

Implementation of GDML Parser in MAUS

Ryan Bayes

School of Physics and Astronomy
University of Glasgow

17 July, 2014

Purpose and Goals

Reduce Geometry Loading Time

- Used a rendering of complex geometry using tessellated solids in MICE Modules
- Loading the geometry takes 15 minutes with MICE Modules
- Loading tessellated solids faster using G4GDMLParser. **How Fast?**

Methods

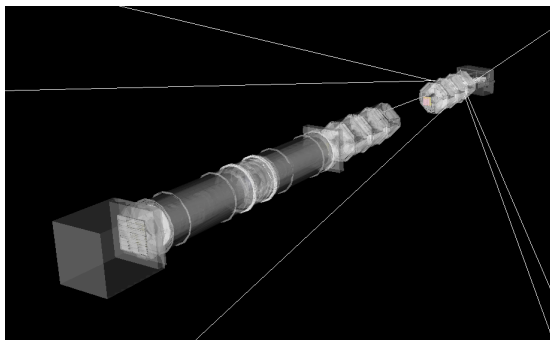
- Produced a version of MAUS with an altered Geant4Manager.
- Introduce G4GDMLParser to define geometry.
- Use MICE Modules to define fields (and virtual planes?).
- Define active detectors in GDML to make a fair test.

Metrics

- Would like to have a simulation that can be used for online

Status of Geometry in GDML

MICE Geometry from GDML



- Complete geometry from GDMLParser shown.
 - CKOV not shown — still not finalized.
 - Visualization from FreeWRL
-
- Step IV geometry with all detectors (except finalized CKOV) available in CDB.
 - Detectors not compatible with GDMLParser
 - Requires addition of G4Detectors by MAUS.
 - Raises a segmentation fault when a daughter is added to parser

Simulation Loading Time

- Use Step IV geometry in MAUS v0.8.5.
- Start 226 MeV μ^+ solenoid beam at position of Q7.
- Simulate different number of spills to check loading time.

Consider two timing metrics

- User time: time in reality.
- Real time: time the computer is actually doing something.
- Loading time for GDML parser:
 - User: 350 s (5 min 20 s)
 - Real: 56 s
- Visualization engaged.

