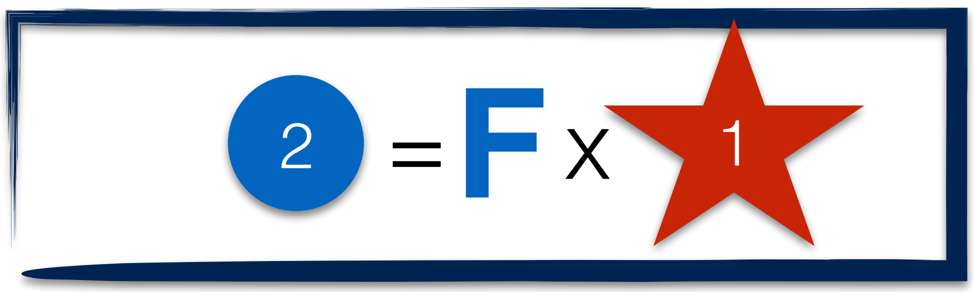
1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.

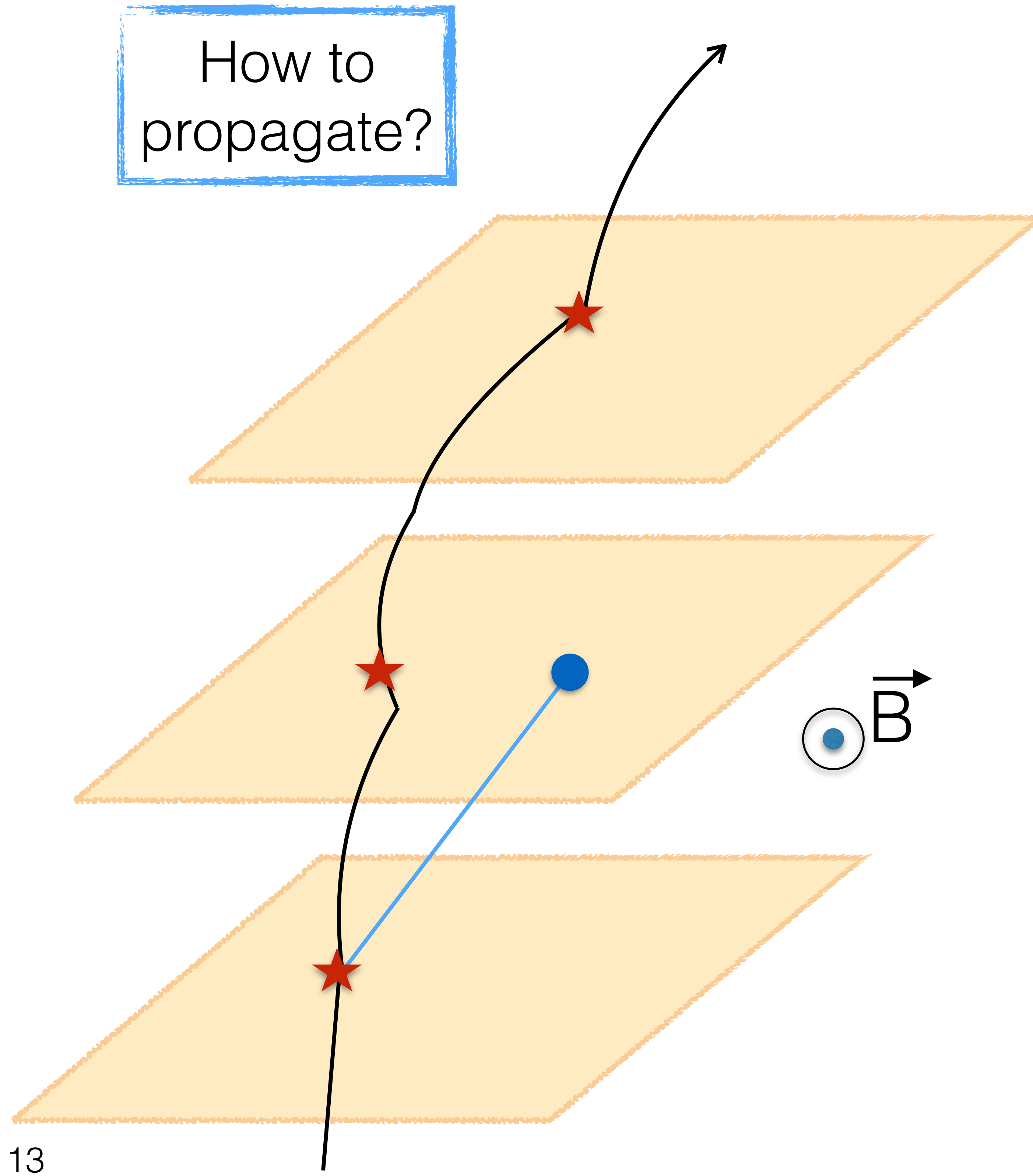
2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).

3. **Propagate** the first hit to the next layer. Propagator Matrix **F**.

4. Find the best compromise between the propagated point and the closest hit on the 2^{nd} layer. Use a Chi2 and a Projection Matrix H.

5. Iterate 3 and 4 for the next leayers.





Kalman in pills

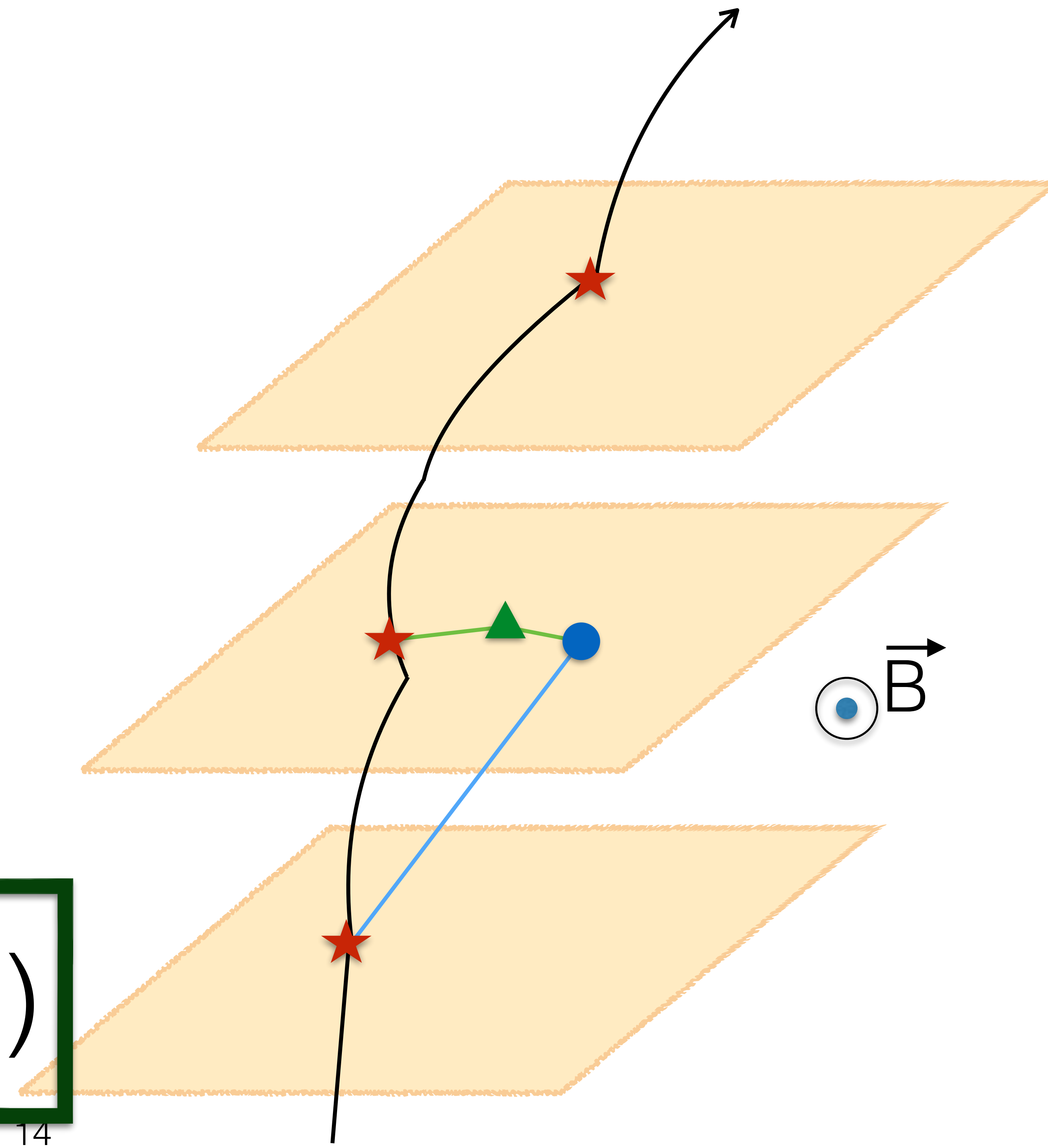
1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.

2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).

3. Propagate the first hit to the next layer. Propagator Matrix F.

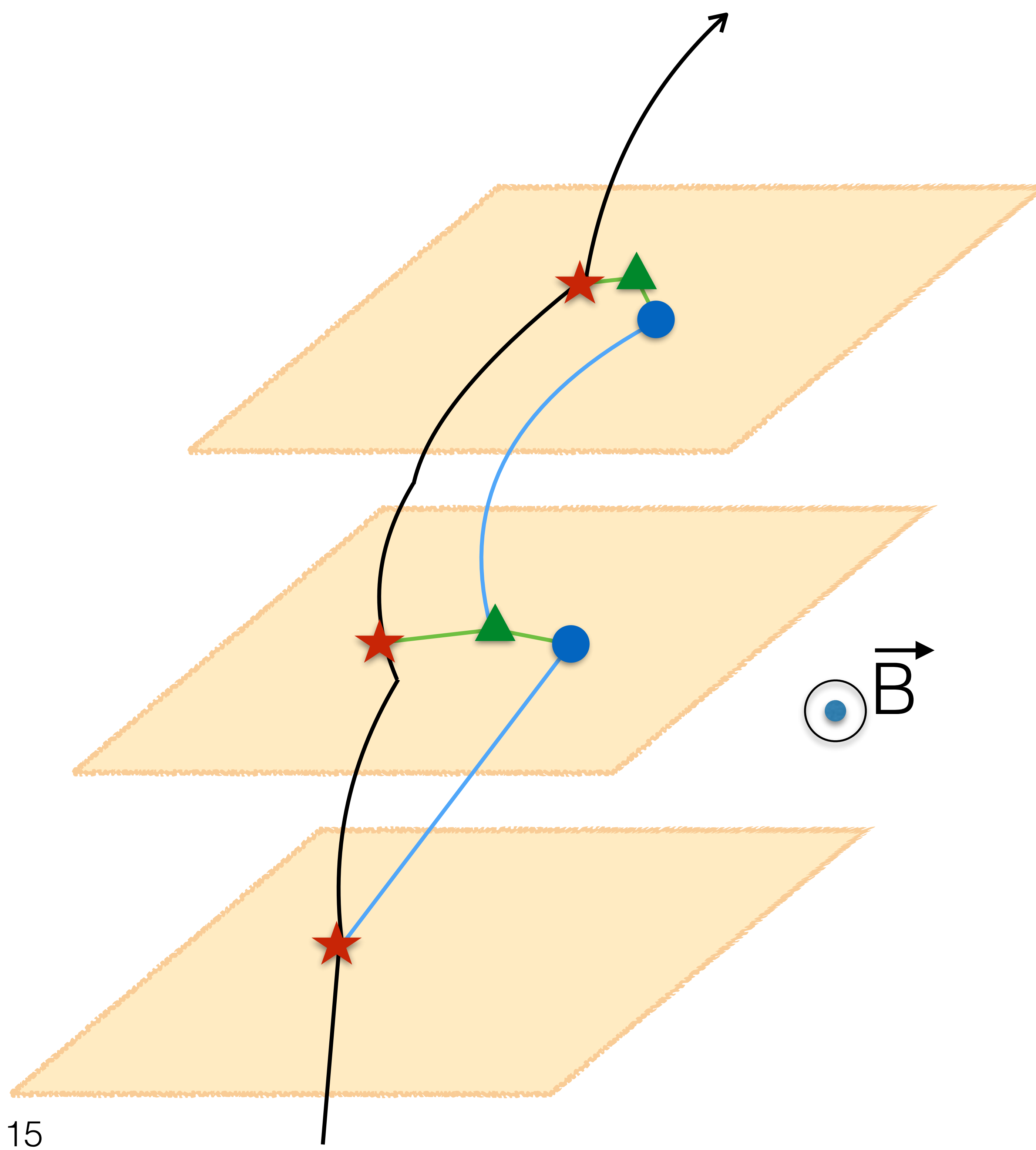
4. Find the best compromise between the propagated point and the closest hit on the 2nd layer. Use a **Chi2** and a Projection Matrix **H**.

5. Iterate 3 and 4 for the next leayers.



Kalman in pills

1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.
2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).
3. Propagate the first hit to the next layer. Propagator Matrix F.
4. Find the best compromise between the propagated point and the closest hit on the 2nd layer. Use a Chi2 and a Projection Matrix H.
5. **Iterate** 3 and 4 for the next leayers.



Kalman in pills

1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.

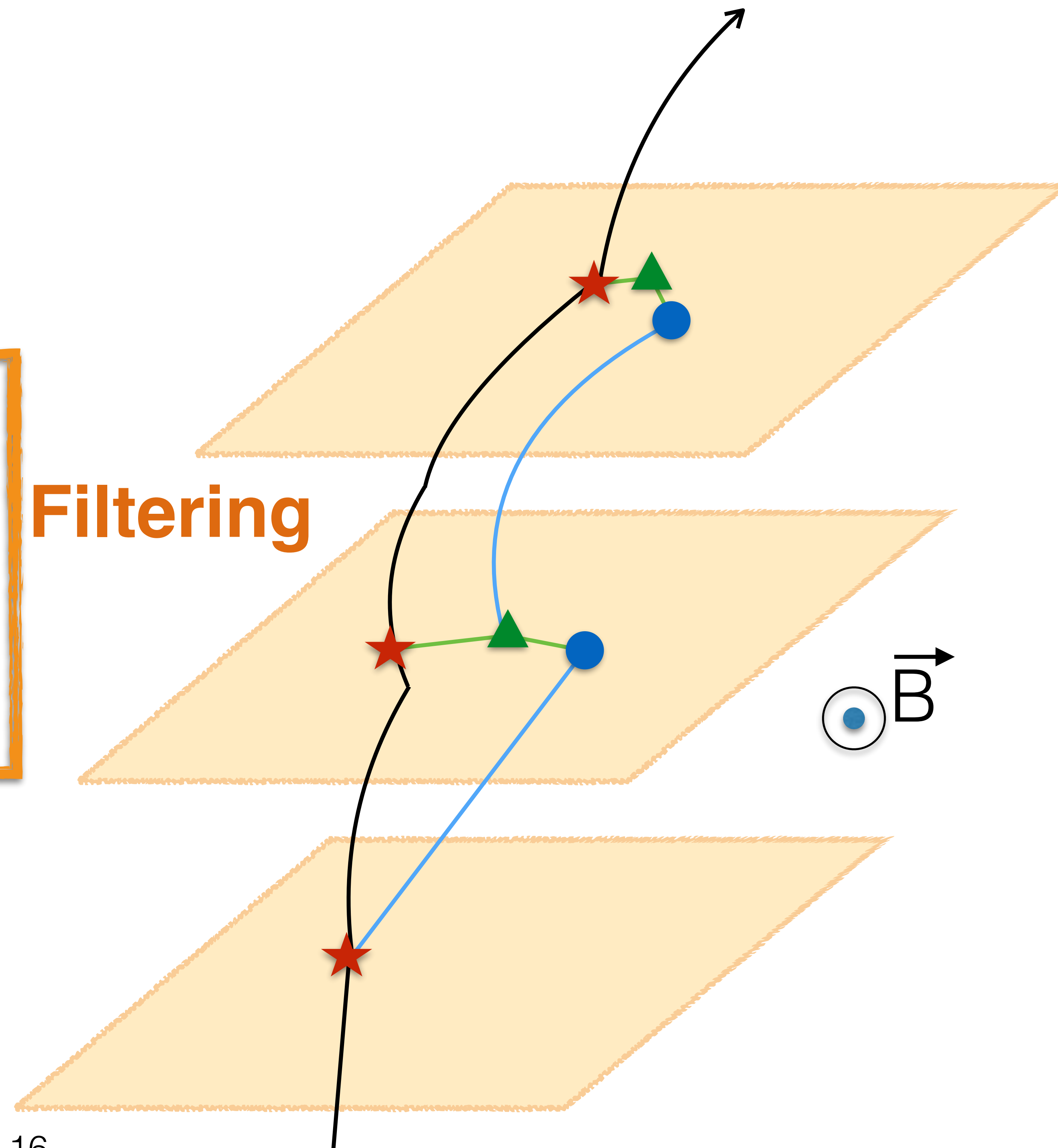
2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).

3. Propagate the first hit to the next layer. Propagator Matrix F.

4. Find the best compromise between the propagated point and the closest hit on the 2nd layer. Use a Chi2 and a Projection Matrix H.

5. **Iterate** 3 and 4 for the next leayers.

Filtering

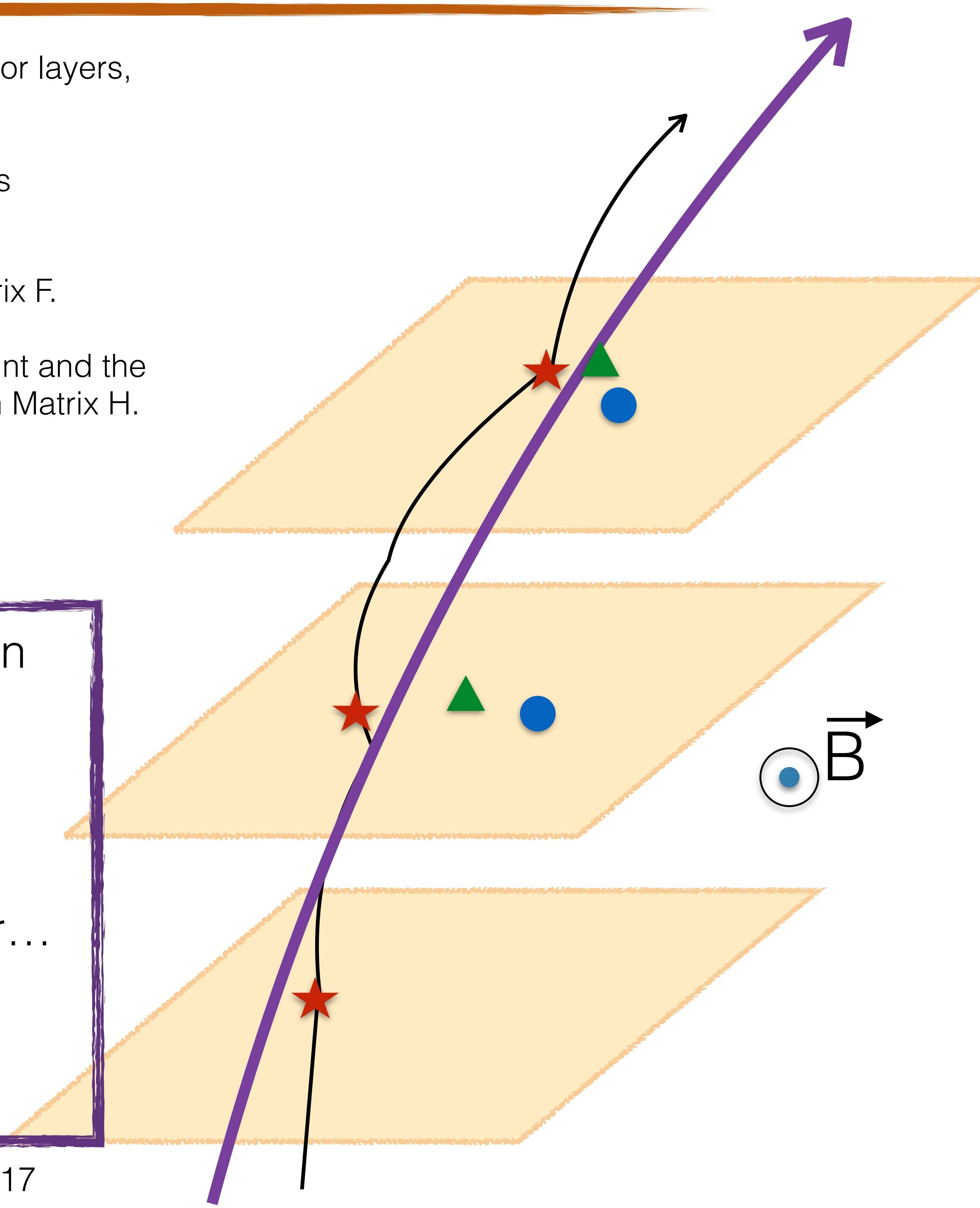


Kalman in pills

1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.
2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).
3. Propagate the first hit to the next layer. Propagator Matrix F.
4. Find the best compromise between the propagated point and the closest hit on the 2^{nd} layer. Use a Chi2 and a Projection Matrix H.
5. Iterate 3 and 4 for the next leayers.

Nota bene

- At each layer we do not evaluate a point in space, but a **curve (helix)** that best approximate the trajectory at ▲ point;
- The curve found has an associated uncertainty that decrease layer after layer…
- …so we redo the filtering backward! (Smoothing)



Kalman in pills

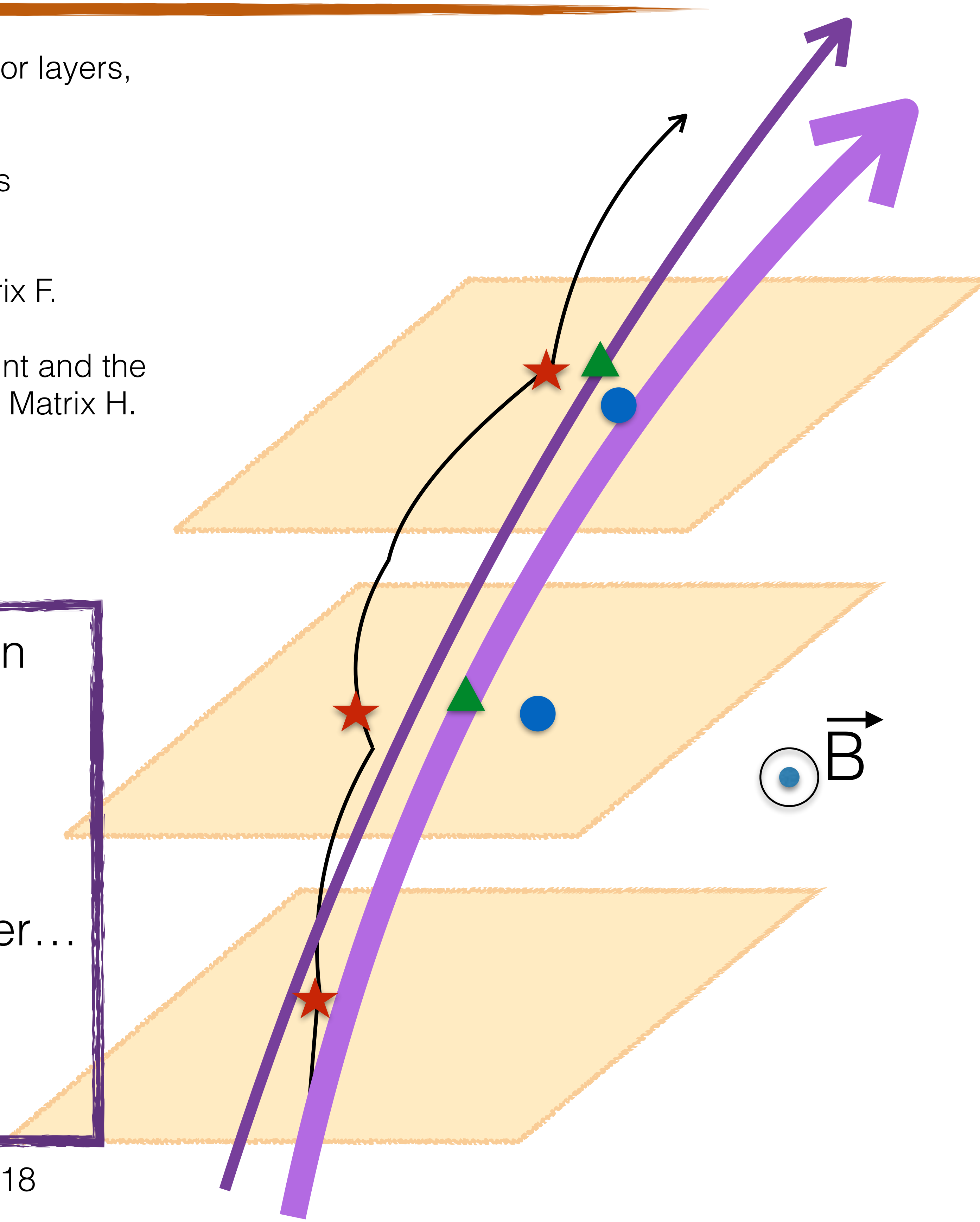
1. Take an ideal particle in vacuum. If we add air + detector layers, trajectory changes due to M.S. and energy loss.

2. We'll see some measurement hits on the detector layers (considering finite detector uncertainty).

3. Propagate the first hit to the next layer. Propagator Matrix F.

4. Find the best compromise between the propagated point and the closest hit on the 2nd layer. Use a Chi2 and a Projection Matrix H.

5. Iterate 3 and 4 for the next leayers.



Nota bene

- At each layer we do not evaluate a point in space, but a **curve (helix)** that best approximate the trajectory at ▲point;

- The curve found has an **associated uncertainty** that **decrease** layer after layer…

- …so we redo the filtering backward! (*Smoothing*)