GEMC Status

CLAS12 Distribution

What’s new

TODO for CLAS12

GEMC Distribution

gemc: **2.5**
geant4: **10.2.p03** (JLAB_VERSION: 2.0)

Next gemc: devel (**2.6**) (JLAB_VERSION: devel)
Next geant4: **10.3.p01**

- 3 (2 “new”, can’t live with) bugs, all reported
- 1 fixed

GEMC and CLAS12:

- Dedicated clas12 tags, due to fast development of digitization routines and geometry.
- Sync with Reconstruction

https://github.com/gemc/clas12Tags

clas12Tags

This is a series of tags of the GEMC source code to match the various simulation/calibration/reconstruction software versions.

The tags also contain the geometry files and the gcard to run gemc.

Every tag is installed in /group/clas12/gemc

To use:

source /group/clas12/gemc/environment.csh

How to use

If no tag is given, the script will currently load 4a.0.0.

Tags

LOG

- 4a.0.1: Same as 4a.0.0 but with FTOF geometry fix
- 4a.0.0: KPP configuration. Fixes in source and hit process for FTOF, added EC gains. Java geometry uses now coatjava 3. Database fixed for DC geometry. Linear time-to-distance for DC. CTOF in the KPP position configuration in the new kpp.gcard.
- 3a.1.0: same as 3a.0.2 but with DC time-to-distance implementation.
- 3a.0.2: same as 3a.0.1 but with CND fix.
- 3a.0.1: same as 3a.0.0 but ctof has status working.
- 3a.0.0: git commit d3a5dc1, Dec 2 2016. Includes FTOF and CTOF paddle delays from CCDB, and CTOF center off-set.

Not the “git” way. But this mode is necessary for the current fast development.

What's new

Generator:

- SKIPGEN option to skip N events
- BEAGLE support (EIC) (LUND++)
- Writing out generator header and particle info (even info not used by geant4, like weight)
- stops events if no more events in the generator
- ION_P option to generate ions (EIC).

CAD:

- Optical processes support: materials and mirrors
- Can make copies

What's new

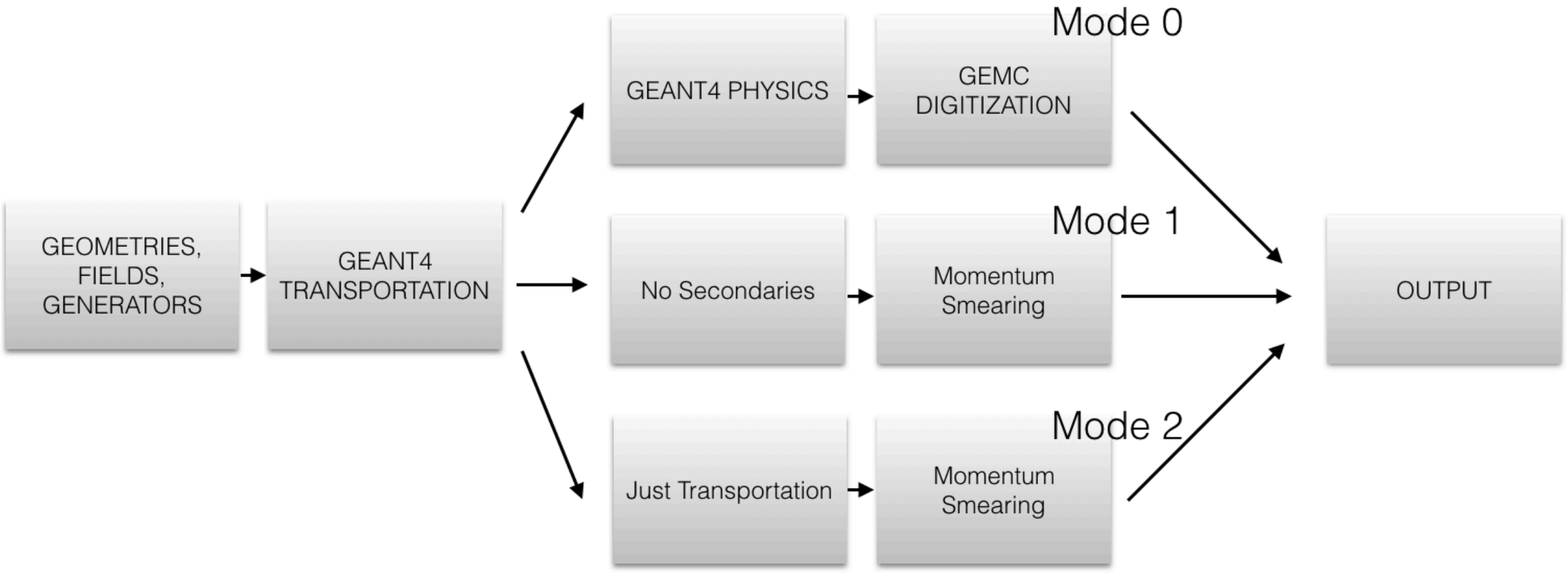
Various

- Added volume informations (GUI and batch):

  - ✓ surface area
  - ✓ volume
  - ✓ weight

- fixed G4LogicalBorderSurface
- added linear interpolation for dipole fields
- added field information in digitization

FAST MC Mode: FASTMCMODE option



Mode 0: is the full geant4 simulation.
Mode 1: turns off secondaries.
Mode 2: turns off all physics except transportation.
Mode 10: Same as mode 1 but with process routines enabled.
Mode 20: Same as mode 2 but with process routines enabled.

FAST MC Mode: FASTMCMODE option

Full CLAS12 setup (geometry, fields, etc)

Output: smeared and unsmeared momentum

```
--- Generated Particles Bank --
 - Particle 1 pid: 11    -  mom: (391.597,-1100.79,3825.56) MeV    -  vert: (0,0,0) mm
 - Hit >dc< Has 0 hit with nphe 1 nphe and time -1 ns
   orig      mom: (391.538, -1096.99, 3821.49)
   smeared   mom: (392.538, -1096.99, 3821.49)
 - Hit >ftof< Has 0 hit with nphe 1 nphe and time -1 ns
   orig      mom: (336.35, -946.503, 3306.99)
   smeared   mom: (336.35, -946.503, 3306.99)
 - Hit >htcc< Has 0 hit with nphe 180 nphe and time -1 ns
   orig      mom: (1.09772e-06, -4.24849e-06, -3.77395e-06)
   smeared   mom: (1.09772e-06, -4.24849e-06, -3.77395e-06)
 - Hit >pcal< Has 0 hit with nphe 3 nphe and time -1 ns
   orig      mom: (120.305, -305.377, 1105.12)
   smeared   mom: (120.305, -305.377, 1105.12)
--- End of Generated Particles Bank --
```

Momentum can be smeared by special digitization routine

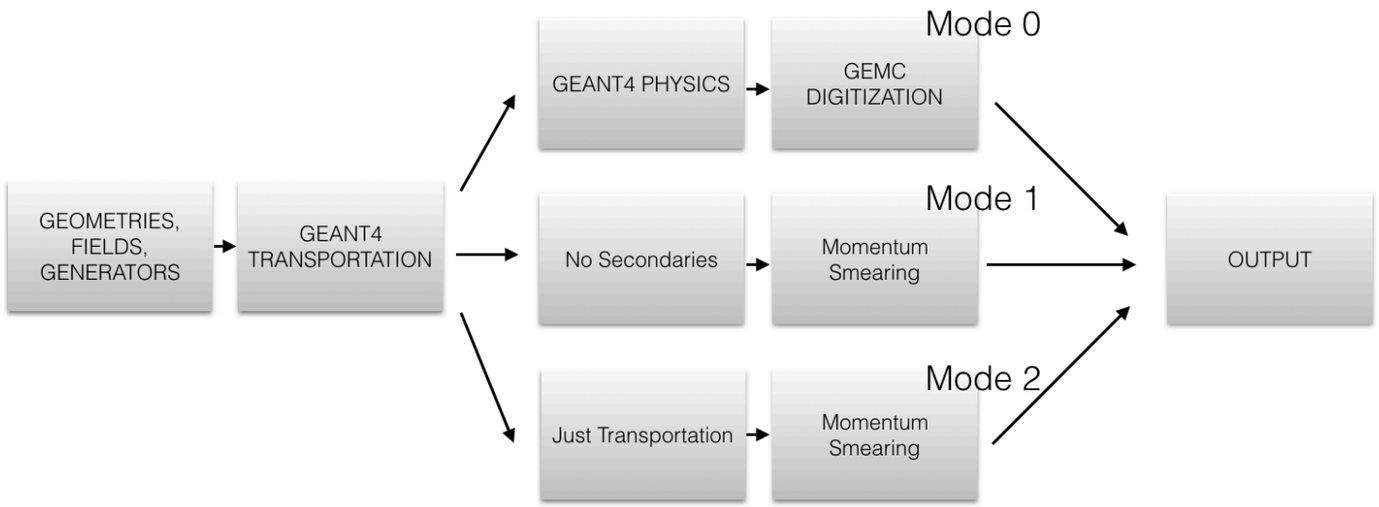
```
G4ThreeVector dc_HitProcess :: psmear(G4ThreeVector p)
{
    G4ThreeVector y(p);

    y.setX(p.x() + 1);

    return y;
}
```

FAST MC Mode: FASTMCMODE option

Performance



- Mode 1: ~10x faster.
- Mode 2: ~200x faster.

Output:

- evio
- txt
- Future: ROOT output

GEMC Performance

~3Hz on one core of my (slow) desktop

CLAS12 will have > 10K dedicated cores

If 10 users run 500 jobs each (1/2 the farm):

`1.5 KHz rate of simulation each user`
`specific physics process >> clas12 rates`

CLAS12 Simulations

In process: Micromegas geometry + digitization

- Full geometry and digitization ✅
- use CCDB database for ALL the digitization ✅
- use CCDB database and Coatjava (many detectors)

~6 months if dedicated manpower

In place, not used yet: FADC

- Translation table mechanism ✅
- Mode 1 and Mode 7 mechanism ✅

TODO:

- provide V(t) function in digitization for full FADC simulation
- Hipo output?

~6 months with help from detector groups

Test Suite

> Running tests on Darwin_macosx10.10-x86_64-gcc4.2.1

> Compilation test
 > Compilation options: -j16 OPT=1
 > Compilation: PASSED
 > Warnings: NO WARNINGS

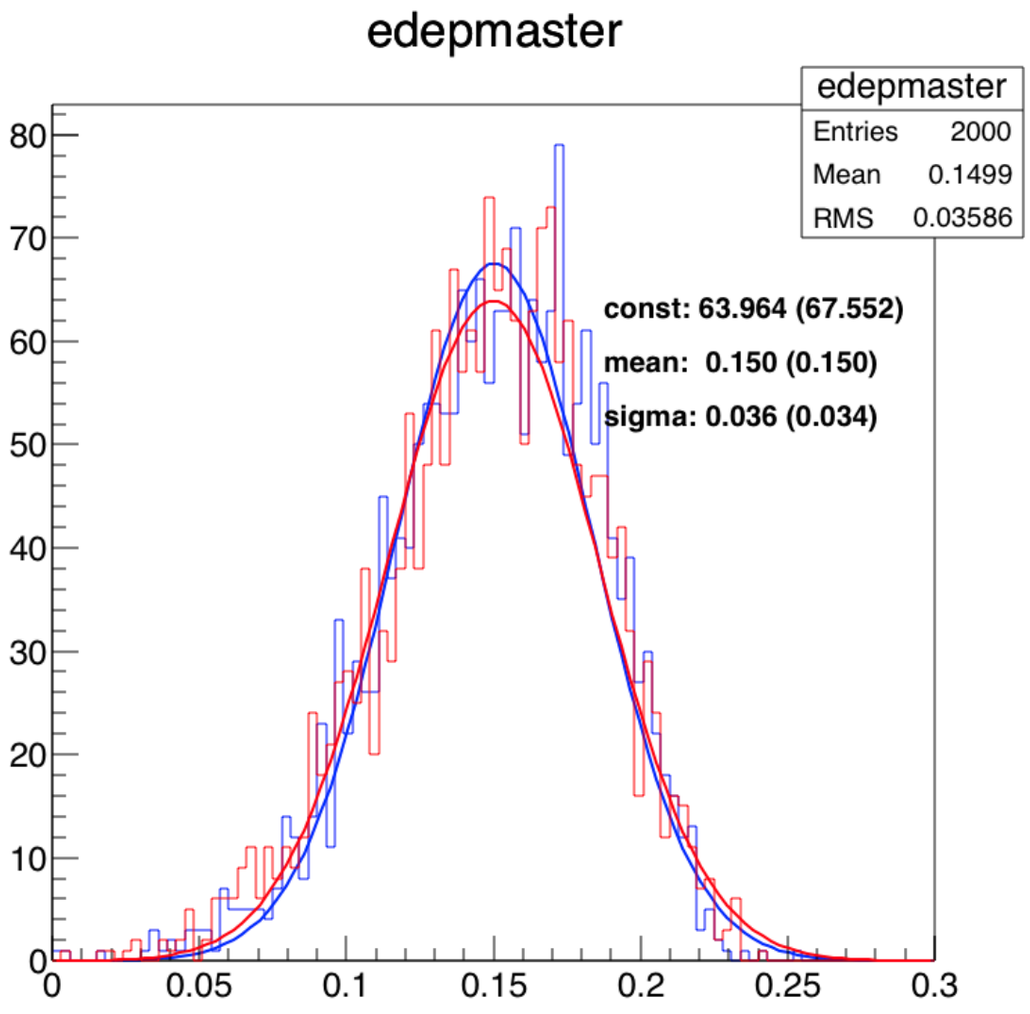
> FTOF Running test for 5000 events
 > master time: 59 - branch time: 58
 > Edep Landau Constant percentage difference: 3.51759 %
 > Edep Landau MPV percentage difference: -0.393551 %
 > Edep Landau Sigma percentage difference: -2.29966 %

> EC Running test for 2000 events
 > master time: 1345 - branch time: 1360
 > Edep Sampling Fraction Constant percentage difference: 3.46719 %
 > Edep Sampling Fraction MPV percentage difference: 0.141233 %
 > Edep Sampling Fraction Sigma percentage difference: -3.1772 %

> Full CLAS12 Running test for 2000 events: PCAL
 > master time: 714 - branch time: 715
 > Edep Sampling Fraction Constant percentage difference: -5.31061 %
 > Edep Sampling Fraction MPV percentage difference: -0.339036 %
 > Edep Sampling Fraction Sigma percentage difference: 6.63857 %

> Solenoid test
 > master time: 1 - branch time: 1
 > Solenoid Position Test: SAME
 > Solenoid Values Test: SAME

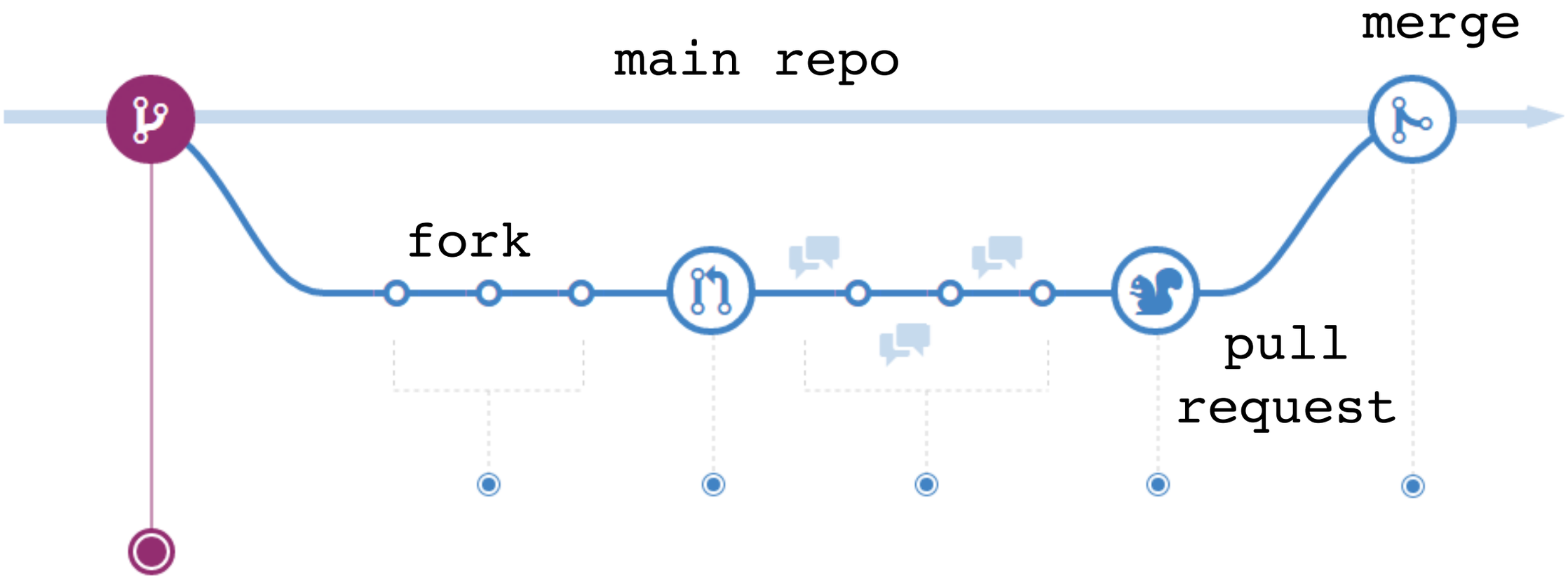
> Torus test
 > master time: 21.5132 - branch time: 21.4379
 > Torus Position Test: SAME
 > Torus Values Test: SAME



GEMC development policy: guidelines

https://gemc.jlab.org

github bitbucket gitlab gitorious + others > 25M users



Many improvements, bug fixes using GitHub “issue” button.

Summary

GEMC development for CLAS12 is mature.
Will now shift focus on:

- User support
- Documentation
- FADC and Digitization for CLAS12: make simulation as realistic as possible
- Bug fixes and Feature Requests (please use the "issue" button!)

work started on gemc 3.0:

- code more modular
- c++11
- optimized
- new geant4 capabilities:
  - ✓ multithread
  - ✓ g4 analysis