Simulation

Aims of CALICE data analysis

1) Measure the performance of the prototype calorimeters used in the beam tests
2) Compare Monte Carlo models with data to measure the degree of accuracy of the models
3) Apply knowledge gained from 2) to optimise the ILC detector calorimeters with a verified, realistic and trustworthy simulation
4) Develop calorimeter jet reconstruction algorithms and test them on real data as well as simulation

Simulation Model

- Geant4 as simulation framework
- Simulated output in LCIO format, directly comparable with data
- Support for multiple testbeam installations and whole detector models within common framework
- Support for wide range of physics models
- Accessible for grid production and individual users
- Models adaptable for systematic studies

Objectives

Aim 2) "Compare Monte Carlo models with data to measure the degree of accuracy of the models"

- Requires detailed description of all aspects of multiple testbeam installations
  - Physics models
  - Detector geometry/materials/placement
  - Beam profile
  - Digitisation

Aim 3), "Apply knowledge gained from 2) to optimise the ILC detector calorimeters with a verified, realistic and trustworthy simulation"

- Requires
  - The ILC detector concept models to be implemented to the same level of detail/accuracy which is found necessary to obtain acceptable level of agreement with testbeam data
  - Use of same physics models and parameter tunes
  - Prescription to attribute testbeam data-derived uncertainty to predictions of ILC detector concept studies

G4 Simulation – Mokka

- Fully detailed Geant4 application
  - Run driven by ascii input steering file (e.g. choice of detector model, etc.) and macro file (G4 native and Mokka implemented actions)
  - Wide range of Physics Lists available
    - ⇒OK, in that we can test any physics model that will be readily accessible for whole ILC detector concept studies
    - ⇒Not OK, if we need to access to Fluka (or MCNPX) physics - could be achieved for testbeam models, but difficult for general whole detector concept studies

Geometry definition

- Organised around single Mokka reference, MySQL relational database
- Detector models, composed of sub-detectors, each implemented by driver code, with parameters (physical dimensions, repeat counts, …) extracted from relevant tables in Mokka database at runtime, e.g.

Model

name
D08

Ingredients

model	sub_detector
D08	ecal02
D08	hcal02
D08	mask00
D08	field01
D08	tpc00
D08	yoke00
D08	coil00

Sub-detectors

name	db	driver
ecal02	ecal02	ecal02
hcal02	hcal02	hcal02
mask00	mask00	mask00
field01	field01	field00
tpc00	tpc00	tpc00
coil00	coil00	coil00
yoke00	yoke00	yoke00

Mokka Detector Model Database Browser - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www-flc.desy.de/ldcoptimization/tools/mokkamodels.php?model=CaliceEcal03#driver_proto03

Mokka Detector Model Database Browser

CaliceEcal03 [Select]

Detector Model "CaliceEcal03"

Description	The Calice Ecal prototype
Status	frozen

Detector Concept "Test Beam Calice"

Description	Test beam with Ecal and Hcal prototypes
World Box	$10000 \times 10000 \times 10000$ mm³ (octant)
Tracker Region	$r < 0$ mm, $\lvert z \rvert < 0$ mm
Calo Region	$r < 0$ mm, $\lvert z \rvert < 0$ mm

Subdetector "Proto03"

Description	Calice Ecal prototype
C++ Driver	`proto03`
MySQL Database	`CaliceEcal03`
Parameters	`configuration_angle`, `use_tracker`

Parameter "configuration_angle"

Description	Test Beam configuration angle for the Calice test beam.
Value	0.0 (default) – may change at runtime!
Setups	`TB00` (Calice Test Beam with setup angle of 0 degrees) with value 0.0
	`TB10` (Calice Test Beam with setup angle of 10 degrees) with value 10.0
	`TB20` (Calice Test Beam with setup angle of 20 degrees) with value 20.0
	`TB30` (Calice Test Beam with setup angle of 30 degrees) with value 30.0
	`TB40` (Calice Test Beam with setup angle of 40 degrees) with value 40.0
	`TB45` (Calice Test Beam with setup angle of 45 degrees) with value 45.0
Drivers	`proto03`

Parameter "use_tracker"

Description	use tracker in front of proto03 to save momentum at the entrance of proto
Value	false (default) – may change at runtime!
Drivers	`proto03`

Caveat

Keep in mind that further parameters may be created at runtime.

Database `models03` on `pollin1.in2p3.fr` – 2007-12-17, 17:27 – Adrian Vogel, DESY FLC

Done

Mokka Detector Model Database Browser - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www-flc.desy.de/ldcoptimization/tools/mokkamodels.php?model=TBCern0807

Mokka Detector Model Database Browser

TBCern0807 [Select]

TBCern0707
TBCern0806
TBCern0806_01
TBCern0806_01_dchxy_new
TBCern0806_dchxy_new
TBCern0807
TBCern1006
TBCern1006_01
TBCern1006_01_dchxy_new
TBCern1006_dchxy_new
TBDesy0205
TBDesy0205_2
TBDesy0506
TBDesy0506_01
TBDesy0506_01_dchxy
TBDesy0506_01_dchxy_new
TBDesy0506_dchxy
TBDesy0506_dchxy_new
TDR
TM06

Detec

…am model implementation for the Cern August 2007 setup with a partly instrumented Ecal: bottom front 3 slabs replaced with W

…e

Detec… …Cern"

…am setup at Cern August 2006

…× 60000 × 60000 mm³ (octant)

…m, |z| < 0 mm

…m, |z| < 0 mm

Subd…

…entation of the fully instrumented Ecal for the Cern 2007 setup

C++ Driver	`proto05`
MySQL Database	ProtoCern07
Parameters	Ecal_slab_pattern, EcalTranslateY, EcalRotationAngle, EcalTranslateX, shift_module1, shift_module3, use_tracker, configuration_angle

Subdetector "TBcatcher07"

Description	Implementation by Guilherme Lima of the TCMT used at Cern in 2007
C++ Driver	`TBcatcher06`
MySQL Database	TBhcalcatch07
Parameters	Tcmt_layer_pattern, TCMTRotationAngle, TCMTTranslateX, TCMTTranslateY

Subdetector "TBcerenkov03"

Description	Implementation by Fabrizio Salvatore of the Cerenkov detector used at Cern in 2007
C++ Driver	`TBcerenkov02`
MySQL Database	tcherenkov02
Parameters	cherenPressure, cherenGasType

Subdetector "TBdch05"

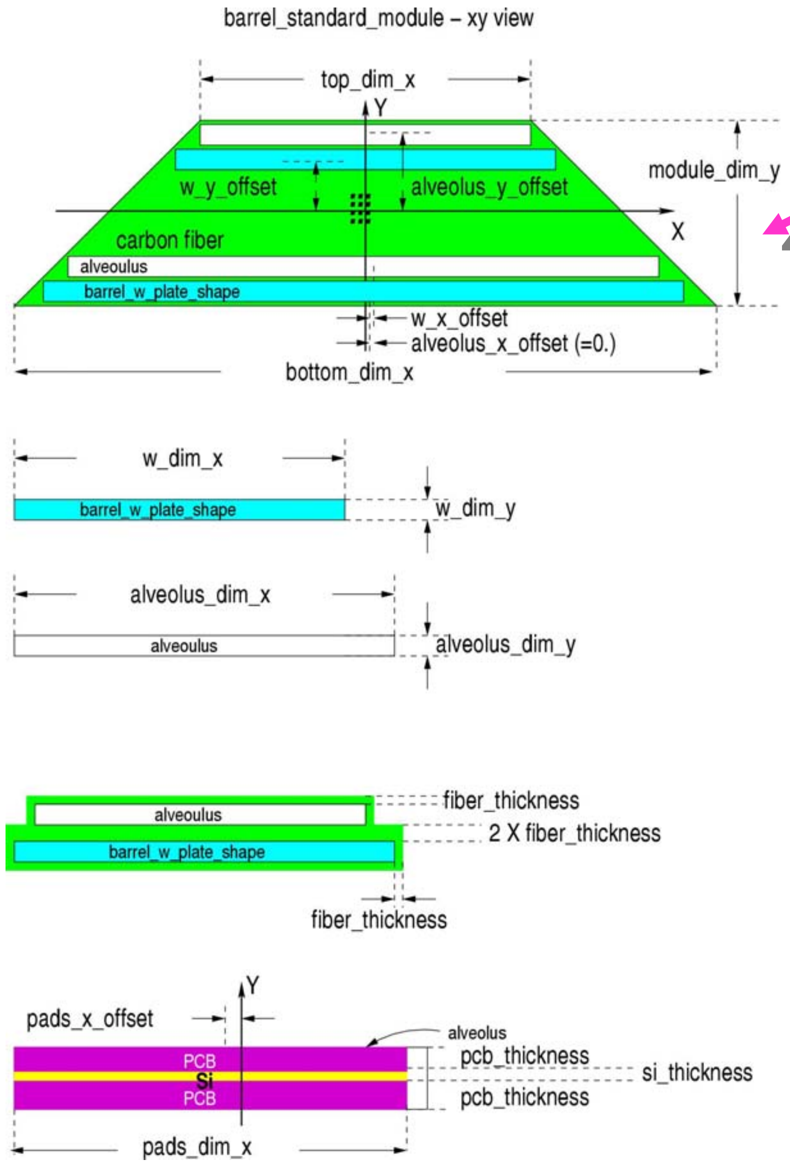
Description	Implementation by Fabrizio Salvatore of the drift chambers used at Cern in 2007
C++ Driver	`TBdch04`
MySQL Database	dch04

Subdetector "TBhcal07"

Description	HCal with virtual cells and rotation/staggering for the CALICE Test Beams in 2007 at CERN, i.e. the models TBCern07, TBCern0707, TBCern0807
C++ Driver	`TBhcal06`

Done

Geometry definition – and reconstruction



- Mokka uses regular geometry definition using symmetry/replication
- What about imperfections in real testbeam data?
  - Relative sub-detector alignment and internal geometry
  - Necessary to simulate this? – if so, where is geometry defined and how does reco find it?
    - ⇒ Calice conditions database - but Mokka not just used by Calice!
    - ⇒ Mokka database – duplicates reference source of data in Calice database!
  - Not a new problem…

Geometry options

- Copy Mokka geometric data into Calice conditions db, use same reconstruction code for data/MC, analyse data and put improved position estimates in Mokka db
- Use GEAR to access geometry in reco, but intended to be ~general, so not ideal to have much experiment-specific code
- Use Mokka conditions db directly to generate Mokka geometry
  - Ideal way, geometry data more easily under Calice control, ensure same used for simulation and reco
  - Most effort intensive, requires drivers to be re-written
  - Implies regenerating MC when new geometrical constants evaluated
- Changes relatively infrequently, few times/year

Beam profile

- Mokka accepts inputs from
  - Particle gun, with independent smearing of position/momentum
  - Stdhep/hepevt
- Option to use BDSIM for beamline modelling, integrated to Geant4
  - Incomplete data from CERN/AB
  - BDSIM expertise/effort not so forthcoming
- Pragmatic solution: use data driven parametrisation of position, generate stdhep file
  - Caveat: requires extrapolation to -25m, through materal upstream of instruments.

- Harder for user to generate small samples of events for personal debugging/study, but….
  - Running the reco is still a relatively an expert task anyway

- Changes for every run – book keeping problem?!
  - Run-specific parameteters extracted from Calice conditions db, into steering files/macro files to drive run
  - Run number to simulate, and database tag immediting prior to MC production written to runHeader
  - Unpacked by reco processors to apply appropriate run-specific treatment (noise, etc.) as necessary.

Comparison with Simulation Model

- Geant4 as simulation framework
  - OK
- Simulated output in LCIO format, directly comparable with data
  - OK, compatibility with data
- Support for wide range of physics models
  - Built into G4 framework
  - No good solution for Fluka
- Support for multiple testbeam installations and whole detector models within common framework
  - OK for testbeams; plausibly OK for ILD detector concept models, but SiD uses different g4 application (slic)
  - Potentially significant problem for Calice results to be used by both ILD and SiD
    - ⇒ Connection to data results at higher/more astract level for SiD?
- Accessible for grid production and individual users
  - OK for production
  - Single users encounter more difficulty when "run-by-run" simulation implementation of real conditions
- Models adaptable for systematic studies
  - Difficult to adapt for non-expert/author
  - Not as flexible e.g. as slic

Conclusions

Aim 2) "Compare Monte Carlo models with data to measure the degree of accuracy of the models"

- Requires detailed description of all aspects of multiple testbeam installations
  - Physics models
  - Detector geometry/materials/placement
  - Beam profile
  - Digitisation

Aim 3), "Apply knowledge gained from 2) to optimise the ILC detector calorimeters with a verified, realistic and trustworthy simulation"

- Requires
  - The ILC detector concept models to be implemented to the same level of detail/accuracy which is found necessary to obtain acceptable level of agreement with testbeam data
  - Use of same physics models and parameter tunes
  - Prescription to attribute testbeam data-derived uncertainty to predictions of ILC detector concept studies

Backup slides

A detailed description of the TB06 models

Detector description: TBDesy0506 I

- Drift Chambers (FS):
  - installed by Kobe collaborators for the 05 test beam
  - gas mixture is non-flammable (96% Ar, 4% Ethane)
    - 4 drift chambers (72x72x88 mm^3)
    - hits written out in LCIO format
      - To reduce number of hits, only hits with E_{rel} > 0.001 are written in output
- Trigger scintillators (FS):
  - 3 scintillators (one 120x120x8 mm^3, two 200x200x8 mm^3) used in the trigger
  - hits written out in LCIO format
    - Hits simulated as Calorimeter hits (one hit per chamber)

Detector description: TBDesy0506 II

- Finger counters (FS):
  - 2 scintillators (5x100x5 mm^3) placed in T shape to monitor beam position
  - hits written out in LCIO format
    - Hits simulated as Calorimeter hits (one hit per chamber)
- ECAL (G.Musat):
  - 3 modules (5 slabs)
    - tungsten thicknesses = 1.4, 2.8, and 4.2 mm.
    - silicon planes divided into wafers
      - 6x6 cells (10x10 mm^2), guard-rings (1 mm width)
      - Two separate hits collections, one for hits in cells and the other for hits in guard-rings

Detector description: TBCern0806 I

- Cerenkov detector (FS):
  - It is upstream of the first trigger scintillator (~25 m)
    - 100x100x11000 mm^3, 180μ mylar windows, helium gas
    - Only the material is simulated
- Drift Chambers (FS):
  - provided by CERN (50% Ar, 50% CO_2)
    - 3 drift chambers (108x108x44 mm^3)
    - hits written out in LCIO format
      - To reduce number of hits, only hits with $E_{rel} > 0.001$ are written in output
- Trigger scintillators (FS):
  - 3 scintillators used in the trigger (one 30x30x15 mm^3, two 100x100x15 mm^3)
  - One veto scintillator (200x200x15 mm^3)
  - hits written out in LCIO format
    - Hits simulated as Calorimeter hits

Detector description: TBCern0806 II

- ECAL (G.Musat):
  - same as for TBDesy0506
- HCAL (R.Poeschl, O.Wendt):
  - 39 layers (900x900x30 mm^3). Each layer is composed by an iron absorber and scintillating material and is sub-divided into 90x90 mm^2 cells of 10x10mm^2 (virtual cell scheme)
    - Cell numbering scheme (from lower left corner of each layer)
      - i = row, j = column, k = layer.
- TailCatcher (J.McCormick, G.Lima):
  - 16 layers (absorber+air+readout module)
    - 2 different absorber thicknesses (19 mm - layers 1 to 8, 101 mm – layers 9 to 16).
      Readout modules: 9.5 mm. X,Y dimensions: 1168x1168 mm^2
    - All absorbers in place, but only 8 readout modules (1, 4, 7, 10 – vertical strips, 2, 5, 8, 11 – horizontal strips)
- Muon Counters (FS):
  - 2 scintillators (1000x1000x50mm^3)
  - hits written out in LCIO format
    - Hits simulated as Calorimeter hits

Detector description: TBCern1006

- Cerenkov detector (FS):
  - same as TBCern0806
- Drift Chambers (FS):
  - same as TBCern0806
- Trigger scintillators (FS):
  - same as TBCern0806
- ECAL (G.Musat):
  - same as TBCern0806
- HCAL (R.Poeschl, O.Wendt):
  - Only 30 layers, with same characteristic as TBCern0806
- TailCatcher (J.McCormick, G.Lima):
  - Same as TBCern0806, but with all layers fully instrumented
- Muon Counters (FS):
  - same as TBCern0806