MAUS – MC and Online Analysis

Chris Rogers,
ASTeC,
Rutherford Appleton Laboratory

Overview

- Summary of Monte Carlo progress to Step IV
- Few thoughts on how we do “online analysis”

Step IV Tracking “wish list”

- Tracking “wish list”
- MAUS as it stands can do everything expected of a standard accelerator code
  - Place fields with arbitrary position and rotation
  - Place physical apertures with arbitrary position and rotation
  - Wide selection of shapes and field types that can be placed
- But... rather an old geometry for MICE Step IV
  - Needs checking/updating - Snopok
- But... confusion about coordinate systems
  - MICE Step I geometries are in engineering coordinate system
  - MICE Step N > 3 geometries are in “MICE Step VI” coordinate system
    - Central absorber of Step VI is at z=0
    - Tracker/SS is fixed
- Input beam rotation, G4BL interface
- Really, missing a post-doc to look after this stuff

Geant4

- At the moment we take the approach with G4
  - “If it aint broke dont fix it”
  - Stick with same version of G4 until someone complains
- Do we want to update to latest version of G4 before running next year?
  - Then (probably) leave it fixed for Step IV

Step IV Digitisation discussion

- Still aiming for a simplistic digitisation, i.e. bare minimum to get reconstructable digits
  - Tracker make digits for hits over a threshold energy deposition
  - TOF makes digits for all hits in the detector
    - Simple model for cable lengths and light yield
    - No time walk incorrection
  - Cerenkov makes digits for hits over
  - KL, EMR?
  - Trigger, Scalars
  - RF?
- Could imagine full simulation of the electronics
  - What do we get?
    - Calibration cross checks
    - Modelling “weird” stuff e.g. TOF time sag
  - Probably – low on priority list
    - If MICE needs this, will be a challenge
- Discuss…

Online Analysis - Requirements

- How do we make the online analysis work?
- Task
  - Make graphs of statistical quantities for a collection of runs
  - e.g. plot beam mean vs magnet current to look for magnet misalignment
  - e.g. plot beam emittance vs diffuser thickness to check input beam parameters
  - e.g. perform some cuts/sampling on the input beam, plot beam emittance vs a few beamline parameters to find equilibrium emittance
- Reminder of online data flow
  - Read network socket
  - Reconstruct all data spill-by-spill
  - Hand data in memory to histogramming routines
  - Write histograms to disk
  - Hand histograms to web front end for display to shifters
- Currently we never write particle data to disk

Option 1

- We write out a reduced data set of global reconstruction hits for each run, which folks can analyse in a semi-offline way
  - Say global reconstruction “hit” consists of
    - Momentum = 3 doubles
    - Position = 3 doubles
    - Energy = 1 double
    - Time = 1 double
    - PID = 1 int+few doubles
    - 8*8 symmetric error matrix = 36 doubles
  - Each hit = ~50*8 bytes = 400 bytes per hit
  - Write out 100,000 tracks at, say 10-20 z-points ~ 200-400 MB per run (or lets say up to 400 MB per hour of running)
    - How many runs would we want to cache at any one time? Do we need a bigger disk?
    - How quickly can we post-process this data? (Probably okay?)
  - Provide a few scripts to make some “standard plots” based on this data
    - Needs a UI? See how it goes... maybe python is good enough
  - Experimenters bring their own scripts for fancy stuff

Option 2

- Add reducer(s) that calculates beam moments/emittances/etc on data in memory and write in a summary table
  - Module in MAUS framework comes after reconstruction
    - Extra bureaucracy
    - Potentially more reliable + usable
  - Don't ever need to write particle data to disk
  - But can't “change a cut” and then rerun
    - Once the data is out of memory, it's gone
- Probably use Option 1 for “experiments”, Option 2 for standard set of data
  - So for every run, we calculate something like:
  - first and second moments,
  - emittances, Twiss functions,
  - beam envelope at 1 sigma, 3 sigma,
  - angular momentum, Penn-style twiss functions,
  - dispersions, ...

Revised data flow (Assume Option 1)

- Reminder of online data flow
  - Read network socket
  - Reconstruct all data spill-by-spill
  - Hand data in memory to histogramming routines
  - Write histograms to disk
  - Write reduced data set to disk
    - Experimenters do their thing
  - Hand histograms to web front end for display to shifters
- Assume no online Monte Carlo required

FIN