# LCPolMC
# Linear Collider Polarimeter Simulation

Jenny List

DESY

13. April 2011

# LCPolMC Overview (1)

- generate Compton events according to laser and beam parameters
- track scattered $e^{\pm}$ through chicane to detector front face
- calculates analysing power
- simulates the detector response
- calculates the measured polarisation and statistical errors
- allows to introduce misalignments, non-linearities, ....

typical use case:
simulate 100k BX ( 2000 $e^{-}$/BX!) with several non-perfect values of a detector parameter and study effect on measured polarisation

# LCPolMC Overview (2)

- past: was main working horse against ideas like scaling the B-field of the chicane with $E_{\mathrm{beam}}$
- since: a bit dormant since main focus was on detector prototype and testbeam data.
- now:
  - incorporate testbeam results
  - interface to spin tracking (BMAD, Moritz)
  - implement 6-magnet chicane a la downstream
  - $\Rightarrow$ revisit optimisation of the whole systems (laser, chicane, detector)

# Interface to Spin Tracking

- up to now: LCPolMC works with $\delta$-distribution of polarisation and beam energy
- BMAD: delivers collection of macro-particles with individual 4-momenta and spin vectors
- $\Rightarrow$ Compton generator has to be upgraded to cope with different $E_{\mathrm{beam}}$ and $\vec{P}$
- find an efficient way to define the phase space to be sampled during event generation
- work in progress, but not highest priority right now ....

# Incorporation of testbeam results

**NEW** in the last months

- svn repository on FLC afs pool: `/afs/desy.de/group/flc/pool/polarimeter/svn/lcpolmc`
- sharing with non-DESY users still requires tar-ball $\longrightarrow$ is improvement needed?
- introduced simple (= completely blind) channel walls $\longrightarrow$ introduce real response function observed in testbeam
- improved non-linearity modelling according to results from Christian's thesis
- improved digitisation modelling according QDC used in testbeam
- started to clean up...

# Current detector simulation (1)

- `GasDetector::readQE(filename)`:
  calculate expectation value $\kappa$ for the number of detected photoelectrons per primary electron
  (integral over $dN^{\gamma}/d\lambda(\lambda) \times \mathrm{QE}(\lambda) \times$ overall efficiency)
- `main`:
  for each BX and for each channel:
- `Detector::sortE(x,y)`:
  determine number of Compton electrons $N_{\mathrm{Compton}}$ in this channel
- `GasDetector::fastdetect(ichan, ilr, ibx)`:
  throw actual number of photo-electrons poisson-distributed around $N_{\mathrm{Compton}} \cdot \kappa$

# Current detector simulation (2)

- `Detector::toADC(ichan)`: optionally: add darkcounts
- `Detector::toADC(ichan)`: apply gain, convert to charge
- `Detector::toADC(ichan)`:

  optionally: apply PMT non-linearity
- `Detector::toADC(ichan)`:
  optionally: apply gaussian smearing (noise)
- `Detector::toADC(ichan)`:
  digitise to QCD counts (optionally two ranges)

BUT: hard to find out if you look at the code - it is a mess: badly chosen abstractions, obsolete methods, ... :-(

# New design - Detector part

get rid of confusion between methods implemented in `Detector` or `GasDetector`:

- *one* detector class holds objects of (base) type
  - `Geometry`: channel geometry, possibly including misalignments....
  - `Medium`: properties of the Cherenkov medium
  - `PMT`: characteristics of PMT (QE, dark rate, gain, non-linearity, segmentation, ....)
  - `Amplifier`: additional amplification if needed
  - `QDC`: QDC resolution, noise, ...
- handles information flow between the detection steps (currently in main program!)
- needs a quiet day to be implemented...

# New design - Analysis part

- AP calculation, detector simulation and reconstruction are tied closely together
  - hard to understand, even harder to extend...
  - AP calcualtion and simulation need a lot common functionality
  - introduce persistency between sim & rec?
  - $\Rightarrow$ started to think, but no complete new design yet
- get rid of cernlib?
  - HBOOK output option probably used only by me ;-)
  - would need replacement of random number generation etc
  - core of Compton generator is also FORTRAN, but no cernlib

# Summary

- LCPolMC reactivated
- interfacing to BMAD ongoing
- first steps to incorporate testbeam results
- major redesign of the code needed and underway
- ...but there is and will always be a usable version!