Simulation. Status and plans

V. Perevoztchikov
Brookhaven National Laboratory,USA

STAR

Directions

- StarSim
- New StarSim (geometry definition)
- Star VMC simulation
- Event generators
- Star VMC reconstruction

StarSim

StarSim was used in STAR many years. It is written in Fortran/Mortran. We are going gracefully retired it. But some development is still going on.

- Few bugs fixed;
- Improvements:
  - Volume rotations. Several rotations of different kind(alphax=,ort=,Phix=,...)in the given order will performed now;
  - Filtering in points:after Pythia, after vertex generation, and after Geant, is implemented. HEPEVT standard used
- Thermal Pt distribution will be soon;

Star VMC simulation

The future StarSim will be based on ROOT Virtual Monte Carlo (VMC), which allows to run within the BFC.

- Simplifies embedding;
- Run event generators in the same chain;
- Output directly to .geant.root or .MuDst.root (no more fzd files)
- Geometry handled by ROOT classes
- Particle transport handled by one of:
    - Geant 3
    - Geant 4
    - Fluka

Event generators

Now we use only Pythia & Hijing generators. They are in StarSim environment and communicated with BFC via .fz files. VMC approach will simplify it and allow:

- Separate library for each event generator which could be plug-in in BFC
- Special interface, based on ROOT one, will allow easily to add new generator. It could be more generators in future;
- No intermediate file between generator and Geant

StiVMC Reconstruction

VMC approach also allows to improve reconstruction. What is new in StiVmc?

- StiVMC uses (existing) Geant 3 geometry in VMC framework
- Improved description of energy loss and multiple scattering
- Simplifies addition of new detectors. No need to define interface classes such as StiTpc, StiSvt, etc...
- Tracks are propagated using standard Geant 3 machinery, which is well tuned/debugged during last 20 years

StiVMC reconstruction(2)

StiVMC features not related to VMC:

- Current Sti framework assumes only cylindrical detectors. StiVMC allows arbitrary sensitive detector orientation and shape;
- Sti has difficulties when tracks nearly touch the detector plane. It works well only for tracks close to orthogonal to plane.(~60). This is related to sin(x)=x+x*x*x/6. When x is big, linear approximation is wrong. In fit we always use linear approximation. StiVMC uses different fitting plane, and in result it will work well with the almost zero touching angle;

StiVMC reconstruction(3)

- In Sti Hit error parameterization is rather bad. Much better one is developped for StiVMC

StiVMC will be much more precise for low momentum tracks and good for such precision searches, as D0 mainly by:

- More accurate account of dead material;
- No accuracy loss for touching tracks;

StiVMC performance

Right now, main goal of StiVMC is a correctness, not a speed.

But:

- Due to special organization of Geant geometry, track propagation will be slightly faster.
- In cases when track is close to orthogonal, simplified algorithm could be used.
- Parallelization is possible in seed finder and even in track fitting, if to use tracks one far from the other.
- And of course, the final cleanup of code will double the total speedup;

StiVMC implementation

- Each sensitive TGeoVolume has proxy object (done);
- This proxy created and filled automatically and contains:
    - Orientation, positions and type of detector plane (done);;
    - Hit containers filled automatically for each event (done), using general StEventHit iterator (done)
- Seed finder, using 3d map hit container finds the seeds(done)
- Hit error calculator provides errors using track info(done)
- Track finder propagates seeds, using VMC (in debug)
- Track fitter not yet ready, but most of "bricks" will be reused from Sti;
- Also a lot of other Sti code will be reused(not done);