SAND simulation software overview

Lea Di Noto

University of Genova and INFN Genova

on behalf of SAND detector group

Near Detector meeting, July, 22th 2020

Overview

→ Many **geometries** are under study

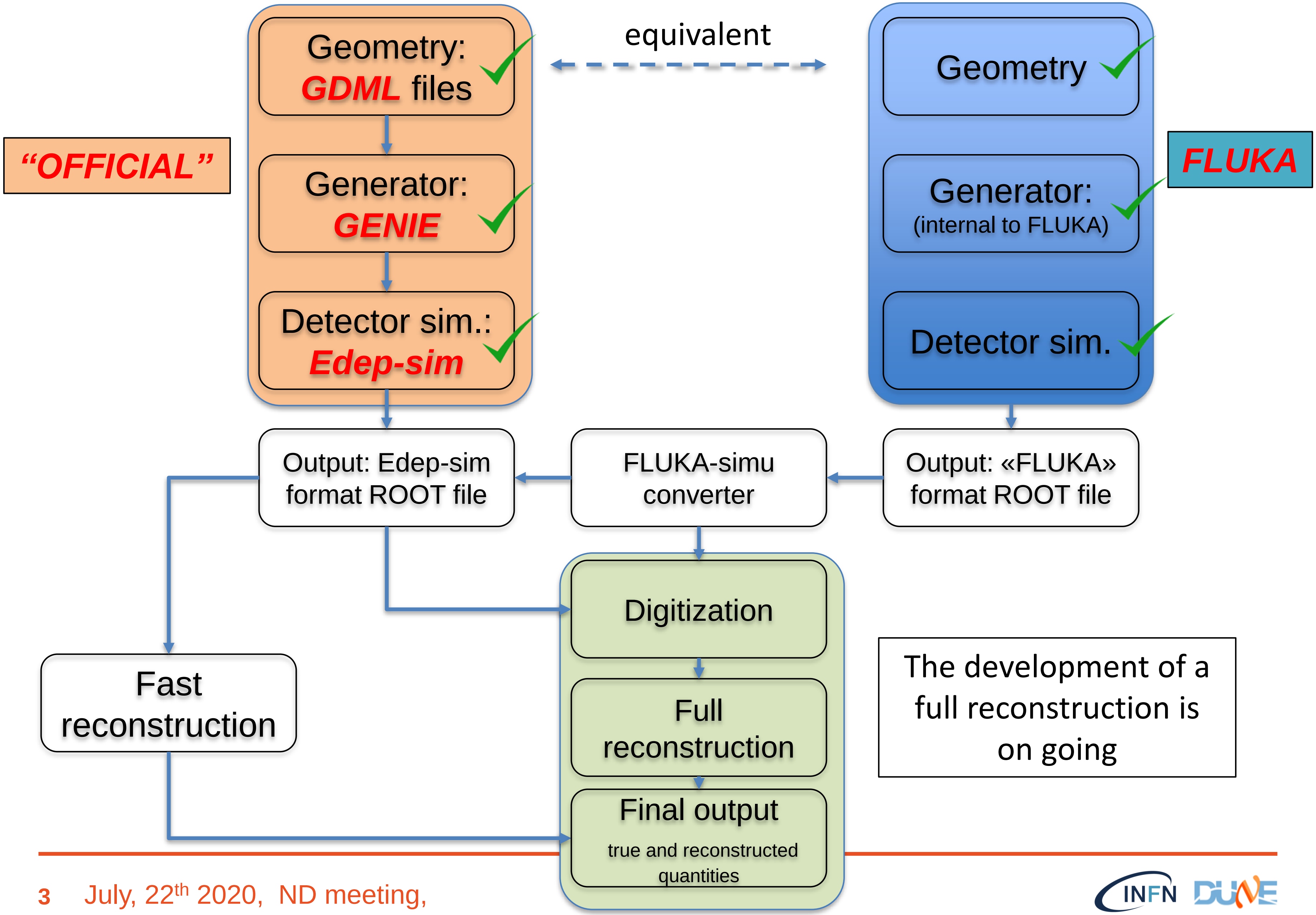
→ Two **generators**: GENIE and FLUKA

→ Two **simulations** packages: edep-sim (GEANT4) and FLUKA

→ A unique **reconstruction code + reconstruction in 3DST**

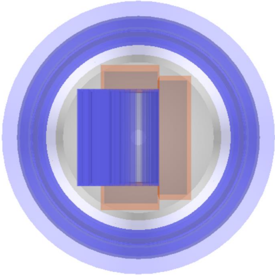
→ + many software tools developed independently

The software in a nutshell

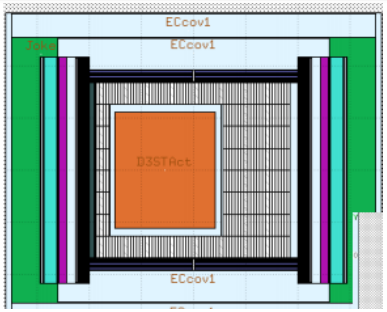


Many geometries

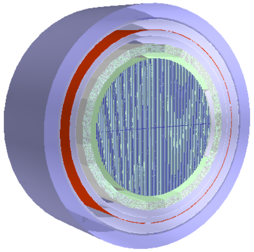
- ECAL + 3DST + TPC → GDML

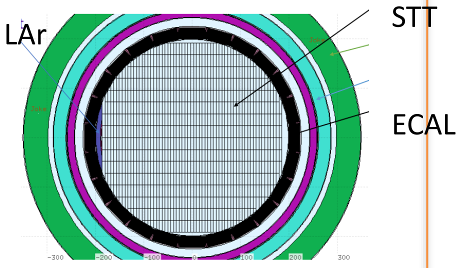


- ECAL + 3DST + STT → GDML and FLUKA



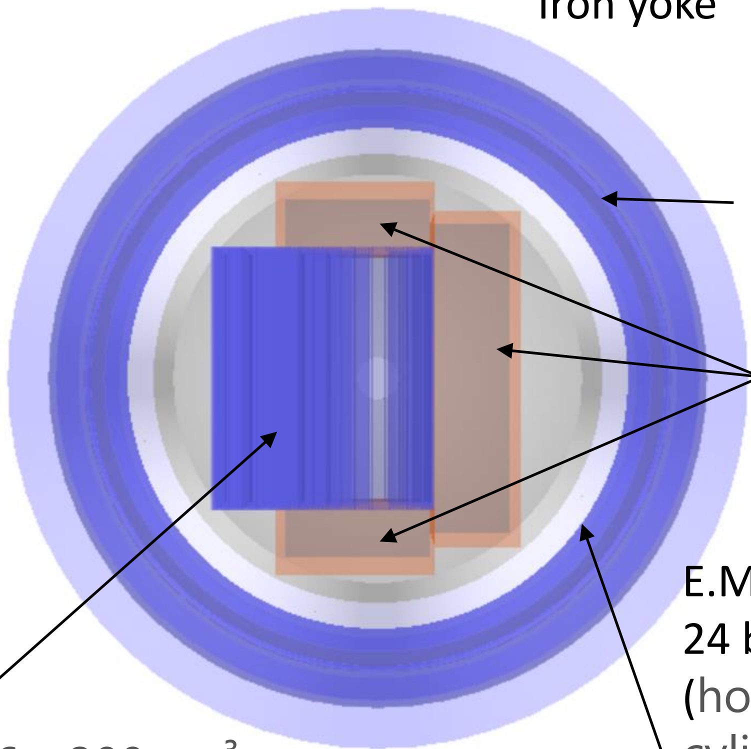
- ECAL + STT only → GDML and FLUKA





GDML geometry: 3DST + TPC

from Guang



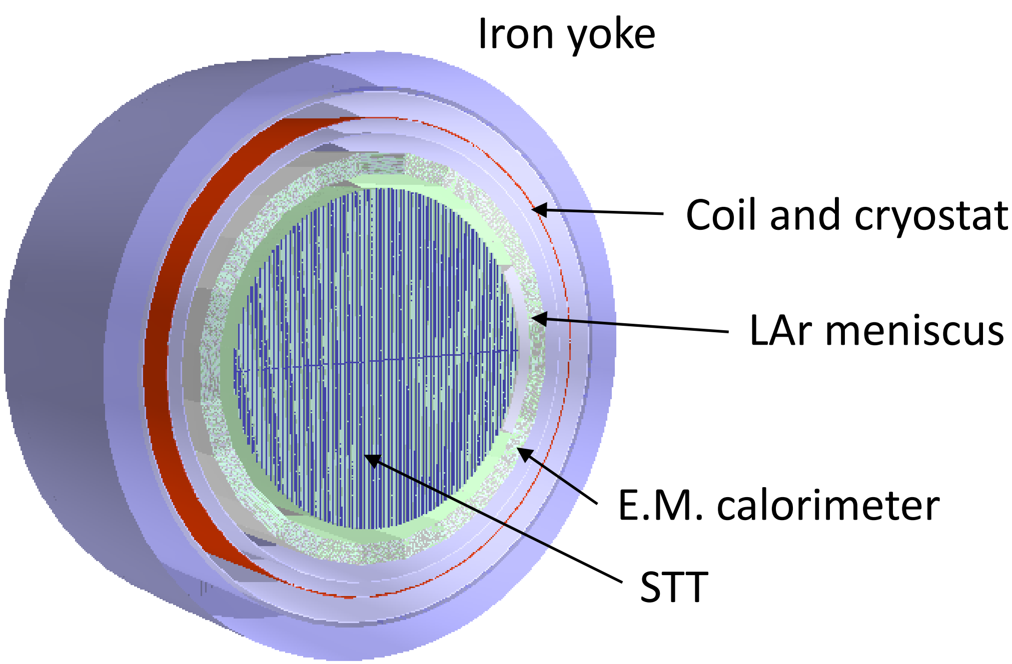
3DST: 252 x 236 x 200 cm^3

E.M. calorimeter:
24 barrel modules
(homogeneous hollow cylinder segments)
2 endcaps:
(homogeneous cylinders)

We will improve this

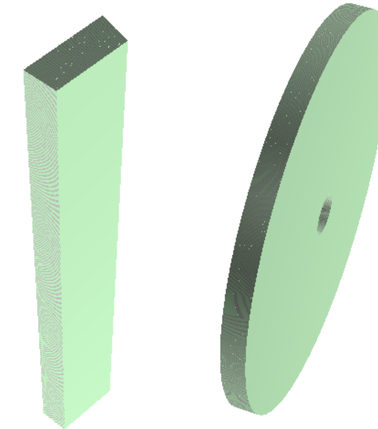
GDML geometry: STT-only

from Matteo



E.M. calorimeter

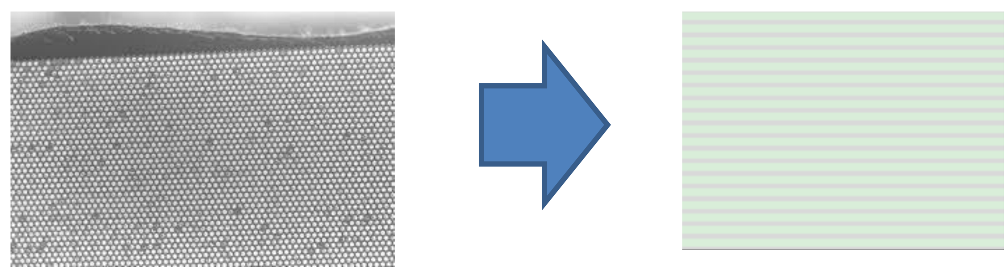
24 barrel modules 2 endcaps



Calorimeter approximated as 209 scintillation layers alternated with 209 lead layers

~ 90 STT modules:

- target (CH2 or C)
- radiator (plastic foils)
- XX straw tube plane
- YY straw tube plane



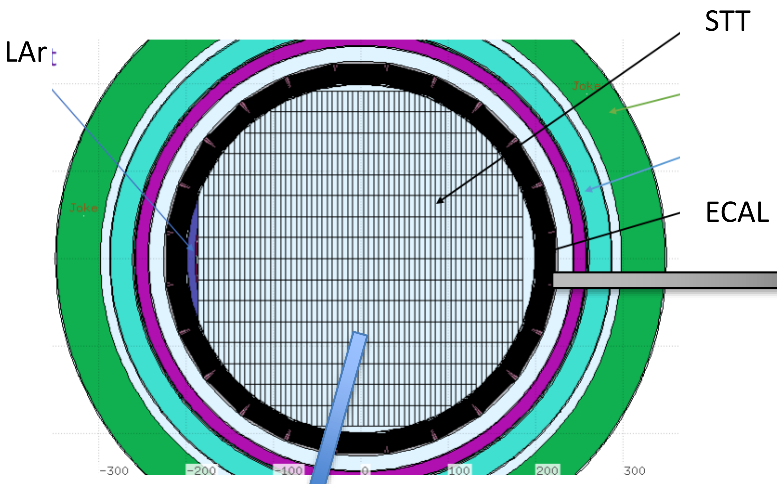
0.7 mm scintillation layer (green)

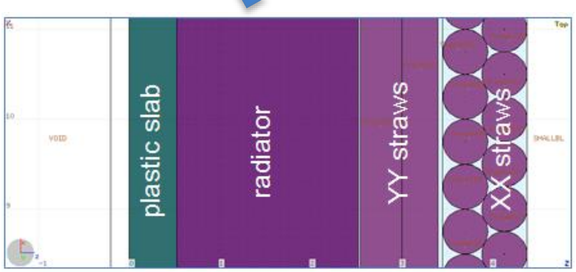
0.4 mm lead layer (gray)

FLUKA geometries:

from Paola, Anna

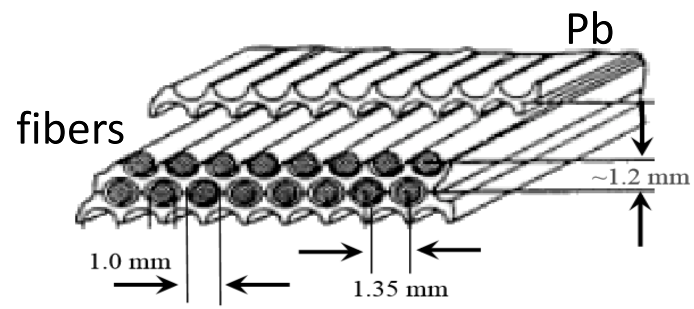
STT only and 3DST + STT



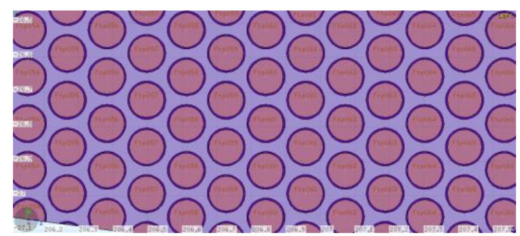


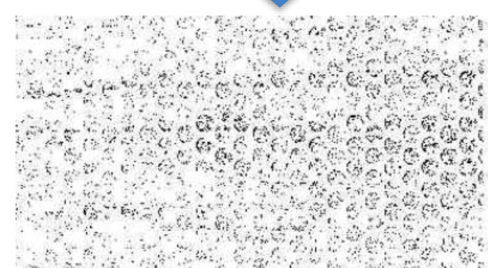
STT modules as in GDML

Very detailed simulation of the calorimeter for barrel and endcaps



Kloecal fine structure





Detail of an ECAL barrel module implemented in FLUKA: fibers, glue, lead

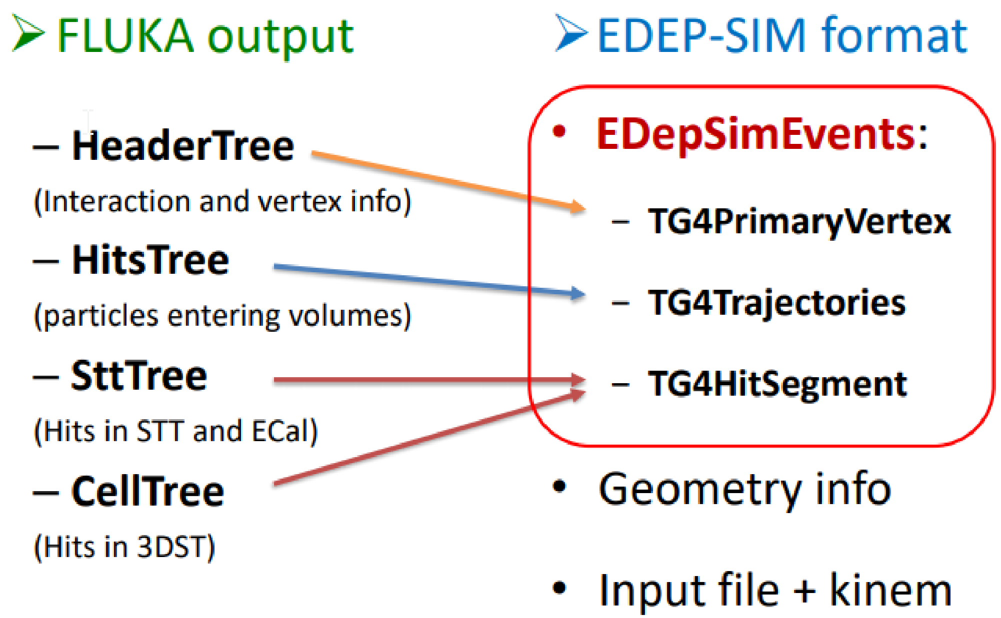
Hits in fibers from simulation output

Why FLUKA simulations?

from Lea, Antonio, Zahra

- It has a different generator
- A more detailed geometry for ECAL was implemented
- It enforces result reliability especially for neutrons and low energy processes

output conversion



→The fluka simulation output was done equal to the edepsim output

→ Differences are only in the geometry information

GENIE

from Carla

- We developed a dedicated code based on GENIE 3 (GENIE 3.00.06) to generate neutrino events in the detector.
- The code:
  - accepts geometry files in gdml format
  - accepts DUNE beam spectrum files in root format
  - takes into account the beam direction (theta=0.101 rad). Beam size set to 3 m.
  - generates any neutrino flavors in the beam and all possible interactions but it is possible to change:
    - the neutrino flavor list
    - the GENIE tune (default: G18_02a_00_000)
    - the GENIE event generator list (default: Default)

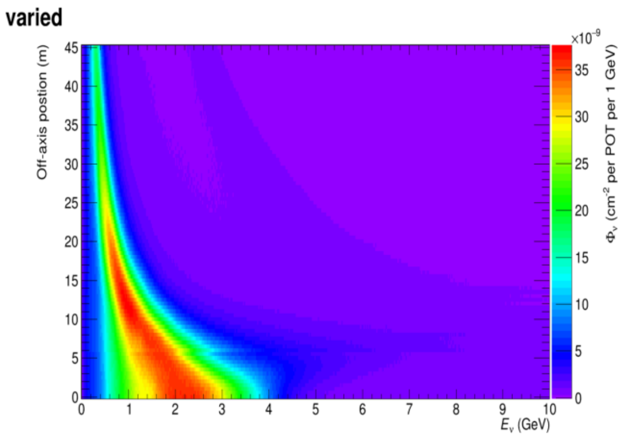
THE GENIE version can be set equal to other ND simulations

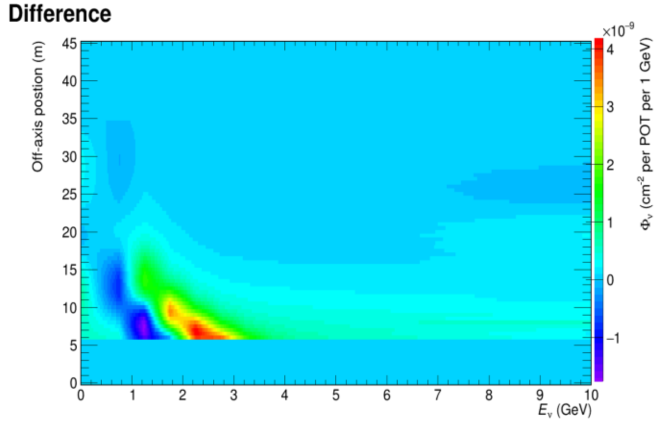
Variated beam simulations

from Guang

The neutrino flux can be:

- taken from rootfiles **for normal beam in N**D hall delivered by beam people https://home.fnal.gov/~ljf26/DUNEFluxes/
- **directly generated** (by using dk2nu) from beamline g4 (g4lbnf) for different beam variation studies https://cdcvs.fnal.gov/redmine/projects/dk2nu/wiki
  → For some beam variations some dk2nu files are already available from the samples that was used for the beam uncertainty evaluation,
  → For other beam variations (not in the table) we need our own g4lbnf runs.





Beam parameter	Variation
Horn current	+3 kA
Water layer thickness	+0.5 mm
Decay pipe radius	+0.1 m
Proton target density	+2%
Proton beam radius	+0.1 mm
Proton beam offset X	+0.45 mm
Proton beam θ, ϕ	0.07 mrad θ, 1.57 ϕ
Proton beam θ	0.070 mrad
Horn 1 X shift	+0.5 mm
Horn 1 Y shift	+0.5 mm
Horn 2 X shift	+0.5 mm
Horn 2 Y shift	+0.5 mm

Digitization

- ECAL: tuned parameters implemented based on the KLOE published papers

- STT: space and time resolution, energy threshold included

- 3DST: saturation, light yield in scintillator, fiber attenuation MPPC, efficiency, ADC response included

- TPC: it is still missing (we are using a fast reconstruction for detected particles)

Digitization: ECAL

from Federico, Lea

- Detailed digitization of the ECAL response takes into account:
  - Number of photons per deposited energy; scintillation time; attenuation and propagation time along the fibers; response of PMT
- Reproduction of measured performances:

Time resolution

Intrinsic resolution:
$\sigma_t = 54\ \text{ps}/\sqrt{E(\text{GeV})} \oplus 50\ \text{ps}.$

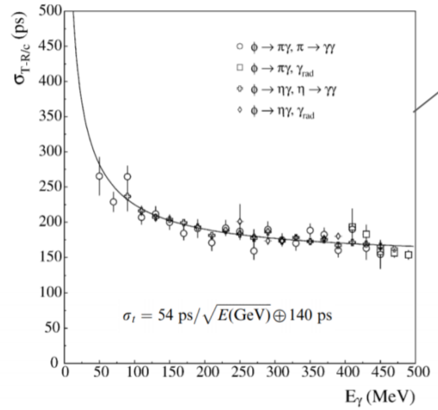
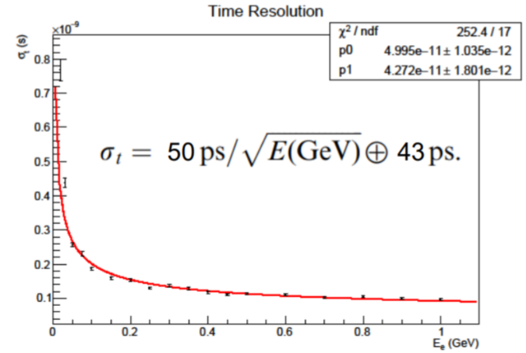


Fig. 32. Time resolution as a function of E_γ for ϕ radiative decays.



Energy resolution

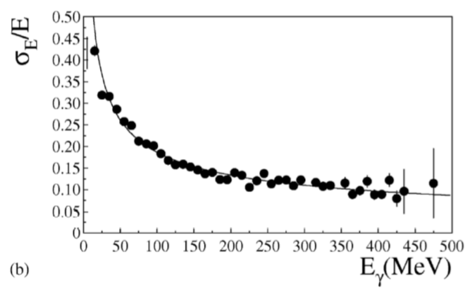
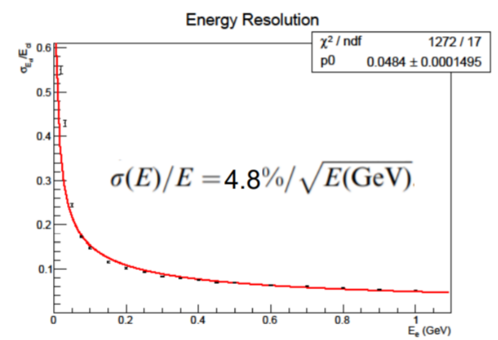


Fig. 20. $(E_{cl} - E_\gamma)/E_\gamma$ (a) and resolution (b) vs. E_γ for $e^+e^-\gamma$ events. The fit gives $\sigma(E)/E = 5.7\%/\sqrt{E(\text{GeV})}$.



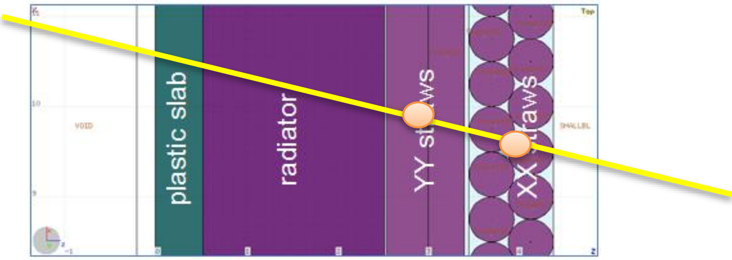
NIM A 482 (2002) 364-386

Digitization: STT

From Antonio, Matteo, Lea

- STT space-resolution simulated by means of Gaussians
  - 0.2 mm for transversal coordinate,
  - 0.1 mm for Z coordinate
- Energy threshold for STT-hits: 0.1 keV

- For any charged particle in MC-tracks, hits for each STT plane are grouped to get the "STT-digits" in X-Z and Y-Z views
  - Digit coordinates from the average of hit coordinates
  - Time-resolution on STT digits: 1 ns (Gaussian smearing)



Reconstruction strategy

From Antonio, Matteo

(without any MC info)

- **Step 0** Vertex reconstruction based on STT-hit topology
- **Step 1**
  - Track finding (global transform method)
  - Linear or circle fits of the tracks
- **Step 2** - Vertex reconstruction from crossing of 2 most "rigid" tracks
- Possible reiteration procedure (step1 and 2)

} in both views (Y-Z and X-Z)

- **Step 3** Track matching ⇒ 3D track
- $p_\perp$ from Larmor radius
- dip-angle λ from x-vs-ρ fit

} momentum estimate $p = p_\perp / \cos \lambda$

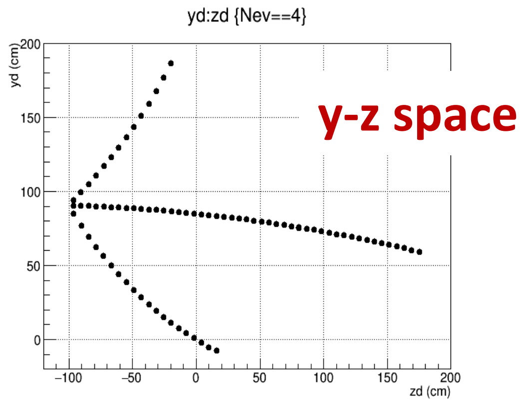
- ECal hits compatible with tracks ⇒ ToF measurement

  ⇒ β estimate for each track ⇒ charged particle Id …

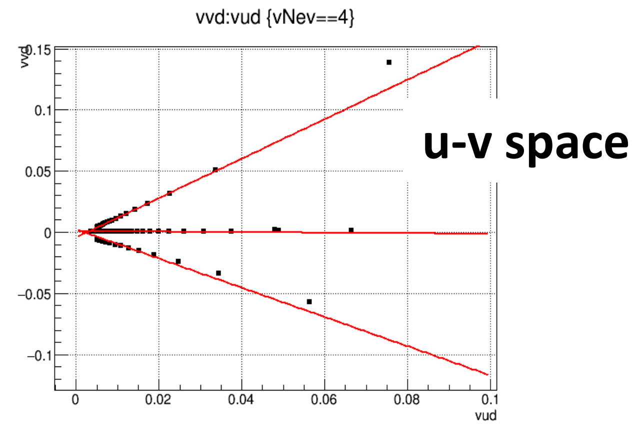
From Vertex to Track reconstruction

Coordinate transformation by using reco-Vertex (z_V,y_V):

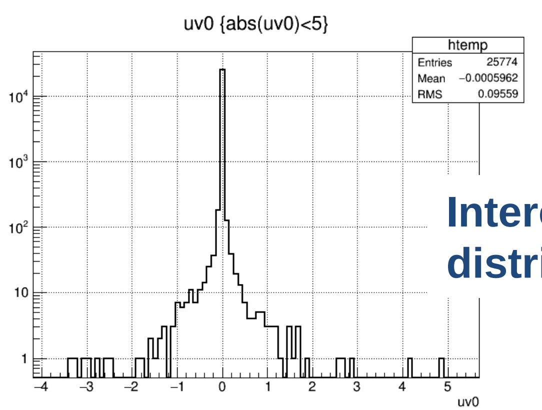
$$u = +(z-z_V) / [(z-z_V)^2 + (y-y_V)^2]$$
$$v = -(y-y_V) / [(z-z_V)^2 + (y-y_V)^2]$$



Curved trajectories become straight lines



Parameters of tracks in u-v space

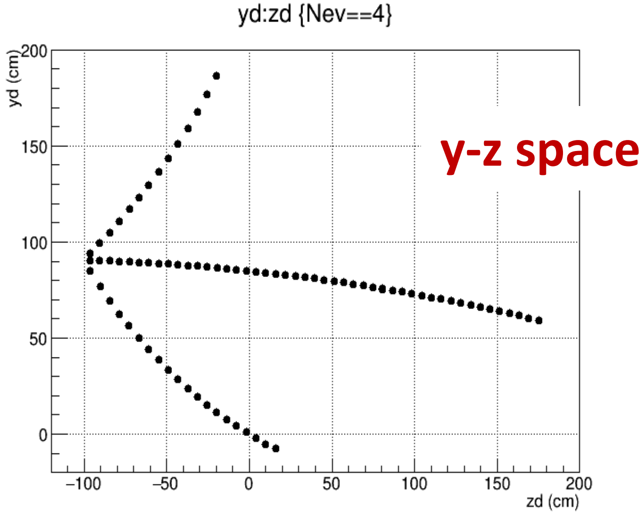


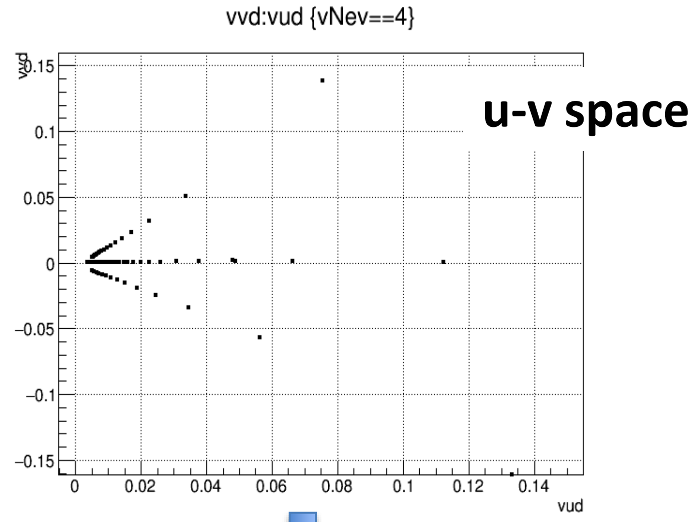
Intercept distribution

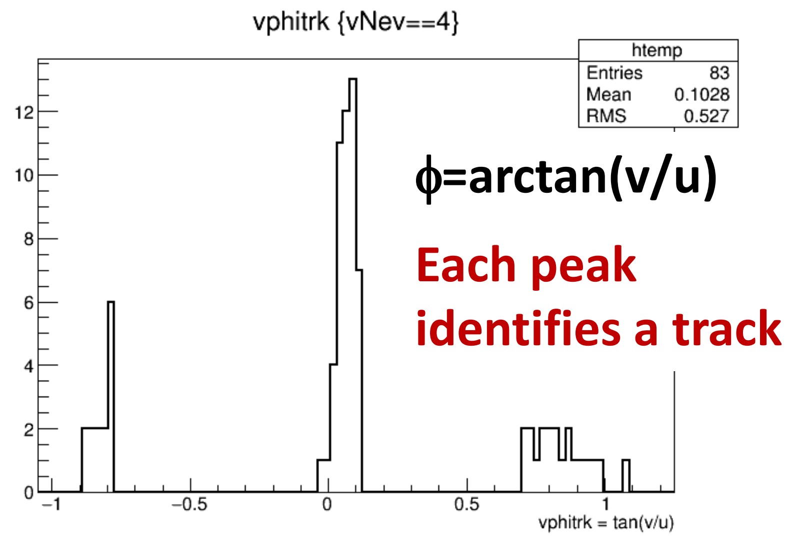
Intercepts more peaked at 0 if Vertices are well reconstructed

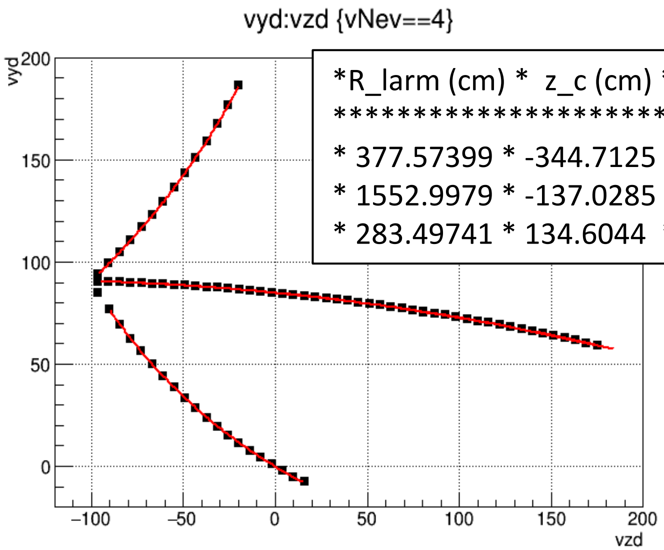
Each trajectory is fully reconstructed!

Track reconstruction by fits









*R_larm (cm) *	z_c (cm) *	y_c (cm) *
* 377.57399 *	-344.7125 *	378.457
* 1552.9979 *	-137.0285 *	-1462.071
* 283.49741 *	134.6044 *	249.4256

Track finding and fits

Identification of charged track

From 3D-track: evaluation of Track-Length (L) and Time of Flight (ToF)

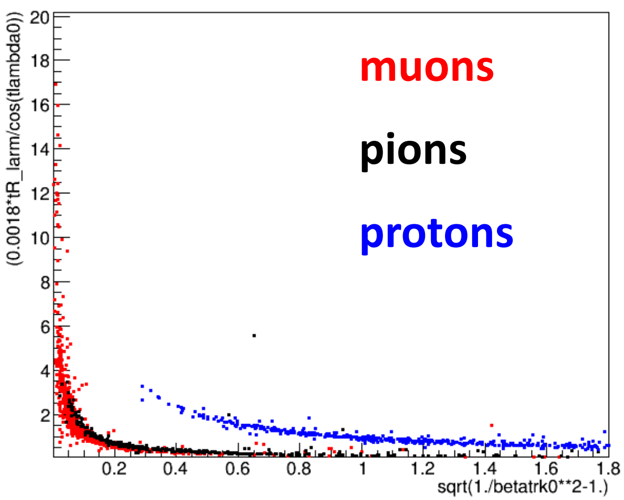
⇒ **velocity estimate:** $\beta = L / ToF$

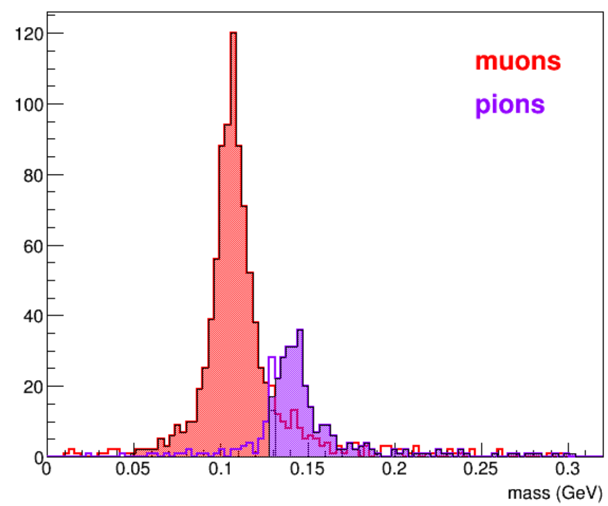
✓ **L from sum of distances between STT-digits along the 3D-trajectory**

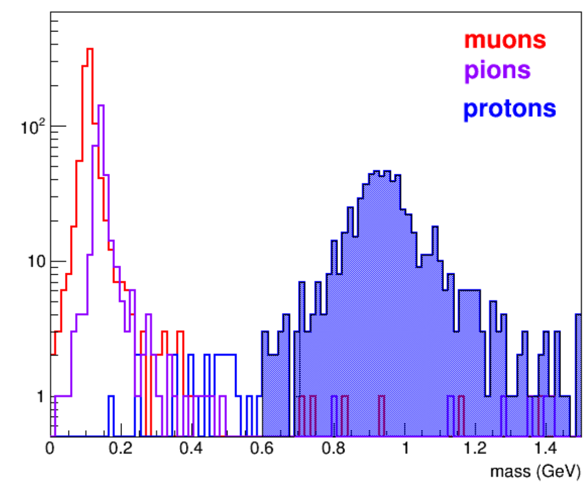
Particle identification → $m = p / \beta\cdot\gamma = p\cdot\sqrt{(1/\beta^2 - 1)}$

→ **p vs $1/(\beta \cdot \gamma)$**

Events with no more than 3 matched tracks in the two views:



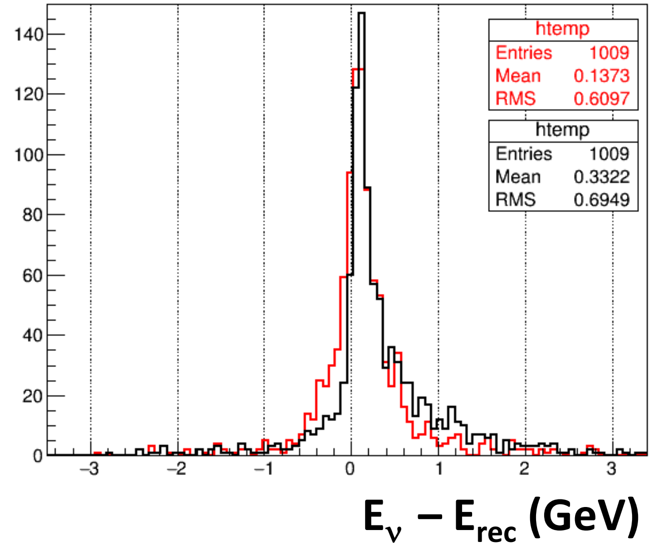




✓ **For those plots: ToF from MC-times of STT-digits time resolution NOT included!**

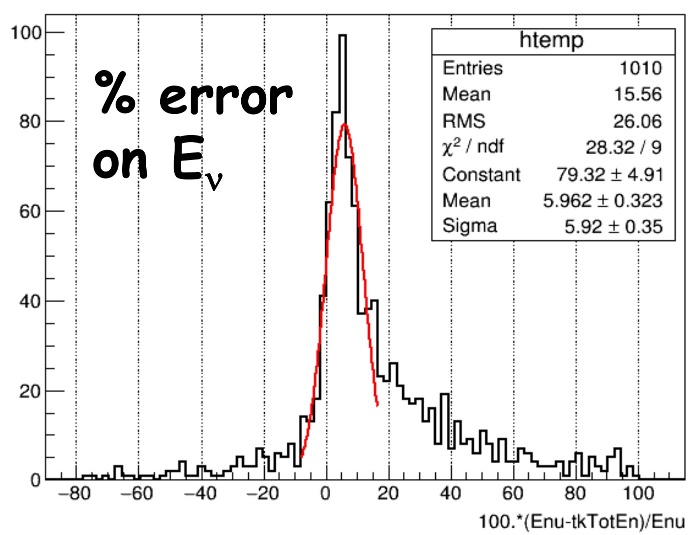
Neutrino energy reconstruction

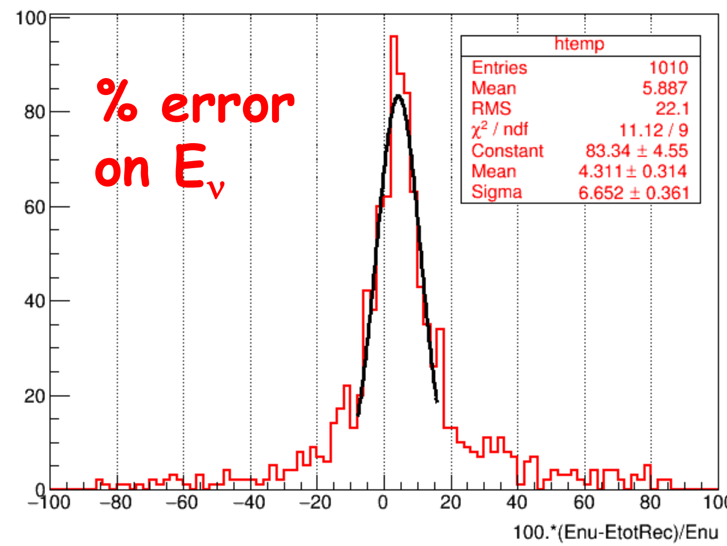
(preliminary)



'All-tracks' energy only

'All-tracks' energy + Off-track ECal energy

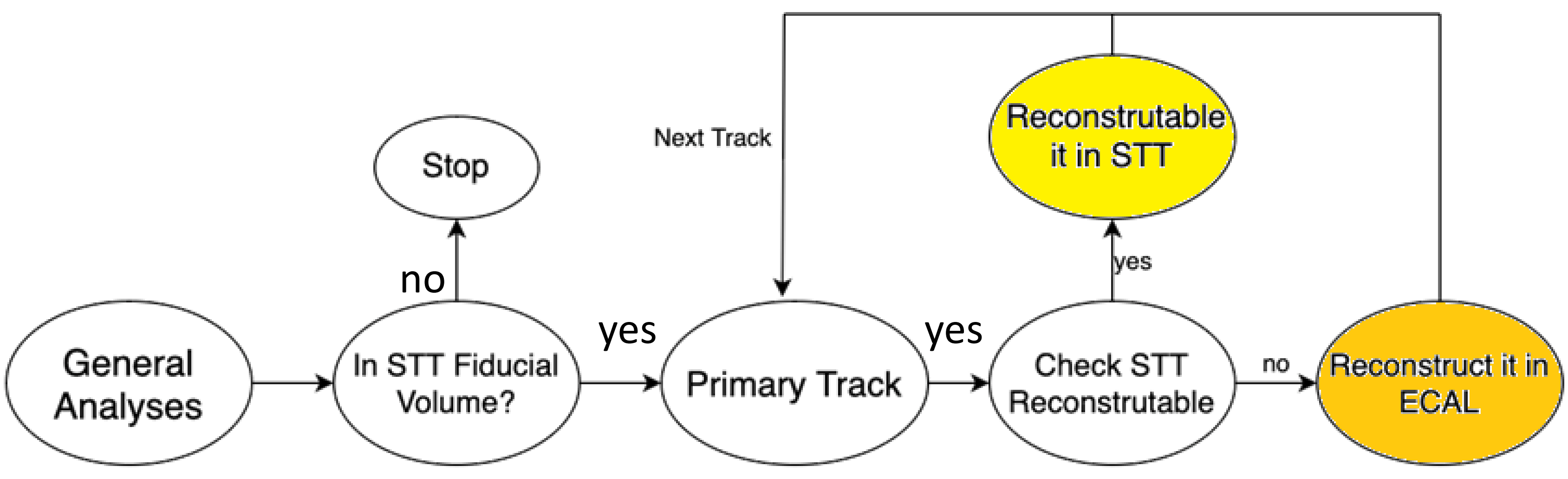




Fast Reconstruction

From Bing

- Based on full detector simulation edep-sim (GEANT4)
- Treat events originating in different detector region differently
- Algorithm depends on specific analysis considered
- Single-particle smearing based on dedicated analysis/reconstruction



Fast Reconstruction in STT

- Charged particles: Check number of Y hit in STT:
  - N(Y) < 4 (6) Stop. No smearing.
  - N(Y) >= 4 (6) Smear it.
- Charged particles: Momentum and angle smearing:
  - Gluckstern formula:
    - Based on track length, N(y), B, X0, single hit resolution.
  - Circular fitting and linear fitting
    - Need smeared position of every hit
- Neutral particles: Check its decay products:
  - Charged
  - Neutral

Neutral Particles Reconstruction

- $\pi^0 \rightarrow 2\gamma$ or $\pi^0 \rightarrow \gamma + e^- e^+$
  - Reconstruct each daughter particle's momentum separately then summing up.
- γ: $e^- e^+$ pair in STT or e.m. shower in ECAL.
  - Convert in STT: Reconstruct $e^- e^+$ track in STT
  - Convert in ECAL: find calibrated energy deposition of the e.m. shower
    - Smear earliest hit position by its resolution, connecting with vertex gives momentum direction
- **NEUTRONS**: hits/cells detached from primary vertex.
  - **Interaction in STT**: connecting first hit (smeared) to vertex (or first hit for single track) gives direction, reconstructing the daughter tracks gives momentum.
  - **Interaction in ECAL**: detached cells are used to define neutral clusters, calibrated energy deposition in the cluster is summed up, connecting earliest cell to the vertex (or first hit for single track) gives momentum direction.
  - **Neutron energy in CC**: time-of-flight from smeared timing at primary vertex (or first hit) and earliest hit of detected neutron candidate and reconstructed direction.
  - **Neutron energy in CC on Hydrogen**: calculated analytically from energy-momentum conservation.

Data output from reconstruction

Event information

Type	Name	Description
double	x	X coordinate of the neutrino interaction
double	y	Y coordinate of the neutrino interaction
double	z	Z coordinate of the neutrino interaction
double	t	Time coordinate of the neutrino interaction
double	Enu	Neutrino energy
double	pxnu	X component of the neutrino momentum
double	pynu	Y component of the neutrino momentum
double	pznu	Z component of the neutrino momentum
double	Enureco	Reconstructed energy of the neutrino
double	pxnureco	Reconstructed x component of the neutrino momentum
double	pynureco	Reconstructed y component of the neutrino momentum
double	pznureco	Reconstructed z component of the neutrino momentum
std::vector	particles	Vector of particles produced in the neutrino interaction

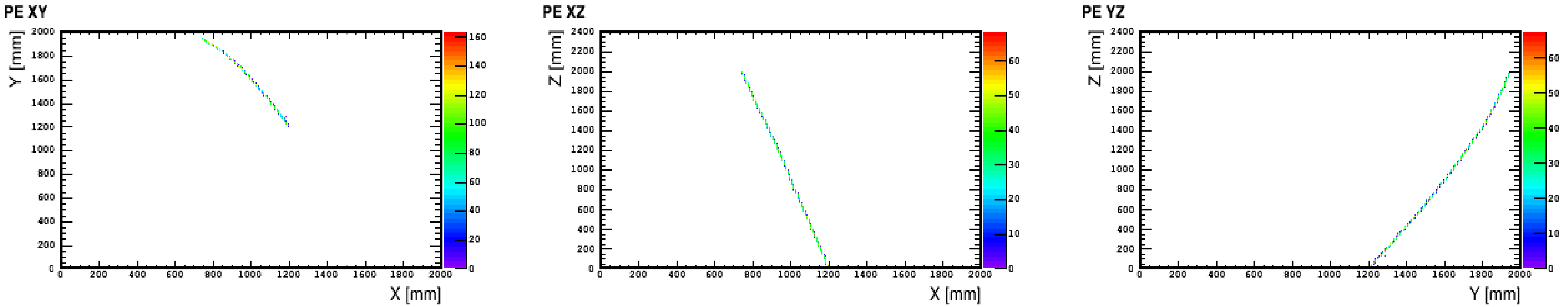
Particle information

double	pxreco	Reconstructed X component of the particle momentum
double	pyreco	Reconstructed Y component of the particle momentum
double	pzreco	Reconstructed Z component of the particle momentum
double	Ereco	Reconstructed particle energy
double	xreco	Reconstructed x coordinate of the starting point of the particle's track
double	yreco	Reconstructed y coordinate of the starting point of the particle's track
double	zreco	Reconstructed z coordinate of the starting point of the particle's track
double	treco	Reconstructed time coordinate of the starting point of the particle's track
bool	has_track	True if a track is associated to the particle, false otherwise
double	charge_reco	Reconstructed charge of the particle
track	tr	Track associted to the particle
bool	has_cluster	True if an em calo cluster is associated to the particle, false otherwise
cluster	cl	Cluster associated to the particle
bool	has_daughter	True if particle has daughter, false otherwise
std::vector	daughters	Vector of the daughter particles

3DST digitization

From Guang, Clark McGrew and Sergey Martynenko

- Input edep-sim and output :
- - analysis tree containing final state particle high-level information
- - three 2D readout maps considering:
- Saturation, light yield in scintillator, fiber attenuation, MPPC efficiency, ADC response.
-



3DST Reconstruction

- Developing a new reconstruction tool dedicated for 3DST and superFGD by Clark McGrew and Sergey Martynenko

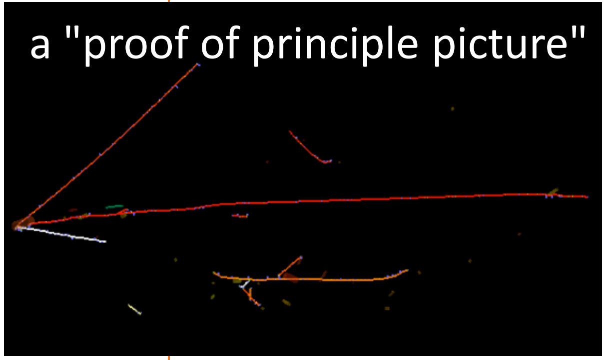
- **Functioning packages**:
  - Read the input file containing fiber hit information;
  - Create 3D Hits from fiber hits;
  - Adjust charge for 3D Hits;
  - Cluster 3D Hits (DB Scan);
  - Define hits order inside each cluster (Minimum Spanning Tree);
  - Split clusters into Track-Like objects (find vertices);

- **In development**:
  - Track fit;
  - Shower search;
  - Other?

3DST reconstruction

- Sergey's conclusion at this point:

- Test reconstruction techniques as a set of separate root scripts:
  - 3D hits are created and clustered;
  - Clusters are split into track-like objects;
- First look at the effectiveness of pattern recognition algorithms:
  - Hit finding works well with crosstalk(small amount of Ghost Hits);
  - Charge Adjustment works with crosstalk;
  - Clustering and track splitting works well with crosstalk, but quantitatively tested only without crosstalk
- Temporary code is on GitHub:
  - https://github.com/rennney/CubeRecon
- Future:
  - Understand True information in MC with crosstalk to quantify clustering properly;
  - Define efficiency for complex events;
  - Continue working on Track fitting and Shower search algorithms

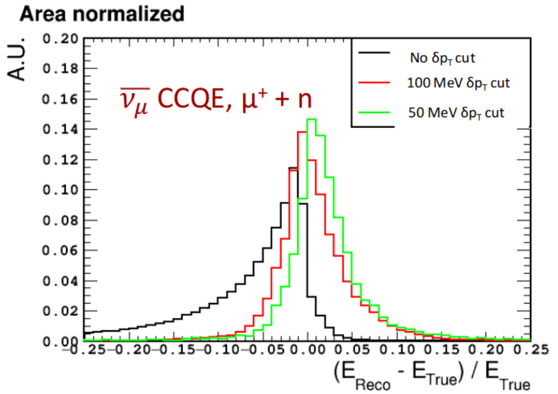


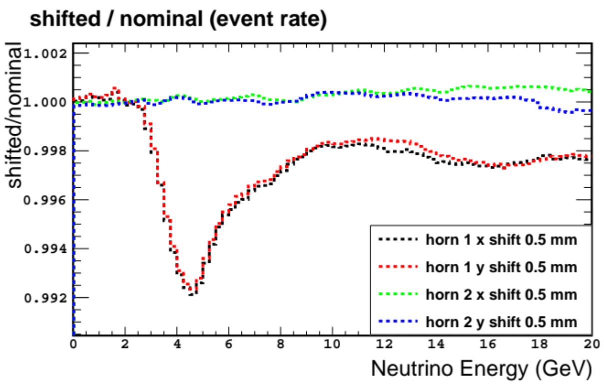
the "lines" are the reconstructed tracks with colors based on the measured dEdX.

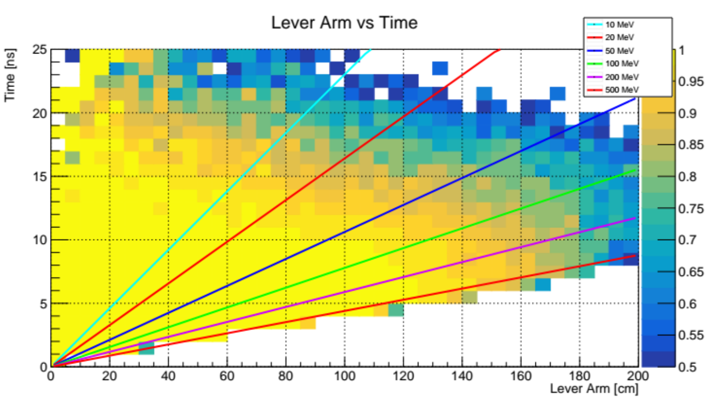
Many analyses tools

- A package has been created compiling all current analysis tools: https://github.com/gyang9/DUNE3dstTools

NuModel	Ar-C model tuning
beamMonitoring	Beam monitoring
elecSim	Electronics simulation
fluxSTV	Single transverse variable with neutron measurement
nBKG	Neutron out-of-FV background evaluation (from Manoa)
reco	
CMakeLists.txt	Reconstruction (from Sergey)







Conclusions and future work

- **Quite mature analysis chain**
  - →Official tools for geometry, generator and detector simulation
  - →Equivalent flow with FLUKA
  - →Converter from FLUKA to edep-sim format to use same reconstruction software
  - →Digitization done for different detectors (except TPC)
  - →Event reconstruction without (or minimal) use of MC info; work in progress to avoid use of MC info at all.
  - →Fast reconstruction from hits/single-particle studies to speed-up analysis
- **Many other software and analysis tools**

- Full reconstruction have to be finalized
- 3DST reconstruction should be integrated
- an integration with the ND data model have to be implemented