

## GGLAND — command line simulations

*H.T. Johansson<sup>1</sup>, R. Thies<sup>1</sup>, S. Buller<sup>1</sup>, A. Heinz<sup>1</sup>, S. Lindberg<sup>1</sup>, J. Magnusson<sup>1</sup>, T. Nilsson<sup>1</sup>, T. Rathsman<sup>1</sup>, and the R<sup>3</sup>B collaboration*

<sup>1</sup>Chalmers Univ. of Technology, Göteborg, Sweden

**GGLAND is a small wrapper program around the GEANT3 and GEANT4 physics simulation libraries. Its task is to make it easy to perform the simulations needed for physics analysis of LAND/R<sup>3</sup>B experiments, as well as other smaller nuclear physics setups.**

The basic idea behind GGLAND [1] is that the physics knowledge needed for particle transport through matter is handled by the general libraries GEANT3/4 [2]/[3]. To perform a simulation, the user has to write code to interface the libraries, to setup and declare the geometry, generate particles, and retrieve results; often an obstacle for quick tests.

The wrapper program has grown organically out of a need to simulate calibrations of the Crystal Ball detector. But for each new feature, the handling has been generalised to allow use in other cases as well. GGLAND has thus become a rather versatile tool, especially for quick back-of-the-envelope simulations (see Fig. 1), since the user generally does not have to write or modify any code. Operation is controlled by command-line options and output data can be recorded in ROOT-files, with content very much resembling real data that has passed the first steps of analysis.

The program can run simulations in parallel by forking itself as several slave processes after setup. The master process gives the pseudo-random seeds for each event, and automatically merges the output data in original event order, with results independent of actual simulation order.

### Geometry definition — detectors

The geometry elements making up the detectors are created on program startup, every time. This avoids bitrot in the creation code, is a very small fraction of the execution time, and gives flexibility in the creation and subsequent hit handling. Each detector type can be maintained in a single source file. The creation function constructs objects according to variables in an associated structure, describing vital measurements and other characteristics of the design. The default values can be overridden on the command line. Several detectors using the same design but with different parameters can be instantiated, as well as exact clones. Magnetic fields are placed using the same mechanism.

### Particle gun

For each simulated event, one or more particles need to be generated, and passed to the physics interaction packages of GEANT3/4. The particle gun takes options to control various forms of randomisation of the source location, particle type, direction and energy. Matching the flexibility

```
./land_geant4 --gun=T=1MeV,gamma,z0=-2cm \  
--test=d=1cm,type=Ge,tree=1 \  
--tree=test4.root --events=100000
```

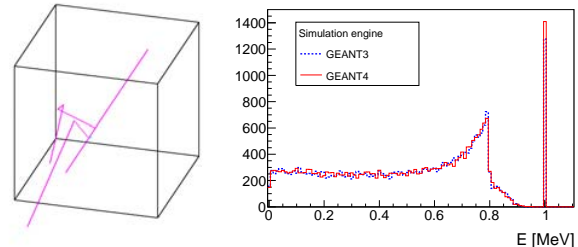


Figure 1: Top: command-line example with a beam of 1 MeV photons directed at a 1 cm<sup>3</sup> cube of Ge, using a flexible test detector. Left: display of 10 events. Right: results, showing full energy detection and the Compton edge.

of physicists in specifying energies, many ways are supported: kinetic, total, momentum,  $\beta$ , and  $B\rho$ . Particles with correlations can be produced, e.g. for  $\gamma$  decay schemes with level feeding and back-to-back photons. Muons can be drawn from a realistic cosmic distribution [4], as well as particles in phase-space distributions, and boosting (e.g. reaction products from a fixed beam velocity).

To use particles from other reaction generator codes, particles can also be created from an input file.

### Digitisation

For selected detector volumes, the discretised energy deposits reported by GEANT3/4 are collected. They are combined by a clusteriser, considering separation in space and time to determine which deposits to join. The resulting hits can be stored in ROOT trees for subsequent analysis. To help understand the processes simulated, the vertices with particles produced can be dumped as structured text, which can be coloured by a special markup program (*ggmark*).

### References

- [1] H.T. Johansson, “The *ggland* command-line simulation wrapper” (2013), Göteborg, <http://fy.chalmers.se/~f96hajo/ggland/>
- [2] “GEANT, Detector Description and Simulation Tool, CERN Program Library Long Write-up W5013”, CERN Geneva.
- [3] S. Agostinelli *et al.*, “Geant4—a simulation toolkit”, NIM A 506 (2003), 250–303.
- [4] J. Kempa A. Krawczynska, “Low energy muons in the cosmic radiation”, Nucl.Phys. B (Proc. Suppl.) 151 (2006) 299-302.