Simulation software updates

Rich Holmes
Syracuse University
SoLID Simulation Meeting 29 Jan 2019

User header variables

In LUND file format for generated particles, 1st line for each event is a header. Typically 10 numbers, includes event weight.

In GEMC versions <= 2.5, all header entries except the first and fifth are written to variables var1, var2, var3... in the "header" bank of the EVIO output file.

In GEMC >= 2.6, all header entries are written to variables userVar001, userVar002, userVar003... in the "userHeader" bank.

User header variables

We normally don’t use EVIO files but convert to ROOT using evio2root utility.

No one told evio2root about this change!

This has been addressed. evio2root updated to write userHeader tree correctly.

Event weight is userHeader.userVar010 instead of header.var08. Not backward compatible but an easy change to make.

Random number saving

Problem: You have a rare signature you want to investigate more closely, e.g. by visualization. You don’t want to click through 10000 event visualizations to get to the next such event.

Solution: In a batch run, save the random number generator state (as of the start of the event) for events that match your desired signature.

Use saved state to initialize RNG and run single events with visualization.

Random number saving

New command line option:

```
-SAVE_SELECTED=“<id>, <pid>, <low limit>, <high limit>, <variable>, <directory>”
```

Saves RNG state for events having a hit where

- Detector ID matches <id> AND
- Particle ID is <pid> AND
- <variable> is in the range from <low limit> to <high limit>
- Saved to <directory> (optional and defaults to the current directory)

Random number saving

For example SAVE_SELECTED="7xx10000, 11, 0.0*MeV, 2000*MeV, trackE, ./” saves RNG state in files in the current directory for events in which there was a hit in detectors 70010000, 70110000, 70210000, etc. by an electron with energy from 0 to 2 GeV.

Caveat: Uses GEANT4 code for RNG saving; this fails if multiple jobs are writing to same directory and is not thread safe. Should improve in new GEANT4 but GEMC code will need updating at that point.

Random number rerunning

Can use GEANT4 /random/resetEngineFrom command to set up an event to rerun single events.

New option to rerun multiple events:

-RERUN_SELECTED=”<run>, <directory>”

<run> is a run number; <directory> is a path to a directory, defaulting to "./".

Random number rerunning

This option causes GEMC to read files in the given directory with filenames of the form run<run><evt>.rndm, in numerical order by event number <evt>, and use them to initialize the random number generator prior to the start of each event, thus rerunning the events.

If reading from a LUND file, lines corresponding to non-rerun events are skipped over so that the rerun events are started with the correct input.

Event numbers shown in output files are set to match those of the original events.

Trajectory history saving

Event banks like “flux” save some information about the mother a particle causing a hit.

But sometimes one wants to trace the ancestry deeper to understand a background.

New command line option:

```
 -SAVE_ALL_ANCESTORS=1
```

Writes a new bank with information on all tracks ancestral to a hit.

Trajectory history saving

E.g. (text output):

```
 Ancestors Bank {
        pid:   11      11      11      11
        tid:   1       13      4       58
        mtid:  0       1       1       4
        E:     9530.25965052   0.800577634026  12.1813536532   5.6758631925
        px:    -540.462206724  0.797468359175  3.16881043694   -1.66161601057
        py:    -4246.15529432  -0.72036962776  -4.74176399307  5.83124923801
        pz:    8515.4949786    0.55156591658   11.3273566356   -1.11879727696
        vx:    -2.16633088223  -91.0892511365  -78.1072916064  507.464208736
        vy:    1.51014964563   -1181.70604397  -886.219728192  -543.725452661
        vz:    230.668352917   2591.75488573   2003.63246453   2288.1396573
  }
```

evio2root correctly generates an ancestors tree from this bank.

Bug noted

If a detector to be treated as sensitive (including mirrors) has identifier “no” in GEMC geometry file, get_identifiers(string var) in identifiers.cc returns an uninitialized value. For example hits in LGC mirrors end up with an id == 29 (became 101 when I recompiled with debug statements added).

Status

- User header variables: Completed, tested, merged
- Random number saving: Completed, tested, merged
- Random number rerunning: Completed, tested, pull request pending
- Trajectory history saving: Completed, tested, merged
- id bug: Reported