SoLID Simulation Software

Seamus Riordan
University of Massachusetts, Amherst
sriordan@physics.umass.edu

January 29, 2011

Overview

- Motivation and Goals
- Status
- Future Work and Timeline

Motivation

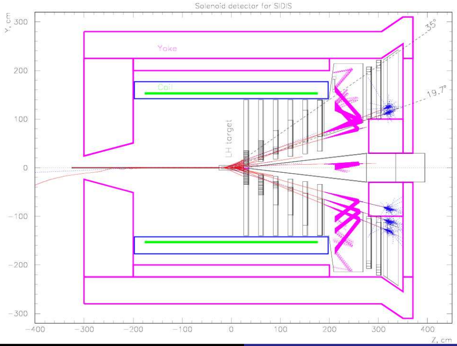
Fully simulate design of SoLID for various experiments

- Optimize magnetic fields
- Optimize baffle design for PVDIS
- Evaluate detector needs
- Determine backgrounds
- Tracking, event reconstruction

Previous Work

Most simulation work for SoLID was done using GEANT3 by Eugene Chudakov continued by Transversity collaboration

- All done in GEANT3 (no longer supported)
- COMGEANT used to specify geometry externally
- Magnetic fields done using POISSON
- Much of the auxiliary code for analysis written in FORTRAN



Present Objectives

For future work

- Use modern programming languages, supported software
    - Re-develop simulation using Geant4
    - Utilize ROOT framework in analysis
- Minimize “wheel reinvention”
- Have evaluations complete for review (end of summer)

Simulation Requirements

- Rapid development of new geometries/detector configurations
  - Geometry/detector specification should be "clean", readable. Debugging, minimizing potential bugs important
  - Geometry/sensitive detectors should be specified externally to code. Nothing "hard wired"
- Allow for variety of evaluations - background rates, intricate detector responses
- Support variety of physics processes
- Configurations must be reproducible
- Fast event generation, accept input from external event generators
- Possible output in compact form
- Interactive 3D geometry viewer
- Good documentation available (important for continuing work over many years)
- Must run on 64-bit farm

Simulation Group

Representation from both PVDIS and Transversity collaborations

- Zhiwen Zhao, UVA - Geometry development, magnetic field generation
- Paul Reimer, Argonne - Magnets
- Lorenzo Zana, Syracuse - Event generation
- Simona Malace, Duke - Output
- Seamus Riordan, UMass - Software design, detector responses

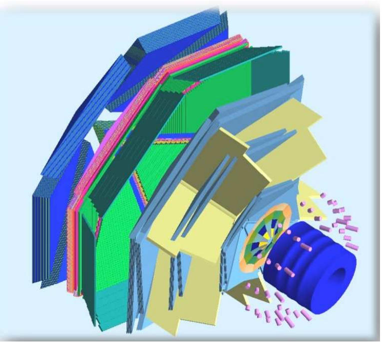
GEMC Support

- Maurizio Ungaro - UConn

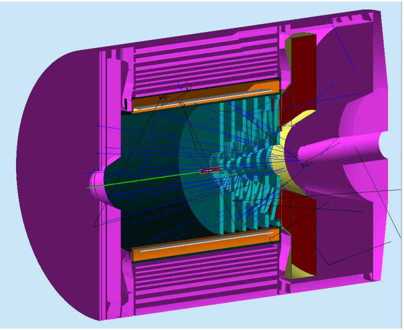
Present Implementation

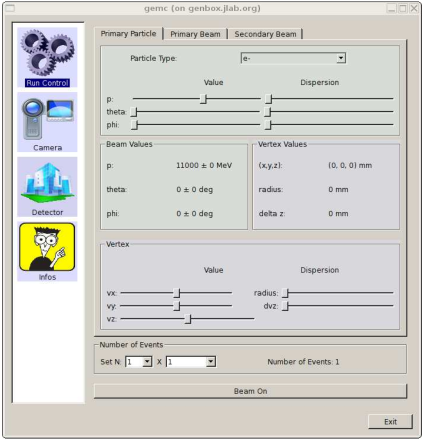
Present design is to implement GEMC, the CLAS12 simulation software `http://gemc.jlab.org`

- Geant4 based
- Geometry, detectors are specified in mysql database
  - Scripts are used to generate geometry, which is then loaded (tools exist)
  - Pre-defined detector outputs exist, can be modified
- Used for CLAS12, GlueX, HPS



- Support for reading in field generated by POISSON
- Output in text or evio (which can be converted to ROOT)
- Nice GUI interface and geometry display in Qt/OpenGL



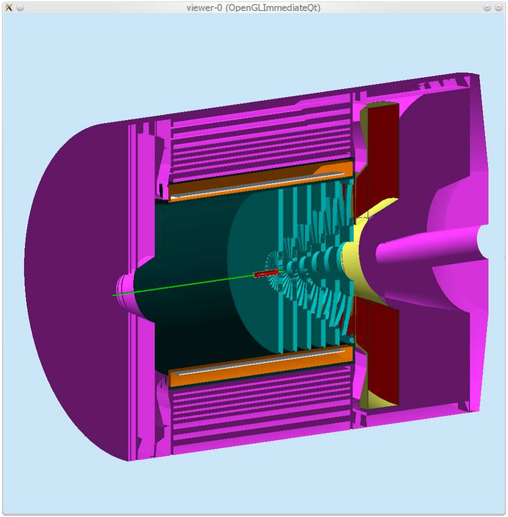


Magnetic Fields

- Magnetic fields can be generated by POISSON
  - Provides 2D map for azimuthally symmetric geometry
  - Software is free
  - Map can be converted using existing tools to something usable by GEMC
- Support for importing TOSCA maps is also available

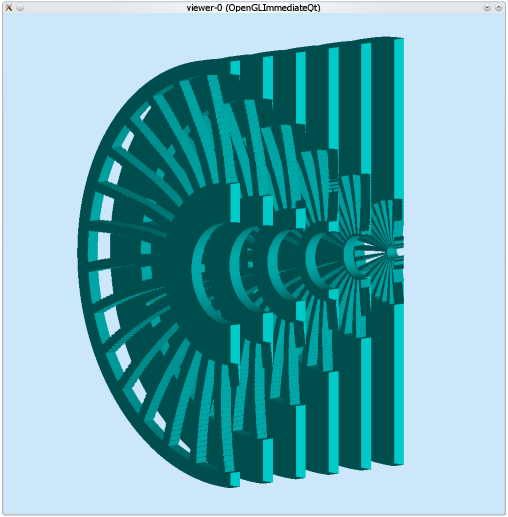
Present Implementation

Babar simulation specified by Eugene implemented in GEMC



Present Implementation

Baffles treated as 30 slit per plate, 20 blocks per slits



Present Implementation - to be done

Needed modifications

- Present physics lists need to be expanded
- All materials are hard-coded
- mysql allows for geometries to be overwritten - need a reproducibility strategy
- Incorporate basic event generators

Must minimize dependence on GEMC collaboration for support

- Build GEMC as shared object library, add in additional classes as needed
- No need to fork code - keep a working version, update it with GEMC collaboration updates
- Should run our own mysql server, off-site access would be desirable

Present Implementation - Other ideas

Python bindings

- Python bindings in Geant4 are shipped with present versions (though not built on CUE machines)
- Embedded Python interpreter can specify geometry and detectors as external script
- Reasonable implementation written, needs to be tested

Nice things to have:

- Native ROOT output, simple interface of ROOT trees to detectors
- Compact storage of inputs, software and field map versions tied to data

Milestones 1/2

We are here, Nov - Jan 2010

- Produce general software as specified above
- Basic field map to start with

Feb 2011

- Implement Eugene's baffle design and reproduce acceptance
- Produce additional magnets

Feb - Apr 2011

- Reproduce baffle design software
- Simple detector responses (done in GEMC)
- First order experiment designs, momentum resolutions

Milestones 2/2

May - Jun 2011

- Realistic detector responses/digitization
- Steal work from SBS for GEM responses, GEM tracking
- Raw background studies

Jul+ 2011

- Tracking algorithm evaluation
- Radiation studies

Upcoming PVDIS Baffle Work

- Ray trace with (charged)geantino with appropriate energy, define baffle profiles
- Run with energy loss for DIS e^- generated and prompt γ
- Adjust profile to optimize acceptance and minimize background

- Should develop a figure-of-merit scheme for optimization
- Need to produce software for optimization, easy to export to geometry, easy to hand to engineers

Summary

- Simulation of SoLID needed to evaluate various configurations of magnetic and detectors
- Moving forward with SoLID simulation using GEMC
- Established a set of milestones to be completed in time for the technical review at the end of the summer