

Background Simulations for the ILC using Mokka

An Application of Geant 4

Adrian Vogel
DESY FLC

Background Sources at the ILC

e^+e^- pairs are a main source of background

- beams have to be focused very strongly ($\sigma_y = 5 \text{ nm}$)
- beam-beam interaction creates beamstrahlung
- beamstrahlung photons scatter to e^+e^- ($10^5 / \text{BX}$)
- e^+e^- smash into forward calorimeters (BeamCal) and magnets of the beam delivery / extraction line
- lots of photons, neutrons, and charged particles

Other sources are supposed to be negligible (beam dump, synchrotron radiation, . . .) or have to be studied in further detail (beam halo, extraction line losses)

Problems with Background

Inner silicon trackers (VXD, SIT, FTD)

- hits from charged particles (direct / indirect)
- silicon bulk damage from neutron fluence

Main gaseous tracker (TPC)

- Compton scattering, photon conversion
- neutron-proton collisions (recoil), neutron decay (?)
- additional primary ionisation, field distortions

Calorimeters (ECAL, HCAL)

- more photons from nuclear reactions, neutron capture
- random low-energy hits, radiation damage (?)

Simulation Tools – Guinea Pig (D. Schulte)

Input

- set of beam parameters ($E, \vec{\sigma}, \vec{\beta}, Q, \dots$)

Output

- particles in the disrupted beams
- beamstrahlung photons
- e^+e^- pair particles
- hadronic scattering products (“minijets”)

Existing simulation data

- TESLA beam parameters (500 GeV, 800 GeV)
- various ILC parameter sets (500 GeV, 1 TeV)

Simulation Tools – Mokka

Mokka is a full detector simulation

- based on the Geant 4 framework
- written in C++, modular design
- main development at LLR, France
- now: contributions from many different users
- successor of Brahms (GEANT3, Fortran)

Mokka uses LCIO as a persistency framework

- predefined storage classes (particle, track, hit, . . .)
- lightweight and robust, cross-platform design
- supported by large parts of the ILC community

Mokka – Detector Construction

- Subdetectors are represented by “drivers” (independent code modules)
- Actual geometry data is kept in MySQL databases
- No generalised geometry description: data in the database is meaningful only to one driver
- Different detectors available (LDC, SiD, testbeam)
- Need for flexibility: Different design options and absolute sizes selectable for detector optimisation
- Cooking a suitable geometry can be slightly convoluted

Mokka – Primary Generators

- Interface to `stdhep` and `HEPEvt` files
- Interface to Guinea Pig pairs
- Particle gun (with additional features)
- General particle source not included (so far)

Mokka – Physics Lists

- All built-in physics lists of Geant 4 available (but which one should be chosen?)
- `LCPhys` by Dennis Wright from SLAC
- `DummyPhysicsList` for visualisation and querying of geometry properties

Mokka – Analysis and Reconstruction

Mokka writes out “raw” hits

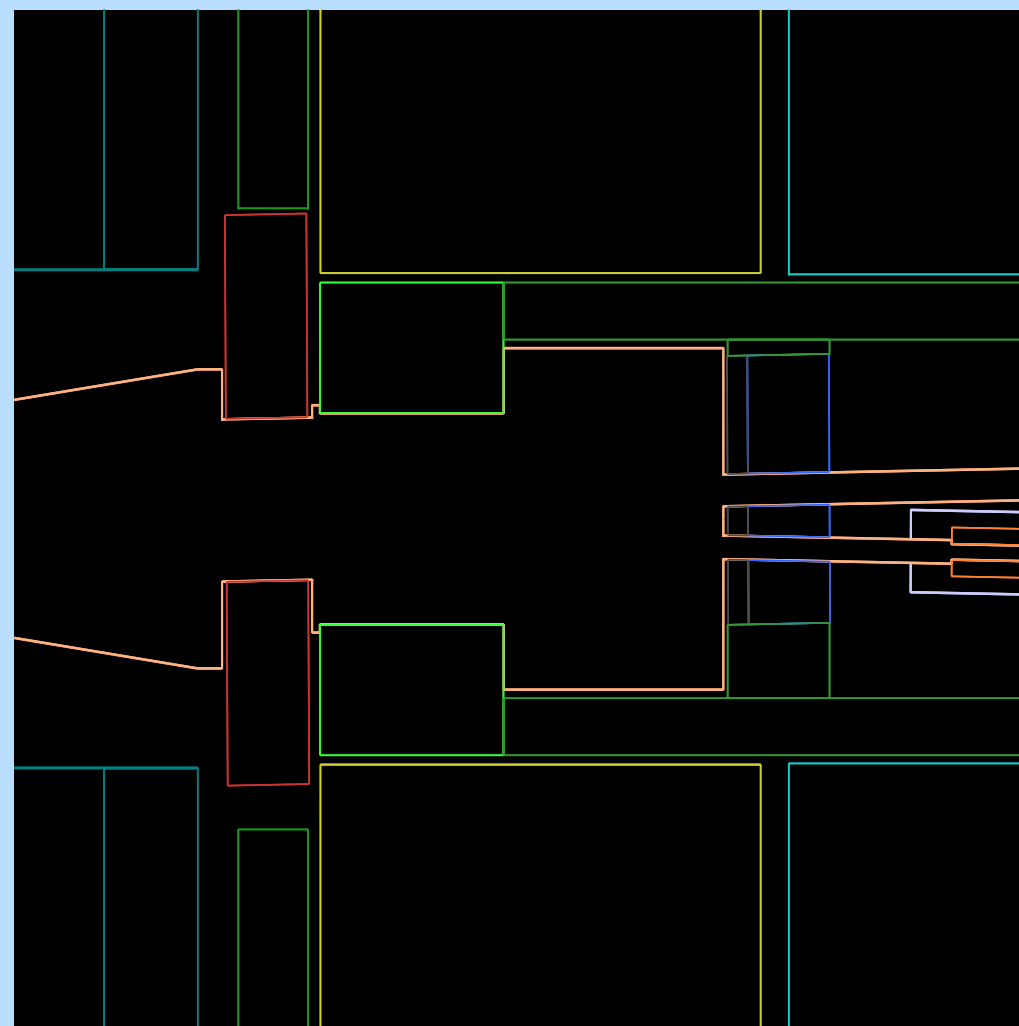
- idea: simulation takes much longer than digitisation, reconstruction, and analysis (presumably)
- write out simulation results as early as possible
- apply digitisation afterwards (with different settings)

Mokka output is processed by Marlin

- modular analysis and reconstruction framework
- set of processors read and write data collections from and to LCIO files
- also used for real data from prototypes

Geometry – LDC Forward Region

- LumiCal (red)
 $R_i = 120 \text{ mm}$
- Low- Z absorber
- BeamCal (blue)
 $R_{i1} = 15 \text{ mm}$
 $R_{i2} = 20 \text{ mm}$
- Centered on the downstream axis
- 20 mrad DID field

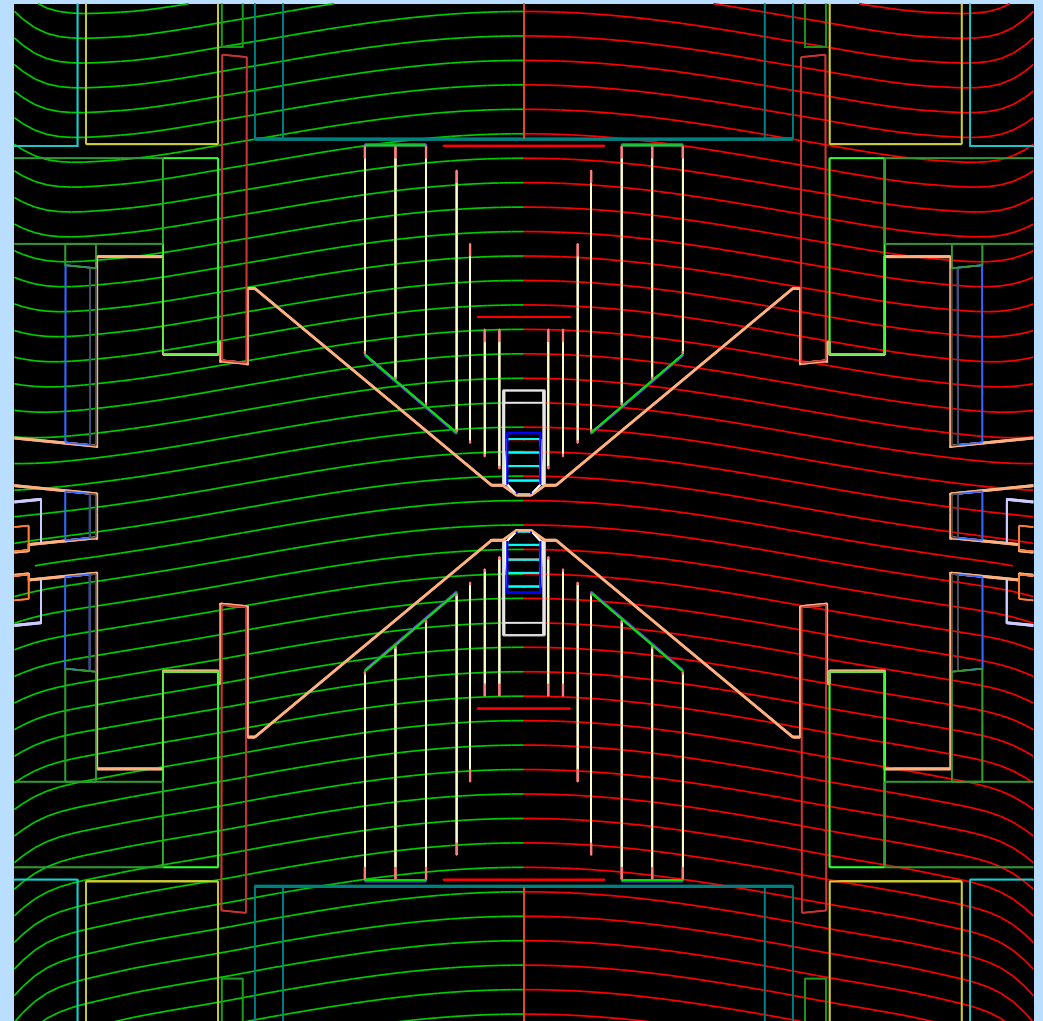


Compressed view 1:2

Geometry – Detector-Integrated Dipole

- Superimposed on the main solenoid field
- Introduced to prevent spin precession
- But has also a major impact on background

- Low-energy particles follow the field lines
- Tracks can be shifted in the x -direction



Compressed view 1:10

Geometry – 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

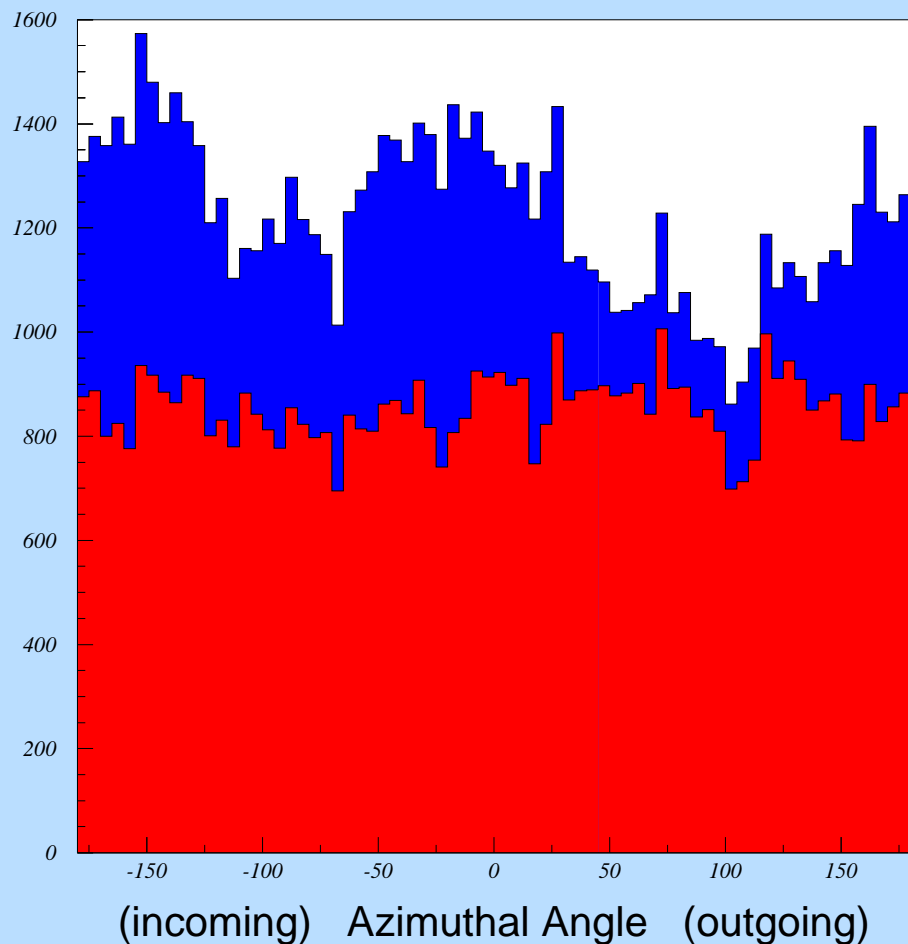
- large number of small energy deposits
- needed for tracking and dE/dx

Steps in the simulation need to be broken down

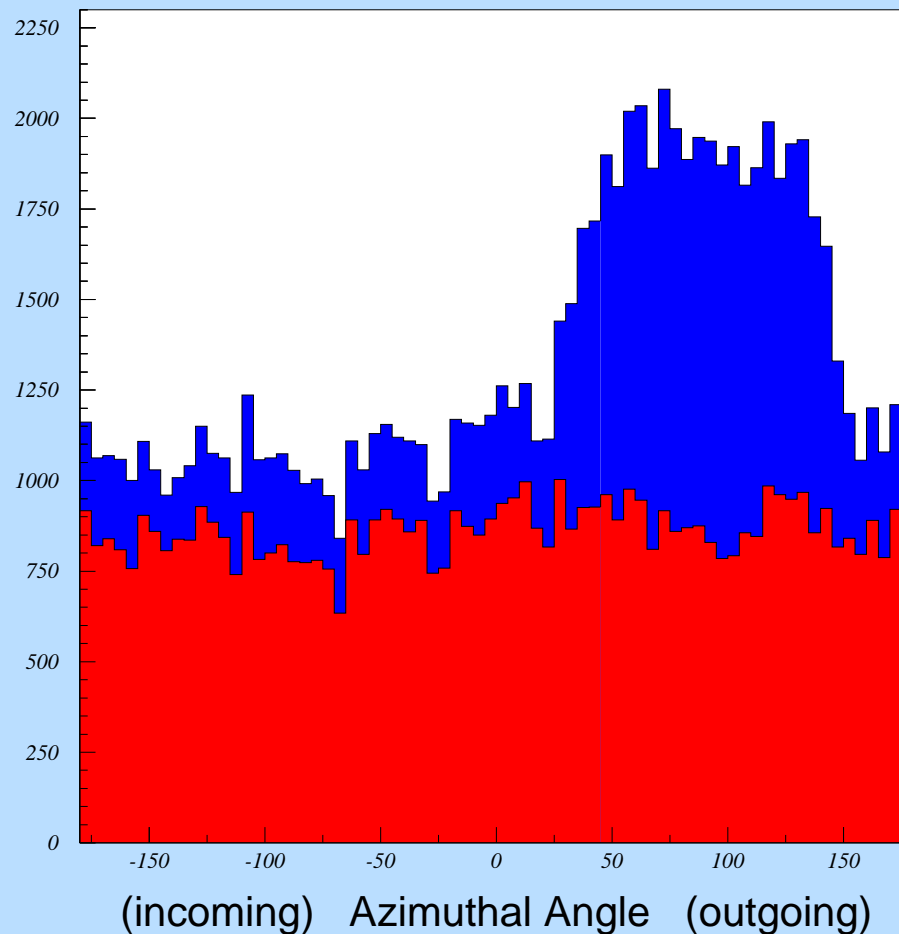
- introduction of artificial volume boundaries (before)
- limitation of the step length (now, needs digitiser)

Some Results – VTX Hits

Separation of immediate hits (red) and backscatterers (blue)



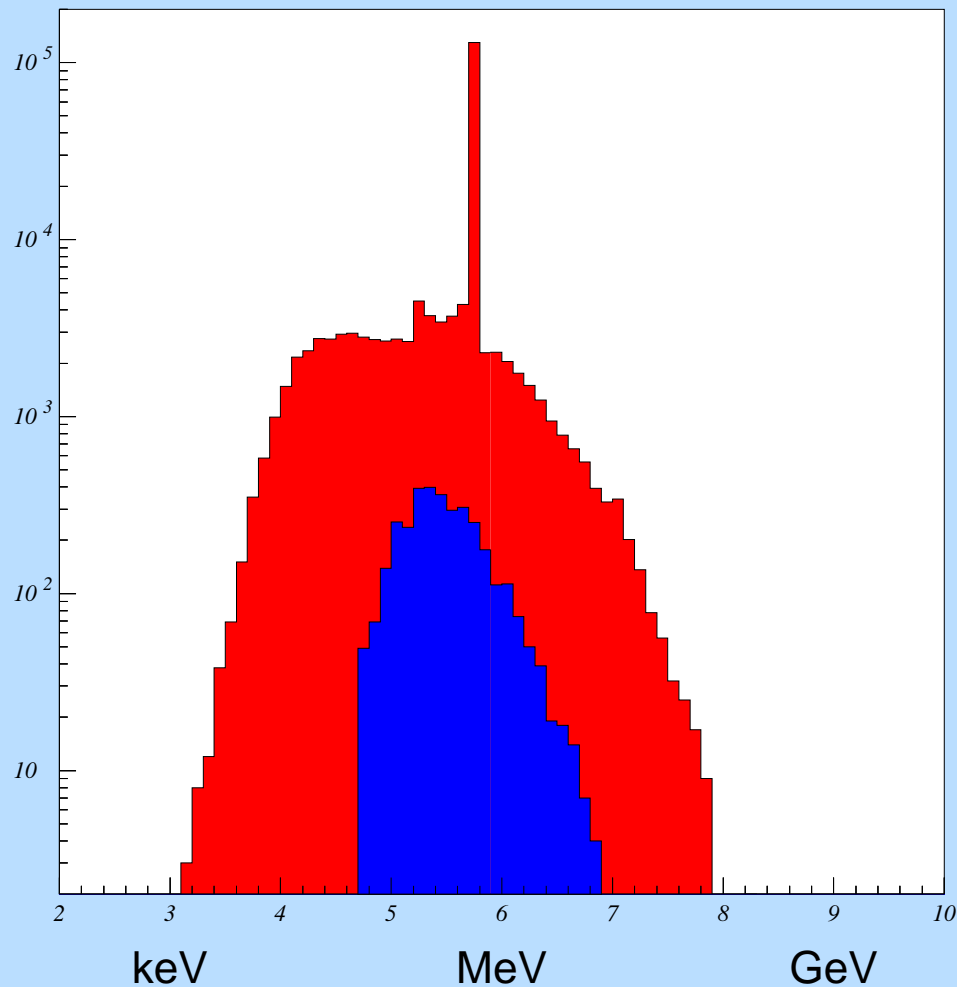
Layer 1, 2 mrad



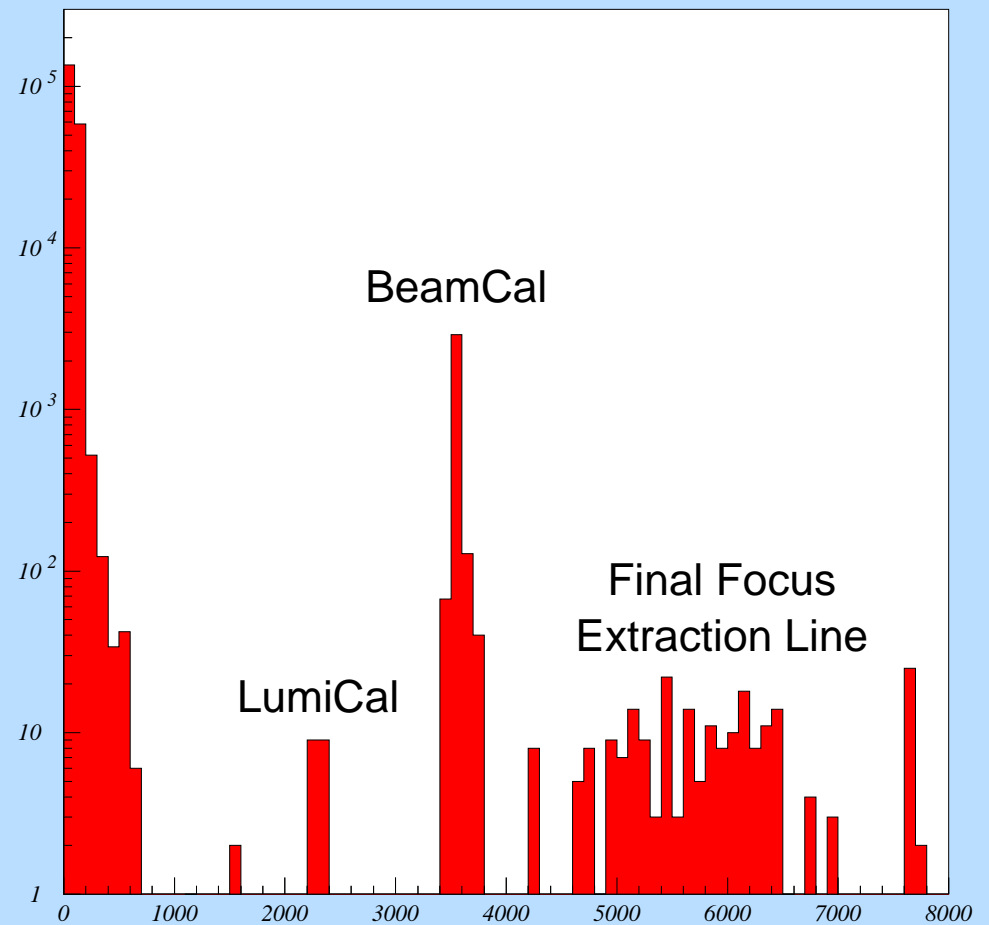
Layer 1, 20 mrad, DID

Some Results – Photons in the VTX

Photons crossing any of the five VTX layers



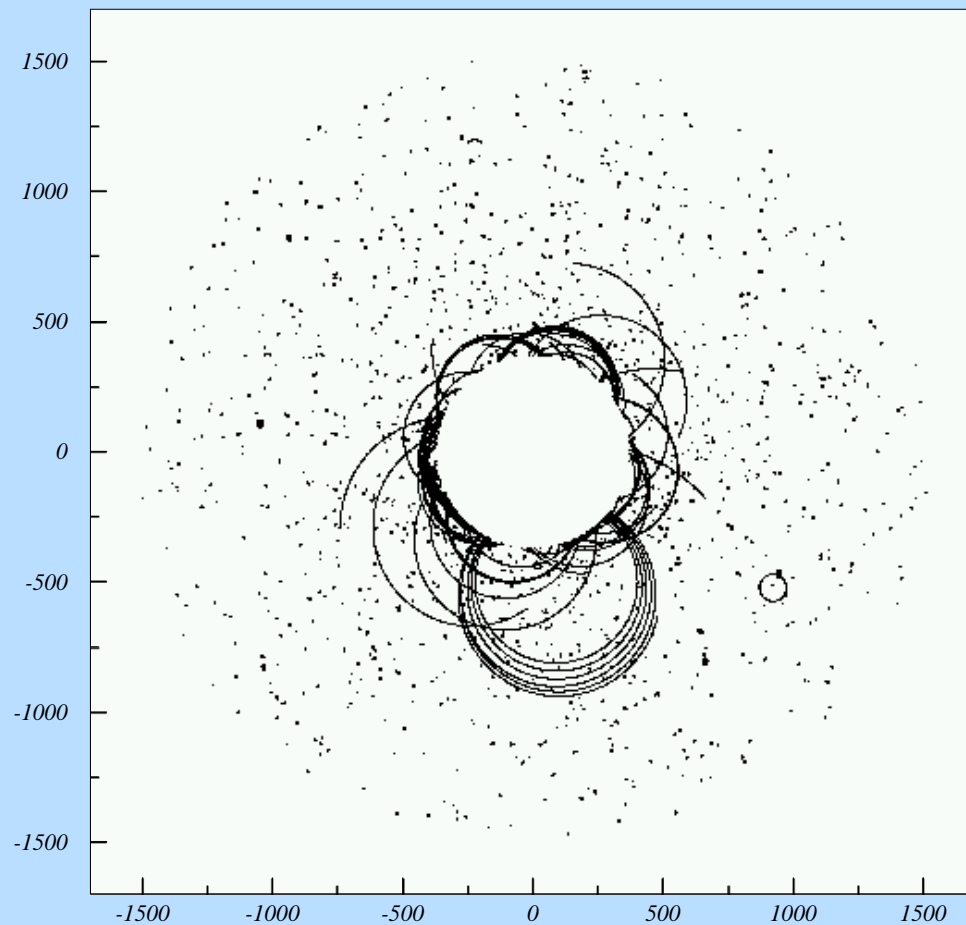
Energy spectrum



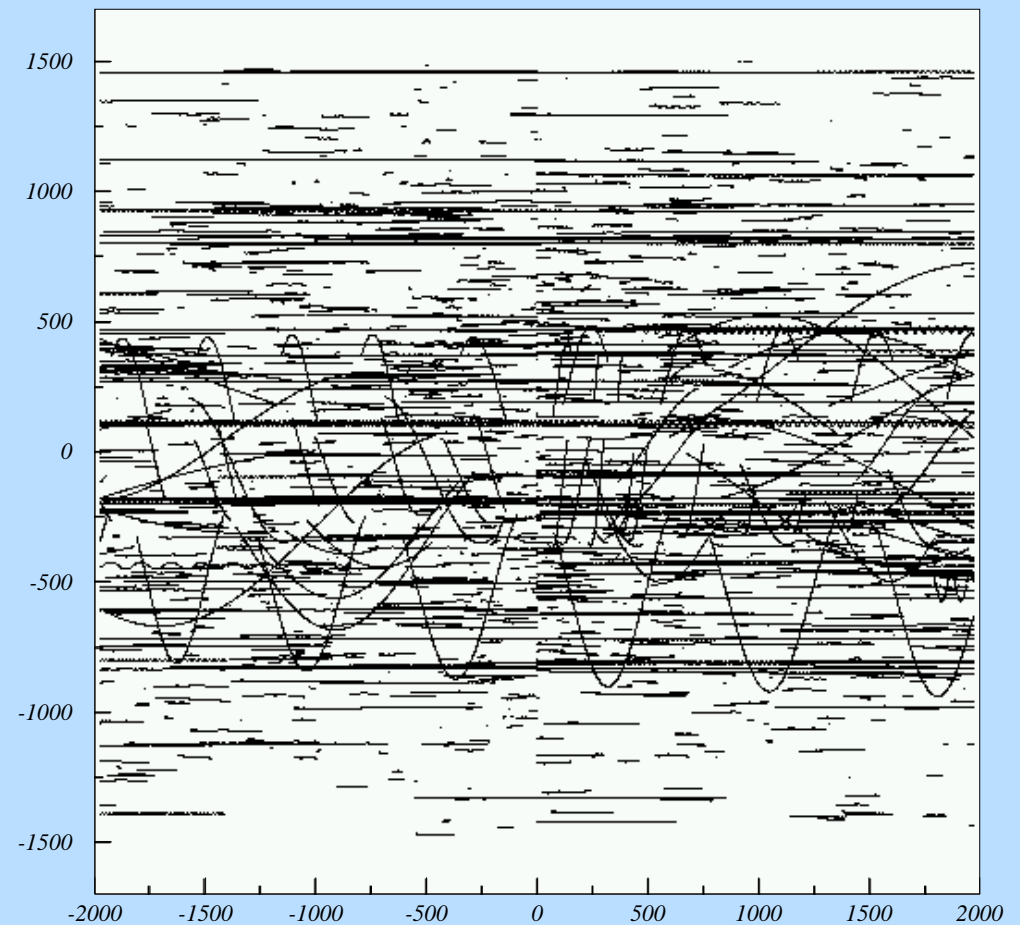
Origins, 2 mrad

Some Results – TPC Hits

Mokka hits in the TPC (overlay of 100 BX)

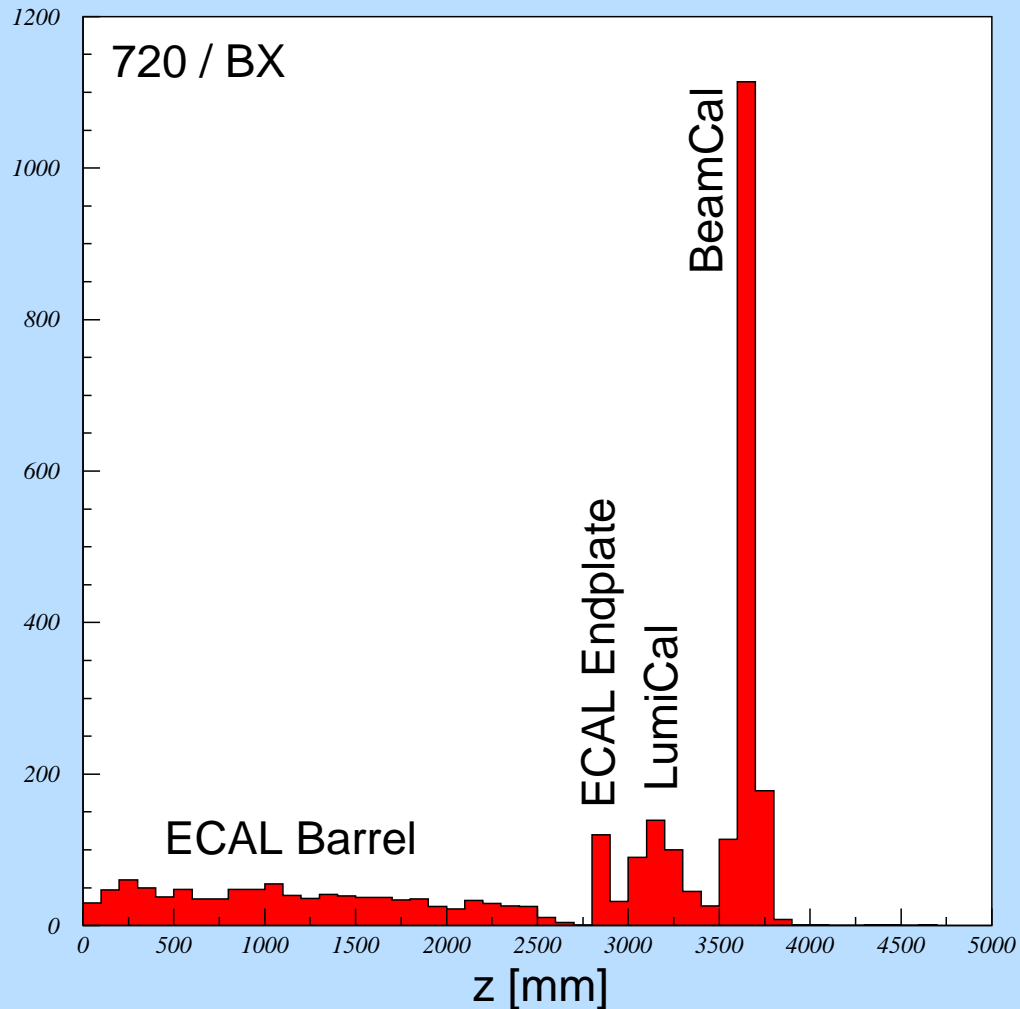


Front view, 2 mrad

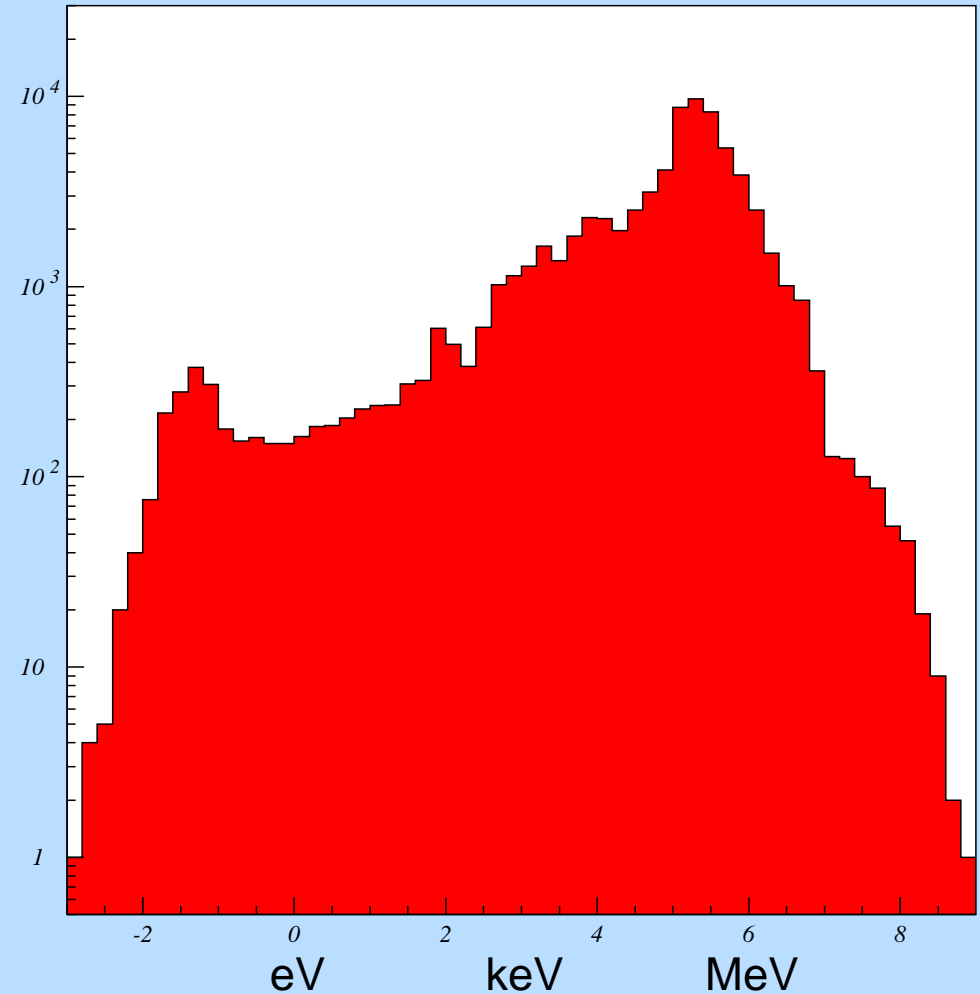


Side view, 2 mrad

Some Results – Neutrons in the TPC (old)



Origins of neutrons
reaching the TPC (Stahl)



Energies of neutrons
when entering the TPC

Mokka – To Be Done

- Improve geometry descriptions (incl. fields, materials)
- Optimise detector geometries (forward mask)
- Get better understanding of neutron models
- Determine and use proper simulation cuts
- Understand differences to Brahms results
- Build up a working software chain
simulation – digitisation – reconstruction
- Keep an eye on execution speed and file sizes
- Get results with high statistics (using the Grid)

TPC Simulations – Final Goals

Tasks to accomplish

- estimate all background tracks and the occupancy at a given time (with superposition of 160 BX)
- create a “background library” with ready-to-use events (e. g. similar to the “minimum bias events” of CMS)

Questions to answer

- how much background will the TPC be able to handle? (before its performance decreases significantly)
- which design decisions affect the TPC, and how?
- will a TPC be feasible as a main tracker at the ILC?