Simulation of a TPC for the ILC Detector

Mokka and Beyond

Adrian Vogel
DESY FLC

Mokka Miscellany – What's New?

Improved TPC geometry

- thin cathode plane in the middle of the chamber
- more realistic endplate (field cage still simplified)

Some helpful plugins for Mokka

- `LogPlugin` writes log files with timestamps
- `MagPlugin` shows field vectors and flux lines
- `MarkerPlugin` draws 3D markers and rulers
- `MaterialPlugin` prints the table of materials

See the readme files and try them out!

Overview

- Why is a TPC simulation difficult?
- How can you solve the problems?
- What do we currently have in Mokka?
- Comparisons: What do you get?
- Digitisation and reconstruction
- TPC software: Status and plans

Tracking – Behaviour of Geant 4

Geant 4 transports particles step by step

Step length is determined by the minimum of:

- the distance to the next physical volume boundary
- the free path to the next discrete physics process (for all applicable processes, randomised)
- the limit of the step length

Discrete processes (e. g. decay) cause a step to end

Continuous processes (e. g. energy loss by ionisation) are applied after a step has ended for some other reason

Tracking – Problems with the TPC

The fundamental process in a TPC is ionisation

- TPC contains low-density material (gas)
- discrete processes are rare, steps are long
- small number of rather large energy deposits

A real TPC is read out by small anode pads

- large number of small energy deposits
- needed for tracking and dE/dx

Steps in the simulation need to be broken down

- introduction of artificial volume boundaries
- limitation of the step length

Option 1 – Layers

Implementation

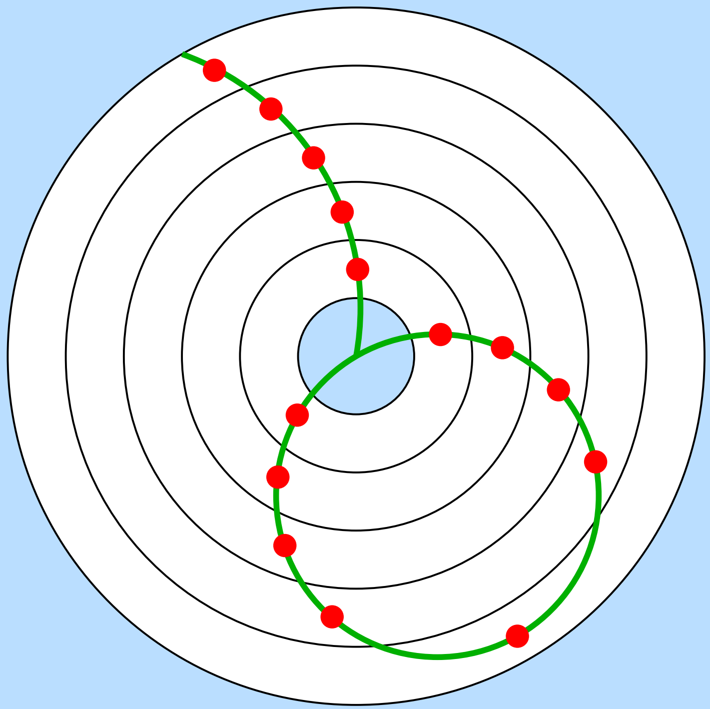
- segmentation in ρ
- divide the TPC volume into 200 “layers” of gas
- sum up the energy deposits in each layer $\longrightarrow$ hits

Pros

- simple and fast
- suitable for high-p_t tracks

Cons

- information loss for low-p_t tracks
- hard-coded readout geometry



Option 2 – Voxels

Implementation

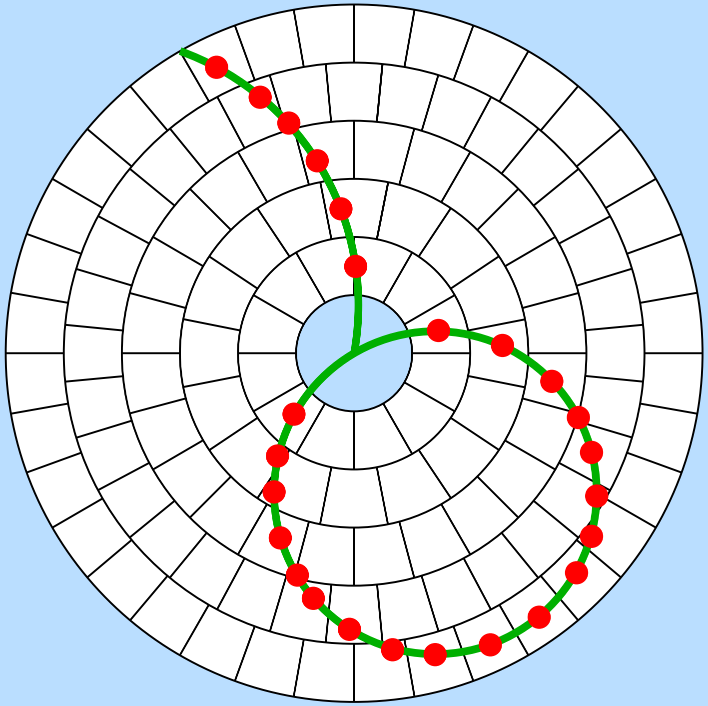
- segmentation in ρ, φ, and z
- divide the TPC volume into layers, wedges, and disks
- sum up the energy deposits in each voxel $\longrightarrow$ hits

Pros

- realistic information for all tracks

Cons

- slow navigation
- hard-coded readout geometry
- many hits, large output files



Option 3 – Step Limits

Implementation

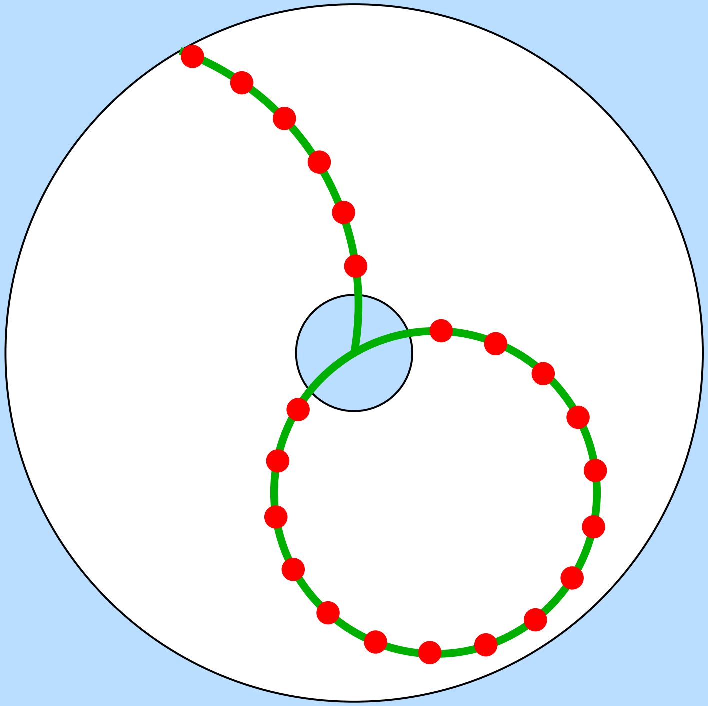
- segmentation in the direction of flight
- assign maximum step length to the TPC volume
- write out energy deposit for each step → hits

Pros

- realistic information for all tracks
- simple and fast

Cons

- binning effects possible
- very large output files possible



Step Limits in Geant 4

In the physics description

- implemented as a “pseudo-process” `G4StepLimiter`
- not included in the built-in physics lists of Geant 4
- added to the selected physics list in Mokka at runtime
  (for all long-lived charged particles)

In the geometry description

- attach an object of the class `G4UserLimits`
  to a logical volume (the TPC gas, in this case)

Cuts in the TPC

Minimum energy deposit of a step

- need at least $\Delta E = 32\,\mathrm{eV}$ for a hit (Argon ionisation)

Minimum kinetic energy of a track

- steering parameter (Mokka default is 10 MeV)
- particles with $E < 10\,\mathrm{MeV}$ curl on one pad
  (So what? They're nevertheless there!)
- what about delta electrons and background hits?
- don't make the simulation too friendly!

`G4UserSpecialCuts` $(\ell_{\mathrm{max}}, t_{\mathrm{max}}, E_{\mathrm{min}}, R_{\mathrm{min}})$

- available, but currently not used

Implementations of the TPC in Mokka

Option 1 – Layers

- available since the first Mokka release
- only minor modifications over the years
- used in all currently predefined geometry models

Option 2 – Voxels

- proof of principle: it works (with $\mathcal{O}(10^9)$ voxels)
- not released to the public (significantly slower)

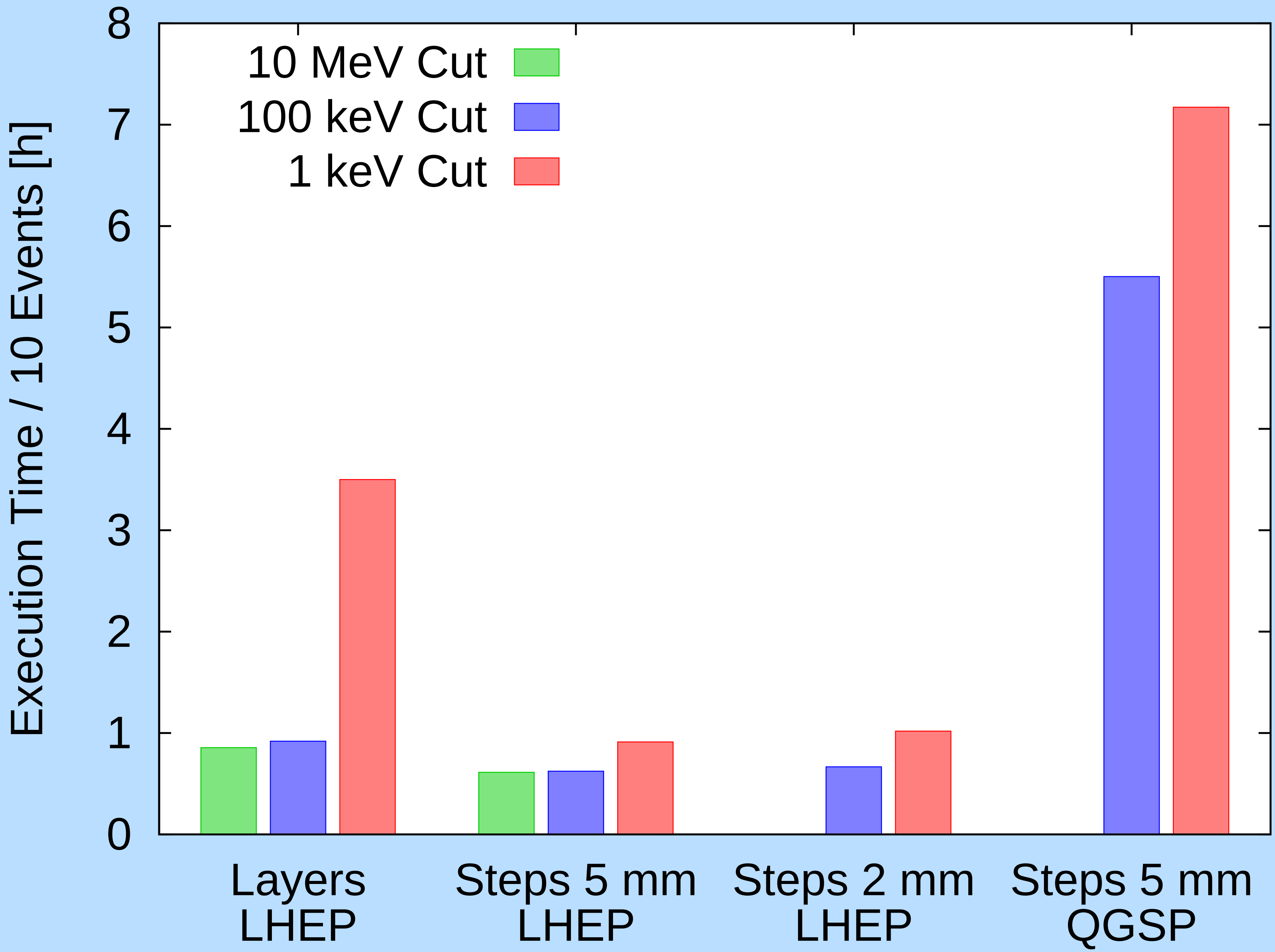
Option 3 – Step Limits

- available since Mokka 06-00 (driver `tpc04`)
- not used in any predefined geometry model yet

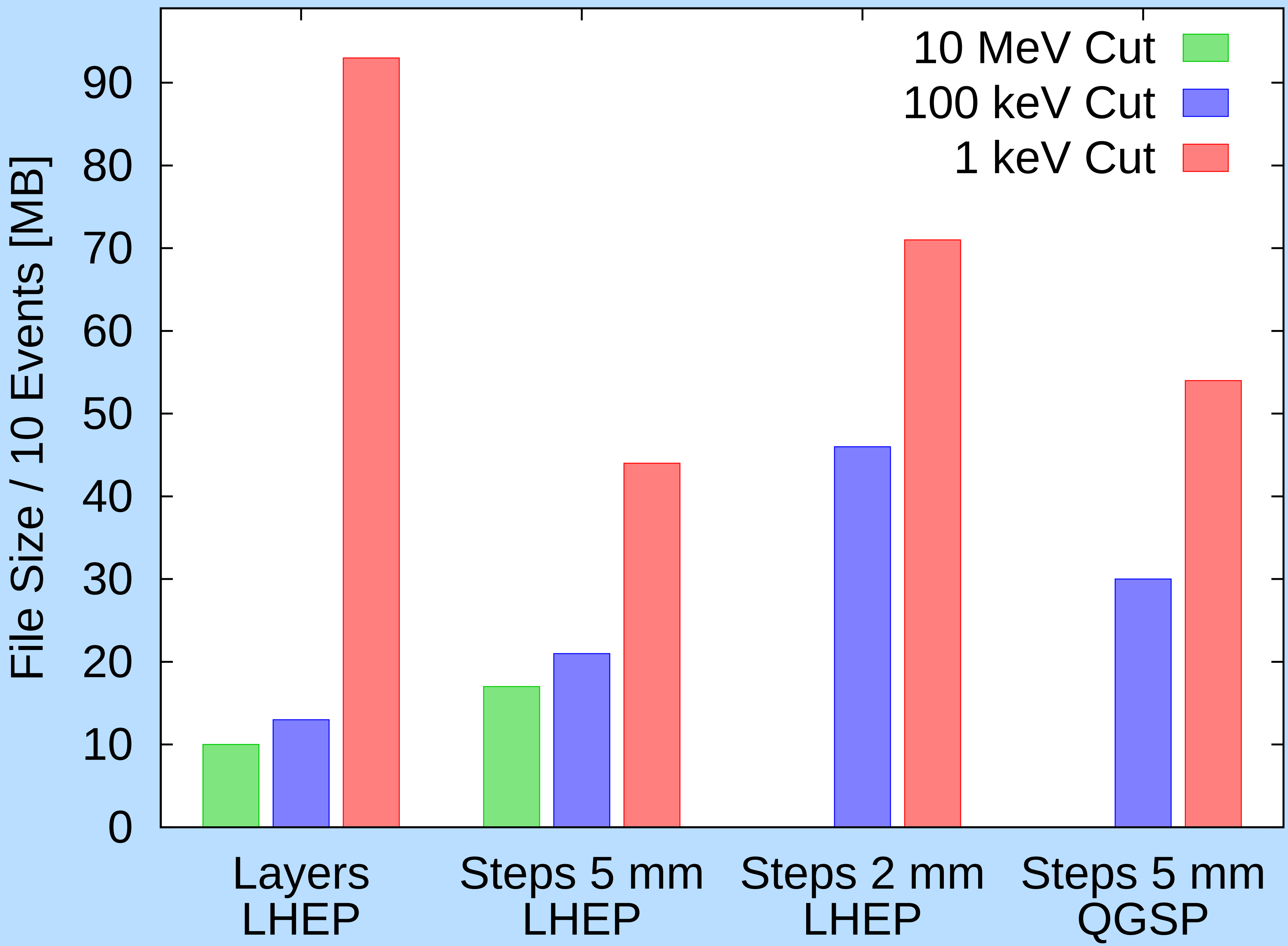
Test Runs with Mokka

- 10 events $e^+e^- \rightarrow t\bar{t} \rightarrow 6j$ with $\sqrt{s} = 500\,\text{GeV}$
- Physics lists `LHEP` and `QGSP_BERT_HP`
- Geometry model `LDC01_01Sc`
  (with `tpc04` replaced by `tpc06` + `etd00`)
- Steps in the TPC limited to 5 mm and 2 mm
- Different track cuts
  (10 MeV, 100 keV, 1 keV minimum energy)
- Standard LCIO output

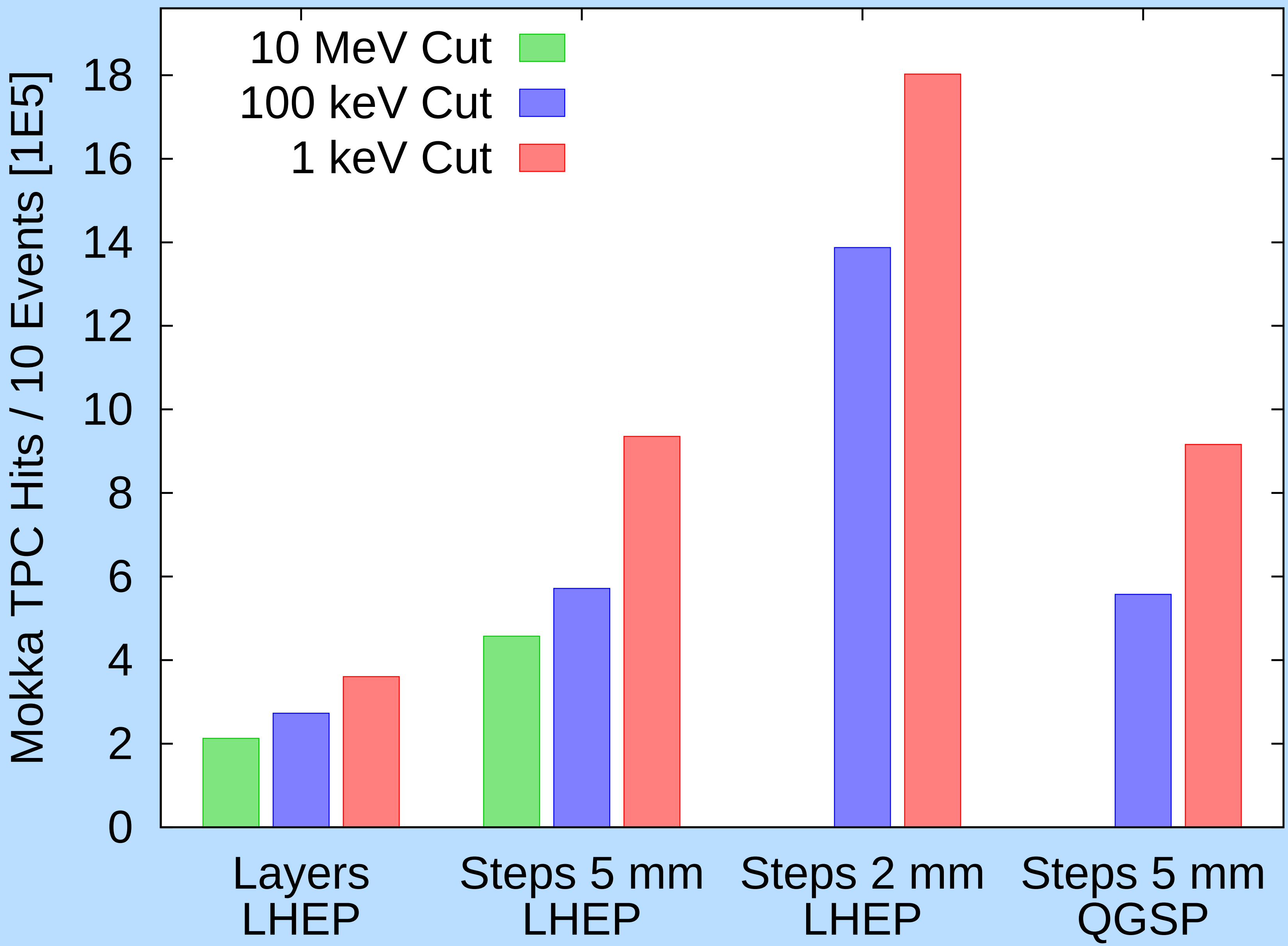
Comparison – Execution Time



Comparison – Output File Size



Comparison – Number of Hits



Tasks – Digitisation

Processes in the chamber

- clustering of the created charges along the track
- drift of the primary electrons, diffusion
- amplification (GEM collection, gain, extraction)
- “defocusing” between GEMs and the readout plane

Processes in the readout system

- collection of charges on the readout pads
- conversion to an electronics signal, pulse shaping
- introduction of electronics noise

Which of these do we really need / want?

Tasks – Reconstruction

Pulse finding

- where and when does the electronics signal contain significant information?

Track finding / pattern recognition

- which pulses belong together?

Track fitting

- row-based approach: determine one point per pad row
- pad-based approach: use the signal pattern as a whole

All this should be the same for simulated and real data!

Status – MarlinReco (ILC Soft)

Digitisation

- Mokka hits are taken as reconstructed points
- simple smearing to simulate detector resolution
- works only with the “gas layer” approach
- works only for row-based track fitting
- this is a radical shortcut!

Reconstruction

- recycled LEP tracking code
- wrapped Fortran routines from the last millennium
- used for all current activities (detector optimisation)

Status – TPCGEMSimulation (A. Münnich)

Digitisation (in the Marlin framework)

- microscopic description (single primary electrons)
- starts with MC particles, performs tracking itself
- HEED-like clustered energy deposition (parameterised)
- Garfield-like diffusion (parameterised)
- GEM behaviour from measurements (parameterised)
- charge collection on pads, pulse shaping
- interface to Mokka output needed

Reconstruction (stand-alone)

- only with ROOT, only for small prototypes (up to now)

Status – MultiFit (M. E. Janssen)

Digitisation

- not needed
- MultiFit runs on real data from a small prototype
- or uses a dedicated monolithic simulation (A. Imhof)

Reconstruction (stand-alone)

- modular structure
- different track fitting algorithms available
- used for ongoing resolution studies
- planned: porting to the Marlin framework (“MaTRIX”)

Status – UVic (C. Hansen, J. McGeachie)

Digitisation (within Mokka itself)

- creation of electron clouds for each step
- drifting, diffusion, amplification, defocusing
- charge collection on pads, pulse shaping
- planned: implemetation of field distortions
- deviates from the Mokka – Marlin paradigm

Reconstruction

- pad-based algorithms exist in JTPC (D. Karlen)
- planned: porting to Marlin / MarlinReco

Summary and Outlook – Status

Simulation

- TPC has always been the stepchild of Mokka
- more realistic descriptions are becoming available
- simulations become more complex (time and file size)
- but after all, the TPC *is* a powerful detector!

Digitisation and reconstruction

- no TPC-specific code in MarlinReco up to now
- new TPC models require dedicated digitisers
- various pieces of software are available,
  but they need to be connected properly

Summary and Outlook – Plans

General structure

- simulation and digitisation need to work together
- reconstruction should be independent (for real data)

Questions

- how detailed should the description be?
- which trade-offs are acceptable for other users? (Mokka is a simulation for the whole detector)

Various groups are working on the topic

- strengthen the world-wide cooperation
- let’s deliver some kick-ass software!