MAUS Status

A. Dobbs

CM43

29th October 2015

Contents

- MAUS Overview
- Infrastructure
- Geometry and CDB
- Detector Updates
  - CKOV
  - EMR
  - KL
  - TOF
  - Tracker
- Global Tracking
- Deployment
- Next Steps

MAUS Overview

- Mice Analysis User Software
- Provides simulation and reconstruction of the MICE beamline and detectors
- Map – Reduce based framework (Python wrapping C++)
- GEANT4 based simulation
- G4BeamLine upstream beam simulation
- Data persistency and analysis from ROOT
- Online data reconstruction
- Distributed via Bazaar and Launchpad, as well as tarballs of releases
- Website: http://micewww.pp.rl.ac.uk/projects/maus/wiki

Infrastructure

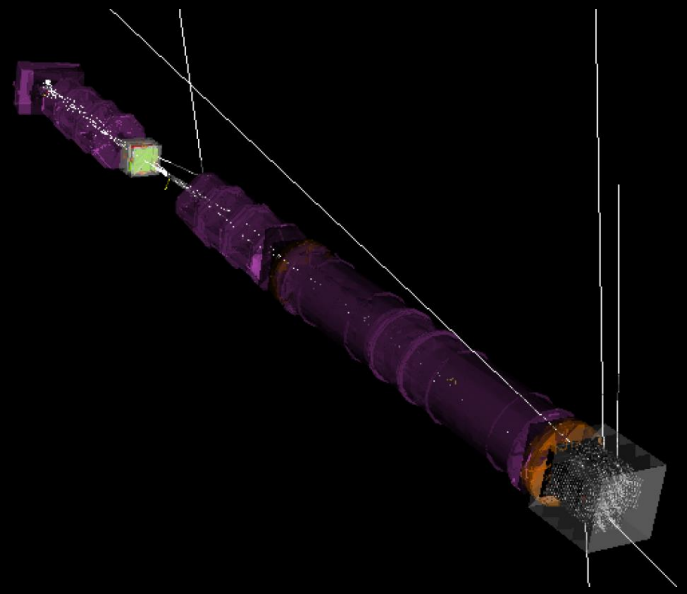
- API has now been completed (including for Reducers)
- Only MAUS::Data* objects passed between modules (no string conversions)

➡ factor **60** or so speed up! Data processed faster than can be taken.

- Memory leaks remain, but no longer stopping GRID running
- New online reconstruction framework currently being tested

Geometry and CDB

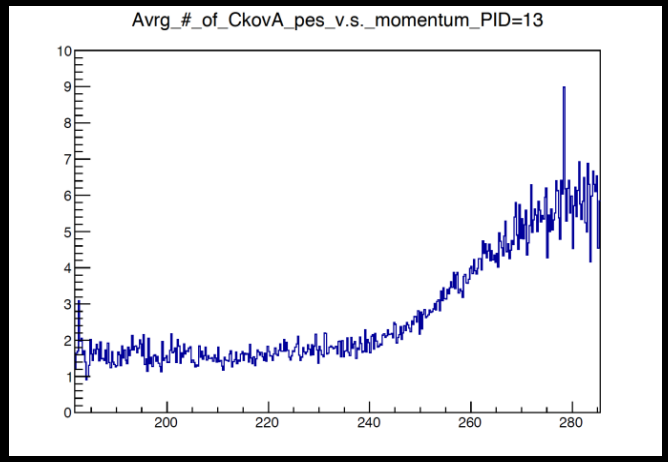
- CDB geometry has undergone a lot of improvements (TOF, KL, Tracker…), almost ready to replace the legacy as the MAUS default
- Tracker configuration files now accessible from CDB
- MAUS CDB interface has been updated
- Almost ready to restart GRID MC running

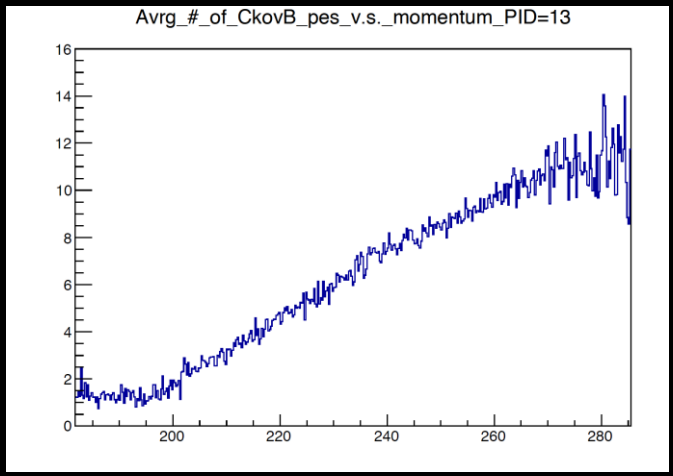


Detectors: CKOV

- Rapid progress thanks to new team member Ao Liu
- Previously missing MC Digitiser now in place
- Geometry implementation updated
- Real data reconstruction producing results

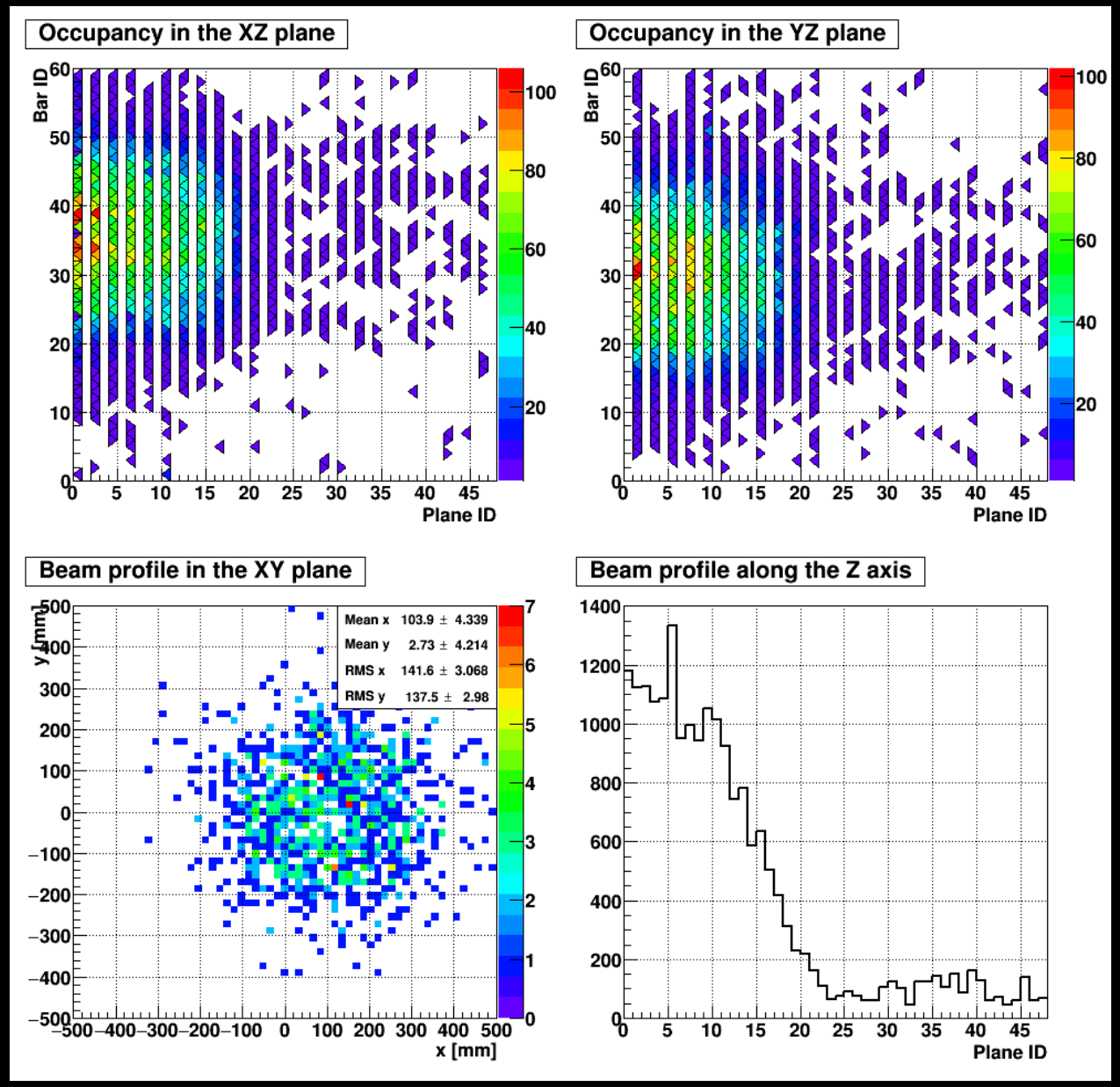
Offline analysis of recent runs





Detectors: EMR

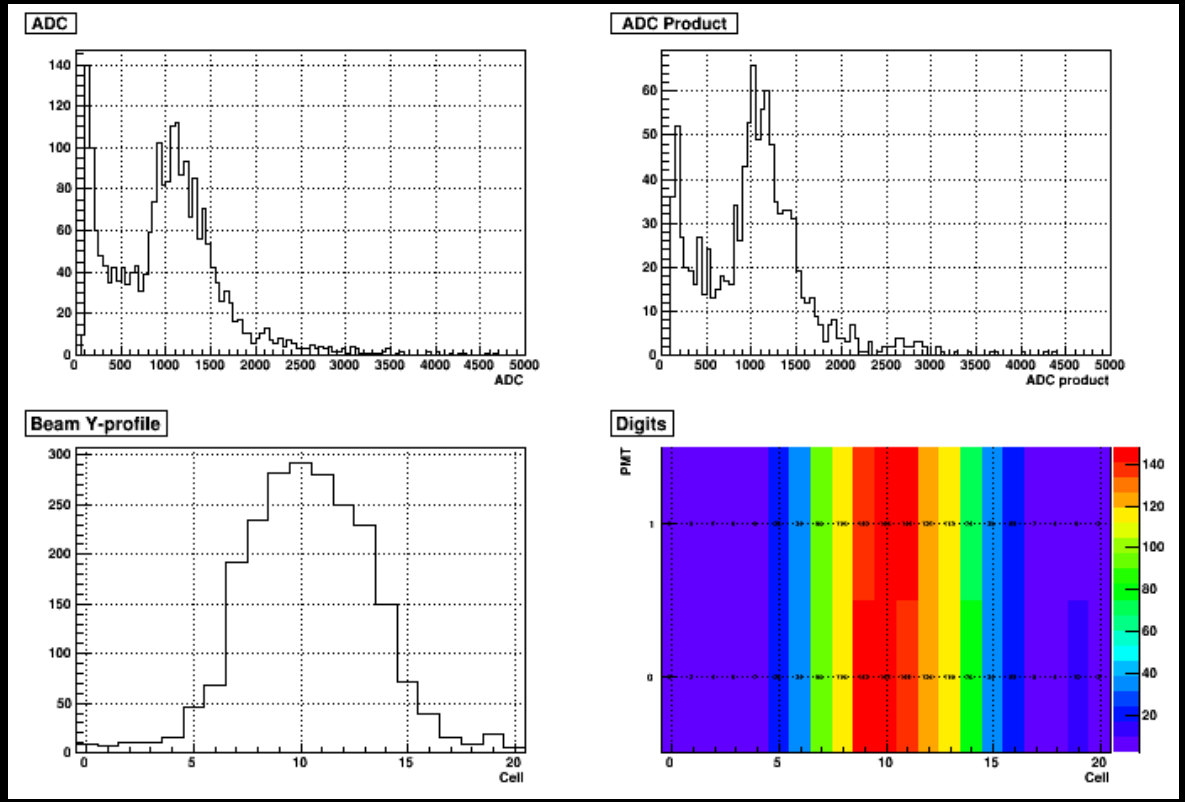
- Position and PID variable code completed – supplies coordinates for hits, muon tracks and decay products, charge deposition and various other PID variables
- Interface to CDB to access calibration and geometry completed
- Reducer for Online Reconstruction added
- Only a few jobs left:
  - New calibration file for single anode PMTs
  - Data quality flag
  - Documentation


Occupancy in the XZ plane
Bar ID
Plane ID
Occupancy in the YZ plane
Bar ID
Plane ID
Beam profile in the XY plane
Mean x 103.9 ± 4.339
Mean y 2.73 ± 4.214
RMS x 141.6 ± 3.068
RMS y 137.5 ± 2.98
y [mm]
x [mm]
Beam profile along the Z axis
Plane ID

Detectors: KL

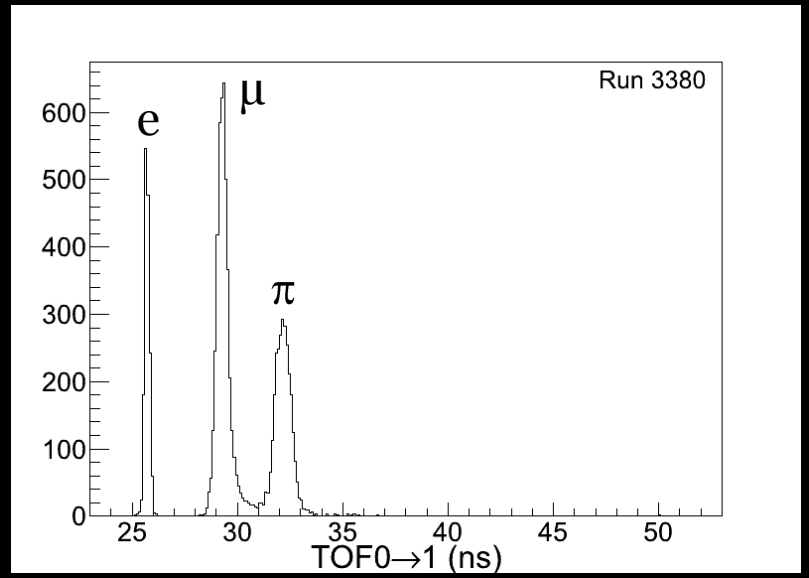
- Functional, stable code base
- Reducer in place
- Global coordinate system output in place
- Data quality flag in place

Online Reconstruction plot from recent running



Detectors: TOF

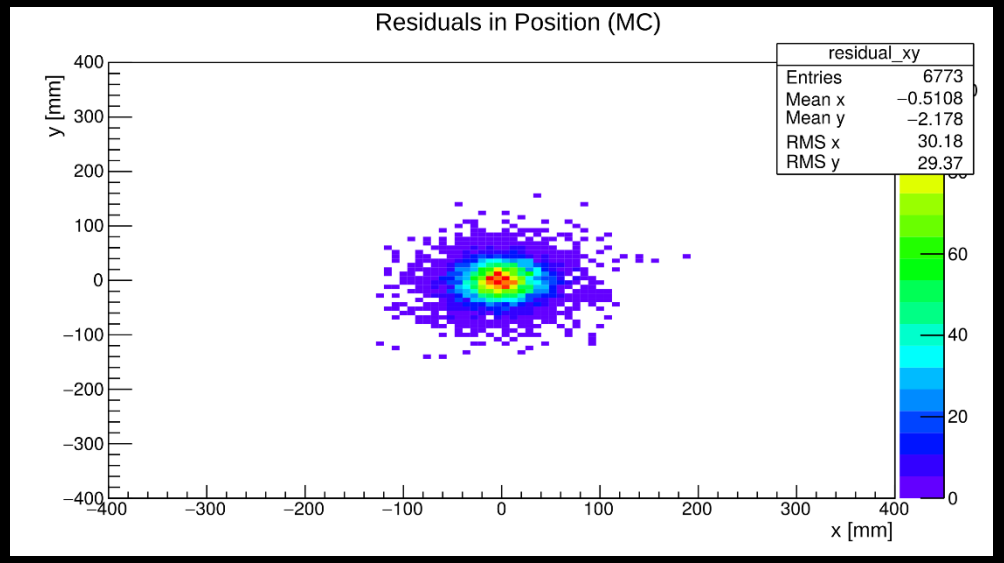
- TOF code stable – longest standing functional detector code
- Geometry bug recently found and fixed (showed up on comparison with tracker)
- Global coordinate system output added
- MC trigger issue remains (needs a person…)

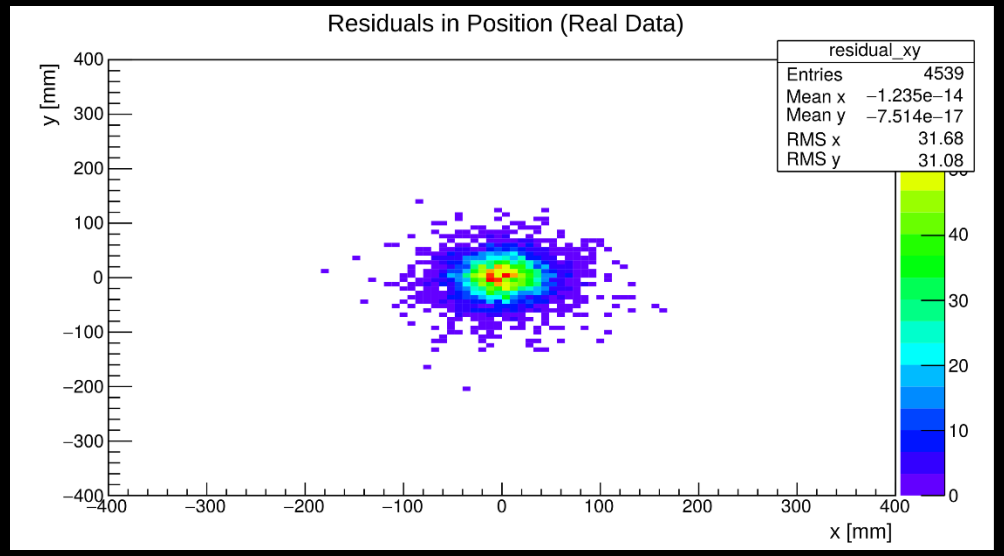


Detectors: Tracker

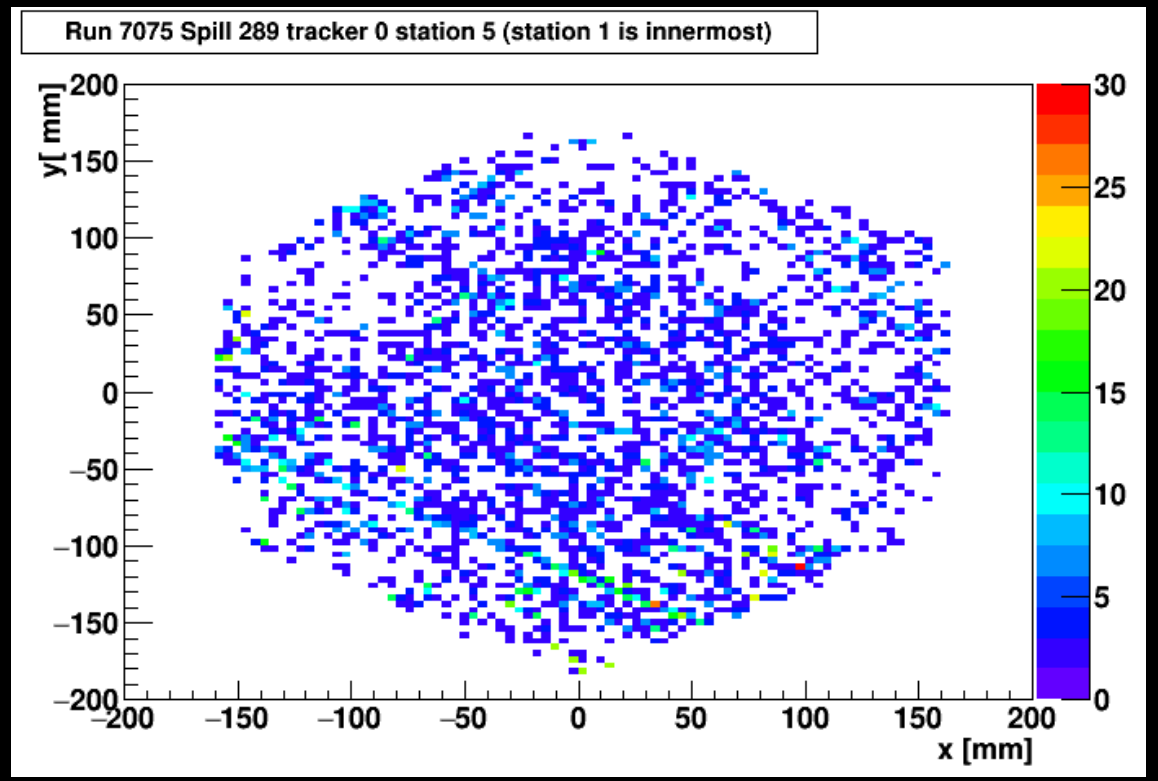
- Tracker tested with full MICE beam for the first time – data looking pretty good
- Bugs found and fixed e.g. plane ordering in mapping file
- Tracker configuration files now in CDB, script in MAUS to access them
- Zero suppression code updated
- MC noise algorithm updated
- Efficiency studies underway – initial results look good
- Kalman fit – in use, but still a number of issues to fix (pz recon not good, 40MeV offset vs TOF calculation)
- Online Reconstruction plots in place, more requested

Kalman Fit Residuals: MC and Real Data





Beam Profile: Real Data



Global Tracking

- Global Tracking: integrate data from all detectors to give best possible track fit and PID through the beamline
- PID analysis now using data from all detectors except CKOV (on its way)

  ➡ PID framework complete excepting CKOV
- PID efficiency and purity MC analyses underway
- Tracking done in two parts: track matching and track fitting
- Runge-Kutta integration (RK4), which propagates tracks for track matching, now complete, under testing
- Track matching functional, undergoing optimisation
- Track fitting method not yet chosen (Kalman and Runge-Kutta contenders)

Deployment

- MLCR for live online reconstruction (miceonrec02 and miceonrec03)
- Fast offline reconstruction (miceoffrec01)
- GRID offline reconstruction and MC simulation
- Many local university / lab installations, batch clusters, personal laptops…
- Various Linux flavours: SL6, CentOS, Ubuntu

Next Steps

- Finish optimisation of final tracker track fit
- Complete efficiency and performance study of tracker software and publish
- Finish CKOV work
- Finish Global tracking and efficiency studies
- Add data quality check flags for all detectors
- Sort out any remaining issues with coordinate system translations
- Add event viewer to MAUS as a third party
- Expanded tests (focus on integration tests)
- Finish new Online Reconstruction framework
- Fix remaining memory leaks

Questions

Backup: Possible Future Upgrades

- Build GCC by default, turn on C++11 flag → multithreading, smart pointers, auto, lambdas…
- Upgrade version control and distribution: move from Launchpad and Bazaar to Git and GitHub (Bazaar going the way of BetaMax and HD DVD)
- Upgrade build system (SCons slow and less favoured, CMake used by ROOT, GEANT4…)