The Geant4 Version 10 Series

Makoto Asai[1], Gabriele Cosmo[2], Andrea Dotti[1], Laurent Garnier[3], Ivana Hřivnáčová[4], Sebastien Incerti[5], Vladimir Ivantchenko[2], Alberto Ribon[2], Marc Verderi[6], and Dennis H. Wright[1]

[1]*SLAC National Accelerator Laboratory*
[2]*European Organization for Nuclear Research (CERN)*
[3]*Laboratoire de L'accélérateur Linéaire, IN2P3*
[4]*Institut de Physique Nucléaire Orsay, Université Paris-Sud, IN2P3*
[5]*Centre d'Etudes nucléaires de Bordeaux Gradignan, IN2P3*
[6]*Laboratoire Leprince-Ringuet - École polytechnique, IN2P3*

on behalf of the Geant4 collaboration

Abstract

A major release of Geant4 version 10.0 was made as scheduled in December 2013, which included the adoption of multithreading, improved treatment of isomers, the extension and improvement of physics models, enhancements in variance reduction options, improvements to the geometry modeler with a revised implementation of most geometrical primitives, advances in histogramming tools, and visualization and graphical user interfaces. In December of 2014, version 10.1 was released, which built on the advances of release 10.0. The implementation of multithreading and its benchmarking results will be discussed. Certain electromagnetic and hadronic model extensions added in 10.0 and 10.1 will also be discussed along with their effects on validation results with experimental data. Also, new variance reduction options, an embedded histogramming tool with multithreading capability, and visualization and graphical user inteface improvements will be highlighted. Finally, the prospects of short and longer term refinements of the toolkit beyond version 10.1 will be discussed.

1 Introduction

In December 2013 a major upgrade of the Geant4 toolkit (Agostinelli et al., 2003; Allison et al., 2006) was released as version 10.0. This upgrade was a response to demands for improved physics modeling, the need for faster and more efficient simulations, and the desire to keep pace with modern hardware and software trends. Since then, development has continued, including improved physics and geometry modeling, and refinement and consolidation of the use of multithreading throughout the toolkit. These improvements were released in December 2014 as version 10.1.

We discuss here the extensive changes to the kernel and how multi-core architectures were exploited to provide more efficient use of hardware, and improvements in other Geant4 categories which were extensive enough to warrant a major release, along with the measurements of both computing performance and memory usage.

Requests to improve physics simulations, in particular for hadronic showers, have come mostly from the high energy physics community. This has led to a concentration on parton string and cascade models which has improved the agreement of simulations with the calorimeter data of experiments. Significant input from the space and medical communities has led to improvements in low and medium energy models, and in particular to the extended treatment of isomers. These and other improvements in both electromagnetic and hadronic processes were included in the 10.0 and 10.1 releases.

To aid in the interpretation of simulated results, Geant4 offers an array of visualization options and graphical user interfaces (GUI), most of which have seen significant advances recently.

Version	Constant term (MB)	Slope (MB/thread)
version 10.0.p02	156	21
version 10.1.beta	128	17
version 10.1	162	10

Table 1: Linearity of memory use as a function of the number of threads. The results of a linear fit $C+m\times N_T$. Results were obtained on an Intel® Xeon Phi™ model 3120A.

2 The Geant4 kernel

2.1 Geant4 multithreading capabilities

To make efficient use of multi-core processors and reduce the memory footprint of the simulation we have developed a version of Geant4 which uses multithreading (MT for short) to share a substantial part of data between threads. For a comprehensive review of the multithreading capabilities of Geant4 see Dong et al. (2010, 2012), Asai et al. (2014) and Ahn et al. (2013).

The memory savings were obtained by sharing the most memory-consuming objects among threads. In version 10.0 this was limited to geometry and the electromagnetic physics tables, while in version 10.1 this was extended to include hadronic processes. Table 1 shows the memory usage as a function of the number of threads for three different Geant4 public releases. The measurements were performed on an application using a simplified geometry description of the CMS experiment at the LHC with a uniform 4 T magnetic field. The FTFP_BERT physics list was used to simulate 50 GeV negatively charged pions incident on the detector.

The improvements obtained with Geant4 version 10.1 are clearly visible: the memory required for each additional thread has been reduced by a factor of two with respect to version 10.0. This important result allows Geant4-based applications to run on co-processors where a large number of cores have access to relatively limited memory.

Threads are independent and require minimal synchronization. Fig. 1 shows the linearity of the speedup obtained on an Intel® Xeon Phi™ as a function of the number of threads. Here, the same application previously described was used. We obtain an almost perfect linearity up to the number of physical cores (in this case 57, shown by the green lines). However in the hyper-threading regime additional throughput can be obtained.

In addition to POSIX-based parallelism Geant4 supports the use of Intel® Threading Building Block (TBB) and MPI. Examples on how to use these technologies are provided with the Geant4 source code.

2.2 Sequential performance

Revision of the most CPU-intensive algorithms used in Geant4 were performed with the goal of improving computing performance. These revisions are done periodically on selected algorithms, and several techniques to improve data and code locality and revise loops were introduced. The results obtained on a simple, sequential application show that we achieved a consistent improvement of CPU performance as a function of the release version (Fig. 2). In certain releases, for example Geant4 9.5, the focus was more on the addition of new physics capabilities than on efficient coding. As a result there was a degradation of CPU performance which was later repaired in release 9.6. The good results obtained in revising these algorithms is clearly visible in the latest version of Geant4. In addition to the introduction of more precise physics and new functionalities, CPU performance has clearly improved on the test applications we selected.

2.3 Fast math library

Profiling results for LHC simulations demonstrated that about 15% of total CPU time is spent in the standard math library which for recent gcc compilers introduced increased precision. As an alternative, a new math library was developed (Piparo et al. 2014). Functions from this library were ported directly into Geant4 to replace the most time-consuming standard functions, and to avoid platform-dependent problems associated

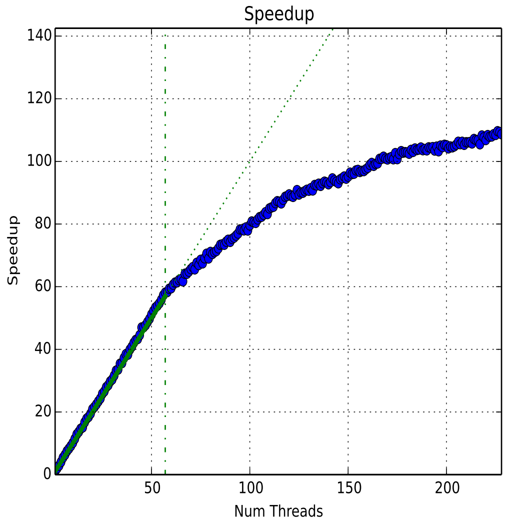

Figure 1: Linearity of speed-up (weak scaling) as a function of the number of threads. The green line, used to guide the eye, shows a perfect linear speedup. The number of physical cores available are marked by the green vertical line. Results were obtained on an Intel® Xeon Phi™ model 3120A.

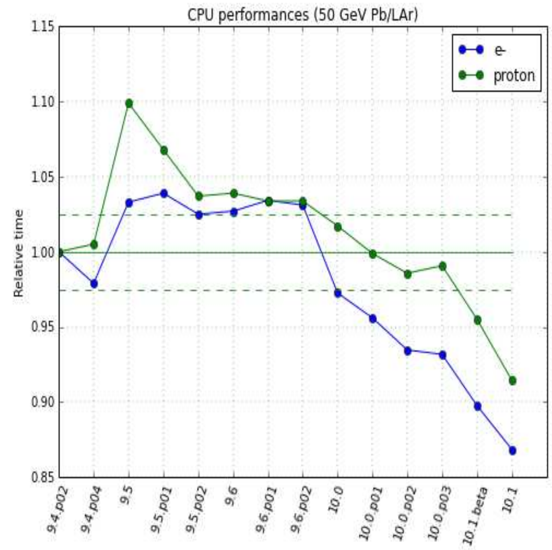

Figure 2: Relative performance as a function of Geant4 version starting from 9.4.p02 (2010) are shown for the simulation of 50 GeV single primaries on a simple *sandwich* calorimeter. Results were obtained on a AMD® Opteron ™ 6128 at 2 GHz.

with linking to an external library. In Geant4 these utilities have been named G4Log and G4Exp. Another utility class, G4Pow, was designed to provide very fast computation of several frequently used functions using lookup tables. This is possible for the case of integer arguments with limited range, such as atomic number and number of nucleons. The overall effect on CPU speedup due to the fast math functions is about 5%.

2.4 Geometry

Multithreading capabilities were introduced in the Geant4 geometry modeler in release 10.0 and further enhancements are underway to organize thread-private memory into workspaces, and to allow for better control and monitoring of the allocated data.

Following the AIDA (Abstract Interfaces for Data Analysis) (AIDA 2012) initiative, it is now possible to make use of the new USolids software library for the modeling of geometrical primitives (Gayer et al. 2012). By selecting the appropriate configuration option at installation time, the new primitives may be adopted in place of the original shapes defined in Geant4. Release 10.1 also allows external installation of the USolids library.

3 Physics

3.1 Physics lists

In Geant4, physics processes and the models and cross sections that implement them are collected, instantiated and assigned to particle types in “physics lists”. While it is the user’s responsibility to compose the physics list classes and make sure that the appropriate physics is used, this is often a difficult task and several fully implemented physics lists have been provided to aid users. These lists are organized around ”physics constructors” which modularize subsets of the physics, such as electromagnetic and hadronic, and make it easier for users to adapt existing physics lists for their purposes.

In release 10.0 there was a consolidation of physics models used in Geant4 which resulted in a reduced number of prepared physics lists offered. This significantly reduced confusion among users as to which list should be used.

In release 10.1 two new physics constructors were introduced: G4HadronHElasticPhysics, which includes the latest hadron elastic models and cross sections, and G4EmStandardPhysics_option4, which includes the most accurate EM models from both standard and low energy sub-libraries and is recommended for many applications in the space and medical fileds. As with other physics constructors, these can be appended to physics lists with very little re-coding.

Also in release 10.1, the ShieldingM physics list, aimed at muon experiments, and NuBeam, aimed at neutrino experiments, were introduced. The former is based on the Shielding physics list with the Bertini to FTFP model transition region at 9.5-9.9GeV, while the latter is based mainly on FTFP_BERT with the Bertini to FTFP model transition region at 3.0-3.5 GeV for protons, pions and kaons. It uses QGSP+G4LundStringFragmentation above 100 GeV (with FTFP used up to 101 GeV) for protons.

3.2 Electromagnetic physics

The Geant4 set of electromagnetic (EM) physics processes and models has been described in detail elsewhere (Agostinelli et al. 2003; Apostolakis et al. 2009; Incerti et al. 2010, Ivanchenko et al. 2011, Asai et al., 2014, Ivanchenko et al. 2014, Karamitros et al., 2014). The EM physics of the toolkit is used in practically all types of simulation application and determines the accuracy of many simulation predictions. Recent developments for Geant4 release 10.1 include the full migration of all EM processes and models to multithreaded mode. All internal tables with cross sections, stopping powers, and ranges are created once and shared between threads at run time. To implement this feature, modifications were introduced to provide read-only run-time access to the data.

3.2.1 Standard models

Multiple and single scattering models (Ivanchenko et al. 2010) were further tuned. A new option was implemented allowing the displacement of a charged particle on a geometry boundary which may be enabled via a user interface command. For electron and positron tracking, a new step limitation algorithm was introduced which may also be used in the case of strong magnetic fields.

Significant refinements were introduced into the high energy EM sub-library. The G4SynchrotronRadiation process is now applicable to all charged particles. The process of positron annihilation to muons or pions (Bogdanov et al. 2006) was reviewed and verified; in particular, cross sections were compared with theoretical expectations. Simulation of these processes allows background predictions for future high energy colliders and for current LHC experiments.

3.2.2 Low energy models

The low energy sub-category of electromagnetic physics processes and models is intended for the simulation of electron, positron and gamma interactions mainly in the keV-MeV range, including the simulation of atomic deexcitation processes (fluorescence and Auger emission). Recent developments have been previously described in (Asai et al., 2014).

In Geant4 10.1, three main features have been added. First, for the simulation of the production of secondaries, such as delta electrons, the new G4VSubCutProducer interface class has been developed in order to generate secondaries below the production threshold. This new feature can be used for example when users are interested in the combination of standard, low energy and Geant4-DNA electromagnetic physics processes. Regarding the simulation of atomic deexcitation, a new user interface command was introduced in order to allow the disabling of production thresholds; if the boolean parameter of the command `/process/em/deexcitationIgnoreCut` is set to "true" then all atomic deexcitation products are explicitly simulated, even if below threshold. Finally, an alternative, updated set of data files for the simulation of atomic fluorescence was provided (Paltani 2014). It can be selected easily in a physics list using a switch in G4AtomicTransitionManager.

3.2.3 DNA processes and models

The DNA sub-category (Incerti et al., 2010) of Geant4 electromagnetic physics processes can simulate the discrete interactions of electrons, protons, neutral hydrogen, alpha particles and their charged states, and a few ions (C, N, O, Fe) for microdosimetry applications in liquid water down to very low energies. For the 10.1 release, new functionalities were added. On the physics side, discrete ionization is now available for other ions: B, Be, Li and Si. In addition, the usage of Geant4 generic ions is also available for ions with atomic number $Z > 2$, which makes it easy to combine Geant4-DNA processes and other Geant4 electromagnetic physics processes. In addition, all Geant4-DNA ionization models now use a common angular generator interface, so-called G4DeltaAngle.

In release 10.1, two new physics constructors are provided: G4EmDNAPhysics_option1, which includes the new G4LowEWentzelVIModel model for simulating electron elastic scattering in liquid water, and the G4EmDNAChemistry constructor to handle simulation of the chemistry stage, namely the diffusion coefficients of the chemical species as well as the reaction rate constants needed to compute the reactions.

During the chemistry stage, the simulation is based on time stepping where all molecules are transported over time steps. The time step durations are computed automatically with respect to the next reaction to occur. However, in order to control the time stepping, a new user class, G4UserTimeStepAction, was added which enables selecting the minimum allowed time step duration, as well as retrieving information before and after time stepping.

3.2.4 Validation of electromagnetic physics

Validation of EM physics is performed on several levels. Because EM physics is used in practically all tests and examples, the Geant4 integration system routinely checks all EM physics models. A specific EM validation suite (Apostolakis et al. 2010) runs on a regular basis (Ivanchenko et al. 2010; Ivanchenko et al. 2011; Schaelicke et al. 2011; Allison et al. 2012; Ivanchenko et al. 2014;) for each reference version of Geant4. This is particularly important for the prompt identification of anomalies and bugs possibly introduced during the development cycle.

Extensive validation activities are also carried out by the user community. Validation and inter-comparison of the different alternative models available within Geant4 is made easier by the common interface which is in place for all electromagnetic models.

A new effort was recently initiated for the systematic validation of the γ-ray attenuation coefficients vs. the XCOM database and of the stopping powers of electrons, protons and αs versus the NIST database. The final aim of this effort is to repeat the analysis of Amako et al. (2005) with a wider set of materials and with the most recent version of Geant4. Dedicated validation of the bremsstrahlung models of Geant4, tailored to the energy range of interest for medical physics, is also underway (Pandola et al. 2014).

3.3 Hadronic physics

Developments in hadronic physics since the 10.0 release have concentrated mostly on consolidation and incremental upgrades. Feedback from experiments has highlighted shortcomings and identified bugs which have been addressed in the 10.1 release. Long-term projects, such as the re-design of several hadronic packages, have continued.

Significant improvement and tuning of the Geant4 FTF parton-string model, based on Fritiof (Andersson et al. 1987), has taken place during the last two years. However, a bug in the Geant4 implementation of the Lund string fragmentation model, which handles hadronization, was recently discovered in the treatment of low mass strings. The result of the bug was that too much energy went into the production of higher neutral resonances, which decayed to charged and neutral mesons, and too little to neutral pions. This incorrectly reduced the electromagnetic portion of showers in calorimeters and made them less compact. The bug was introduced in Geant4 9.5 and subsequent tuning of the FTF model was done with the bug in place. In version 10.1 the bug was removed and the FTF model retuned. The full effect on shower shapes is discussed below.

A new feature were added to FTF for release 10.0: the simulation of nucleus-nucleus interactions. This complements the antinucleus-nucleus interactions added in Geant4 9.5, which were included in physics lists and produce reasonable results for antideuteron scattering as shown in Fig. 3.

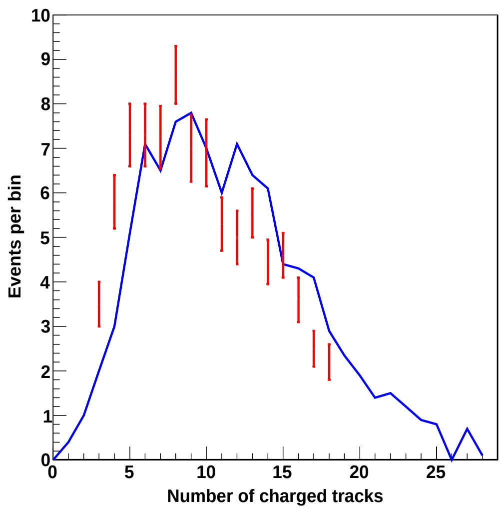

Figure 3: Prediction of the Geant4 Fritiof model (blue) compared to data (red) for antideuterons on Ta at 12.2 GeV/c. The data are form the Lyudmila (Andreev 1989) experiment and are represented as the number of hits recorded versus the number of charged tracks detected.

The re-designed and upgraded INCL++ model (Mancusi et al., 2014) has been part of Geant4 since release 9.6. It has recently been extended to handle nucleon and pion projectiles up to 20 GeV. In the near future it will be extended to handle kaons as well.

In the Bertini-style cascade (Wright and Kelsey 2015), the Barashenkov polynomial angular distributions for intra-nuclear elastic and charge exchange reactions have been replaced by the SAID (SAID 2013) phase shift parameterizations up to about 3 GeV. Above that, fits to data are used. The Bertini cascade is also now used to handle the hadronic vertex of electro-nuclear and muon-nuclear reactions below 10 GeV.

A significant improvement in memory footprint and CPU speed was achieved in the Geant4 nuclear deexcitation models. These models perform the last steps of returning a residual nucleus to ground state after being highly excited by a hadronic collision. To do this, they require access to the nuclear level database for gamma emission. This is done by the on-demand loading of the transition parameters for each nuclear level into a level manager. Until release 10.1, a copy of the level manager was made and stored for each level. Since there are typically tens of levels required for each of typically tens of required nuclei, an unnecessarily large amount of CPU time and memory was expended during a run. As of release 10.1, only one copy of the level manager is created.

A re-design of the deexcitation models is also underway in order to make their code faster, more memory-efficient and more object-oriented. The first stages of this redesign are included in the 10.1 release.

A precise treatment of the elastic, inelastic and capture processes of neutrons from sub-thermal to 20 MeV has a large effect in shielding applications and on some calorimeter observables such as energy deposition and time structure. Several Geant4 models now treat neutrons in this regime: High Precision (HP), LEND, NeutronXS and ParticleHP. Of these, the NeutronXS and ParticleHP models have been added recently. The NeutronXS model uses the precision elastic, inelastic, and capture cross sections of the HP model and performs the final state interaction using the faster, standard hadronic model for neutrons. This results in some loss of precision at very low energy but provides greatly increased speed over the HP model and retains the detailed time structure of neutron propagation. The ParticleHP model is a generalization of the NeutronHP model in that it treats protons and light ions as well as neutrons, up to energies of 200 MeV. It is, like NeutronHP, data-driven and depends on databases. As of the 10.1 release, the NeutronHP and ParticleHP models are both offered, but by the 10.2 release they will be merged into a single package.

The radioactive decay package continues to be improved. It is being re-designed to be faster and more object-oriented, and to address deficiencies pointed out by users. In release 10.1 this model has better energy conservation and improved reproducibility. The latter improvement was brought about by two ongoing projects: improving the mutual consistency of the radioactive decay and photon evaporation databases, and the development of an expanded, hard-coded nuclide table for handling on-demand creation of decay daughters.

3.4 Generic Event Biasing

A new event biasing scheme was introduced in release 10.0 and evolved in 10.1. It allows virtually any type of process-level biasing. A toolkit approach has been adopted in which a small abstract layer defines the interfaces for implementing concrete biasing options. Some typical biasing techniques and several examples have been provided in Geant4 starting from this abstract layer, but the openness of the design allows user-customized biasing techniques to be implemented.

3.4.1 Generic Event Biasing Design

Problems in which biasing techniques are useful often mix several biasing options in one application. Traditionally, decisions on what techniques to apply are taken at the application configuration level. We introduced a mechanism allowing a per-step decision on what biasing technique to be applied, defining two entities:

- G4VBiasingOperator: an abstract class, for decision making on what biasing techniques to be applied in the current step, and which can select:
  - physics biasing techniques, altering the physics process(es) interaction probability, and/or final state production; where these decisions are taken per physics process,
  - a "non-physics" biasing, for techniques which do not alter the behavior of a physics process, like splitting, or Russian roulette, etc.

  A concrete instance of a G4VBiasingOperator is attached to a G4LogicalVolume and will be active only in this volume.
- G4VBiasingOperation: an abstract class to represent any "atomic" biasing technique, like biasing of a process interaction probability, or of its final state production; or "non-physics" biasing, defining where to apply the technique, and what to do then.

A third concrete class, G4BiasingProcessInterface, provides the interface between the tracking and these biasing classes. This G4VProcess checks if the current G4LogicalVolume has a G4VBiasingOperator attached, and if so, messages this operator to get the biasing operations to be applied. A G4BiasingProcessInterface object can hold a physics process, in which case it controls it, and substitutes for the process behavior the ones implemented in the biasing operations. Physics processes not wrapped by a G4BiasingProcessInterface are kept unbiased.

3.4.2 Occurrence Biasing

The biasing of a physics process interaction probability is a problem well enough defined to allow one further level of specification. A physics process has its occurrence probability defined by the exponential law, itself controlled by the process cross section. Occurrence biasing consists in substituting for the analog process interaction law, $p_a(\ell)$, some other arbitrary biasing law, $p_b(\ell)$. Two types of statistical weight have to be computed for each biased process: a weight for non-interaction, w_{NI}, for a track travelling a distance ℓ with no interaction, and a weight for interaction, w_I, for the process which triggers an interaction in the segment $\mathrm{d}\ell$ after ℓ. These weights are multiplicative. w_{NI} is the ratio of the non-interaction probabilities over the flight ℓ for $p_a(\ell)$ and $p_b(\ell)$, and w_I is the ratio of cross sections at ℓ, where, for $p_b(\ell)$, it is defined as an "effective cross section" $\sigma_b(\ell) = -\frac{\mathrm{d}}{\mathrm{d}\ell} \log p_b(\ell)$, so that $p_b(\ell)$ can be recast under the regular $p_b(\ell) = \sigma_b(\ell) \exp\left(-\int_0^\ell \sigma_b(s)\mathrm{d}s\right)$ form.

The abstract class G4VBiasingInteractionLaw defines the interface for interaction laws, and requests the specification of the non-interaction probability and effective cross section. A G4BiasingProcessInterface which has been requested to bias a physics process interaction law then has the necessary ingredients to provide the proper weight computations.

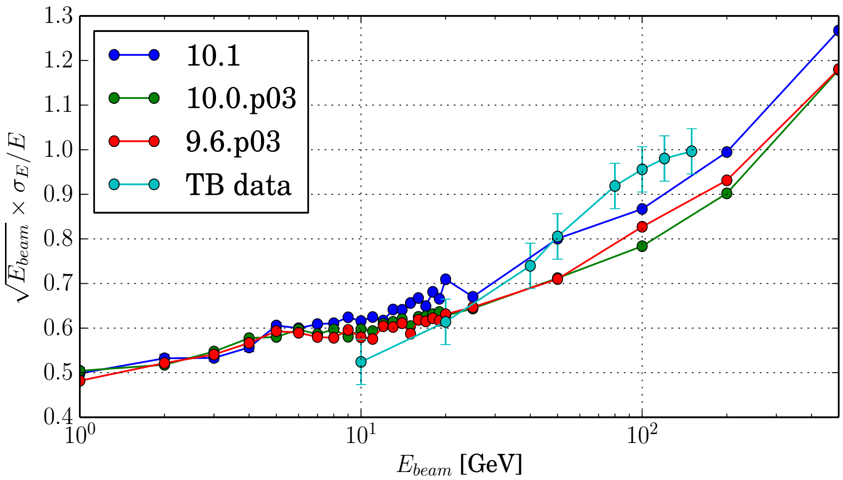

Figure 4: Normalized energy resolution in a copper - liquid argon calorimeter compared to predictions of the FTFP_BERT physics list in releases 9.6 (red curve), 10.0 (green) and 10.1(blue). Incident beam consists of pi-

3.4.3 Currently Provided Biasing Techniques

Cross section biasing can be done using the G4BOptnChangeCrossSection biasing operation. An operator using it will have to specify by what factor the cross section is to be biased. It will choose this factor freely, and can hence make it angle-, or energy-dependent, etc.

An operator, G4BOptrForceCollision, implements a forced collision scheme similar to that of MCNP. This operator makes use of several operations to achieve the necessary logic of the track interacting in the volume, under an exponential interaction law which is limited to the volume extension, and a copy of the track flying freely to the other boundary of the volume, with the proper weight. Up to now, these schemes have been validated only with neutral particles.

Examples are also provided, showing how to use above options, as well as how to implement with this generic scheme importance geometry biasing, and bremsstrahlung splitting.

3.5 Calorimeter response

The response of calorimeters in high energy physics experiments provides a critical check that all physics processes are correctly simulated. The main observables to compare to are the energy response, resolution, lateral and longitudinal shower shapes.

Because electromagnetic processes dominate shower evolution, significant efforts have been made to improve the overall description of EM shower shapes: the bremsstrahlung and multiple scattering descriptions were reviewed and improved, having been identified as key components in defining EM shower shapes. Calorimeters are sensitive to the precise simulation of electron and gamma transport in the MeV energy region. Therefore extensive validation and benchmarking is being carried out for medium and low energy electrons and gammas. The general agreement of Geant4 predictions with experimental data collected at the LHC is better than one percent (Abat et al. 2011).

While hadronic processes are a relatively small part of the shower, they are important for the description of lateral shower shapes, energy resolution and time structure. Extensive validation of Geant4 physics against LHC test-beam calorimeter data has shown that the most critical models for the description of the hadronic part of a shower are parton string models, such as Fritiof (Andersson et al. 1987) at high energy, cascade models at intermediate energies, such as the Bertini intra-nuclear cascade (Heikkinen et al. 2003), and pre-equilibrium and evaporation models at low energies, such as G4Precompound (Quesada et al. 2010).

For the calorimeter response function the agreement between simulation and data for hadron-induced

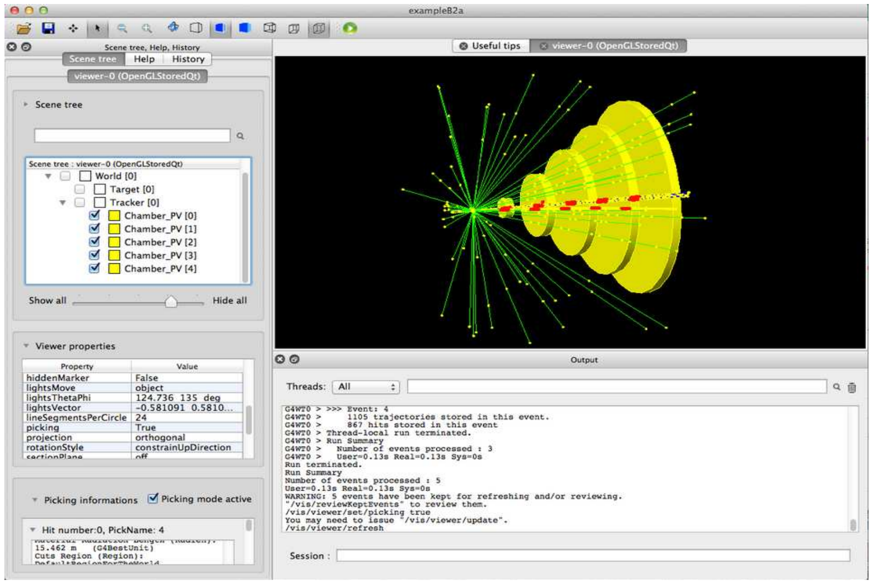

Figure 5: Screen shot of OpenGL viewer using Qt.

showers is at the level of a few percent, but shower shapes and resolution are less precisely described and show an agreement at the level of 10-20%. As mentioned in the hadronic physics section, lateral shower profiles have been affected by bugs in the Fritiof parton string model, leading to a better agreement with data than warranted by the model. With the bugs fixed and subsequent model re-tuning in release 10.1, agreement with the lateral shower shape data has returned to the 10-15% level.

One result of the Fritiof model retuning has been a significant improvement in the agreement with energy resolution data. In the past, and up through release 10.0, the FTFP_BERT physics list has always produced an energy resolution which was 10 - 20% too small when compared to data. The recent bug-fix and subsequent re-tuning of FTF has resulted in better performance for some types of calorimeter. The improvement is clearly evident for medium and light nuclear targets as shown by the copper-liquid argon (Cu-LAr) calorimeter results in Fig. 4. For heavy targets like W or Pb, the effect is much less pronounced.

The high precision neutron model mentioned earlier is not typically included in physics lists for high energy physics experiments, as it is time-consuming and does not play an important role in describing the energy response or shower shapes. However, it is important for other calorimetric observables such as the time structure of the showers and lateral shower development in neutron-rich materials. For these applications the FTFP_BERT_HP and QGSP_BERT_HP physics lists are available, and have shown good agreement with data.

4 Visualization and GUI

The GEANT4 visualization system (Allison et al. 2013) is a multi-driver graphics system designed to serve the GEANT4 Simulation Toolkit. It is aimed at the visualization of GEANT4 data, primarily detector descriptions and simulated particle trajectories and hits. It can handle a variety of graphical technologies simultaneously and interchangeably, allowing the user to choose the most appropriate visual representation. The Geant4 graphics interface was developed to support multiple graphics drivers over several complementary graphics technologies to satisfy a wide variety of users' needs. The current distribution of Geant4 contains, at last count, 14 drivers of various sorts. Those which need external libraries or packages may only be activated at the compilation step if the corresponding external system is installed.

In principle, users may extend this list by implementing their own driver to the specification of the abstract interface. A user may draw to the basic abstract interfaces, either in C++ code or, more usually, via visualization commands through a user interface, and expect it to be rendered in one of a number of different ways: to a computer screen (graphic drivers) or to a file for subsequent browsing.

The workhorse of the Geant4 visualization system is the set of OpenGL drivers. In particular, recent work has focused on our implementation of OpenGL within the context of the popular platform-independent GUI tookit, Qt. The Geant4 Qt visualization/GUI solution allows the user to rotate and zoom the view, to toggle visibility of various viewed objects, and to interrogate detailed information about geometry, trajectories and hits. A sample window is shown in Fig. 5. The same interface allows the user to control the overall simulation, even customizing the GUI with the user's own action buttons.

The most recent Qt driver developments include the porting of OpenGL code based on "display lists" to a more recent version based on vertex buffers, and the addition of a new graphical component to enable the browsing and editing of all viewer properties online. Other improvements, based on the recent multithreading developments, include the handling of output from the Qt driver GUI and improvements in picking mode.

5 Analysis

Analysis tools based on AIDA (AIDA 2012) have been used in Geant4 examples since release 3.0 (December 2000). No analysis code was provided by Geant4 until 2010. The use of AIDA required the installation and use of an external AIDA-compliant tool that was not always straightforward and easy, especially for novice users.

The new analysis category (Hrivnacova 2014) based on g4tools (g4tools 2014) was added in Geant4 9.5 with the aim to provide the users a "light" analysis tool available directly with Geant4 installation without the need to link their Geant4 application to an external analysis package. The migration of Geant4 examples to new analysis tools was accomplished in the 10.1 release. Just one example with a pure AIDA-based analysis has been kept in the examples category (*extended/analysis*, together with g4tools-based and ROOT-based analysis codes implemented for the same physics scenario.

The analysis category is multithreading-capable; no changes are required in the user client analysis code for migration to multithreading mode. g4tools classes have proven to be thread-safe and no issues with concurrent processing were observed in Geant4 tests.

5.1 g4tools

g4tools, included directly in Geant4 provides code to write histograms and ntuples in several formats: ROOT, XML AIDA format, and CSV (comma-separated values format) and HBOOK. It is part of the inlib and exlib (g4tools 2014) libraries, that also include other facilities like fitting and plotting.

5.2 Analysis Classes

The analysis classes provide a uniform, user-friendly interface to g4tools and hide the differences according to a selected output technology from the user. They take care of the higher-level management of the g4tools objects (files, histograms, profiles and ntuples), handle allocation and removal of the objects in memory and provide the access methods to them via indexes. The possibility to activate or deactivate selected histograms was also implemented at the request of users.

The analysis classes are fully integrated in the Geant4 framework: they follow the Geant4 coding style and also implement the built-in Geant4 user interface commands that can be used to define or configure analysis objects. The CLHEP system of units, used in Geant4, is also integrated. Users may select a unit which will be applied automatically to the histogram values, and will also appear in the histogram axis title. Manager classes implement the common interfaces and specific access functions (with a return type specific to the output format) via the classes per analysis object type (see Fig. 6).

Since Geant4 version 10.1, the analysis category was extended with reader classes which allow g4analysis objects to be read in from the files generated by the analysis manager(s) during processing in a Geant4 application.

Writing of HBOOK output files is possible with the use of the ExG4HbookAnalysisManager class which is provided in a dedicated extended example together with all necessary configuration files to build it with the CERNLIB libraries.

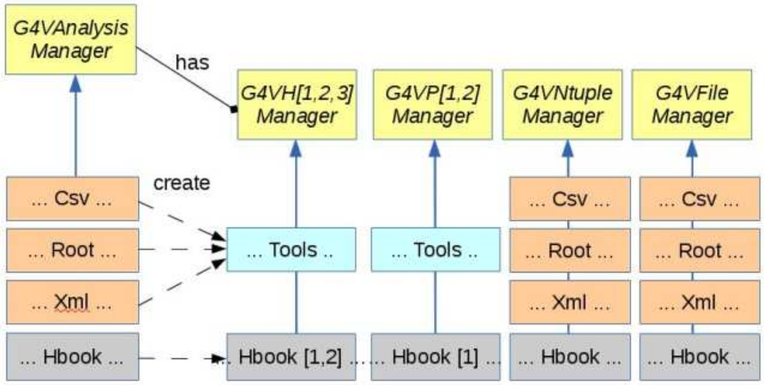

Figure 6: Analysis managers class design. The G4AnalysisManager, seen by the user as a single interface class, can point to one of four output type specific manager classes (Csv, Xml, Root, Hbook) implemented via the classes per analysis object type: [1,2,3] - dimensional histograms, [1,2] - dimensional profiles, an ntuple and a file.

6 Summary and Prospects

The Geant4 version 10 series of releases has introduced extensive changes in its software engineering, physics and geometry modeling, and analysis and visualization tools. Efforts were made throughout the toolkit to increase computing efficiency, physics quality and ease of use. The success of these efforts is borne out by excellent scaling of multithreaded running, improvements of memory efficiency and speed, and better reproduction of physics results from large high energy physics experiemts and users in the space and medical communities.

Geant4 recently celebrated its 20th year of development. It has experienced several global architectural evolutions and it is still steadily evolving. The latest evolution was Geant4 version 10.0 released in December 2013. This made Geant4 the first fully multithreaded large-scale software package in the simulation domain. In addition, version 10.1 offers up to 10% speedup with less than half the memory space per thread and better physics performance compared to version 10.0. Given Geant4 is nowadays mission-critical for many users including all LHC experiments, space missions, medical applications, material science and industrial applications, Geant4 will be maintained and further developed for at least the next decade. Since users prefer stable application programming interfaces (APIs), Geant4 continued with the version 9 series for seven years, and plans to conitnue with the current 10 series for as long as possible. During this period, refinements for better computing and physics performance and usability will be provided without major API changes. Looking forward, the Geant4 collaboration has already begun the next round of architectural revisions, for example full migration to C++11 and C++14, and algorithm-level vectorization.

References

Abat, E. et al., 2011. "Photon reconstruction in the ATLAS Inner Detector and Liquid Argon Barrel Calorimeter at the 2004 Combined Test Beam", Journal of Instrumentation 6-04, P04001.

Agostinelli, S. et al., 2003. Nucl. Instr. Meth. A 506, 250.

Ahn, S. et al., 2013, SNA + MC 2013 - Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo 04213 (2014)

AIDA, 2012. http://cern.ch/aida/.

Allison, J. et al., 2006. IEEE Trans. Nucl. Sci. 53 No. 1, 270.

Allison, J. et al. 2012. "Geant4 electromagnetic physics for high statistics simulation of LHC experiments", J. Phys: Conf. Ser. 396, 022013.

Allison, J., Garnier, L., Kimura, A., Perl, J., 2013. International Journal of Modeling, Simulation, and Scientific Computing 4, Suppl. 1 1340001.

Amako K. et al., 2005, indent IEEE Trans. Nucl. Sci. 52 Issue 4, 910
Andersson, B. et al., 1987, Nuclear Physics B 281-1/2, 289.
Andreev, V.F. et al., Nuovo Cim. 103, 1163 (1989).
Apostolakis, J. et al., 2009, Rad. Phys. Chem. 78 Issue 10, 859.
Apostolakis et al., 2010. "Validation and verification of Geant4 standard electromagnetic physics", J. Phys: Conf. Ser. 219 Part 3, 032044.
Asai, M. et al., 2014, "Recent developments in Geant4", Ann. Nucl. Ener. , in press, doi:10.1016/j.anucene.2014.08.021
Bogdanov, A. G. et al., 2006. IEEE Trans. Nucl. Sci. 53, 513.
Dong, X. et al., 2010. "Multithreaded Geant4: Semi-automatic Transformation into Scalable Thread-Parallel Software", Euro-Par 2010 - Parallel Processing 6272, 287.
Dong, X. et al., 2012. "Creating and Improving Multi-Threaded Geant4", J. Phys.: Conf. Ser. 396, 052029.
g4tools, 2014. http://inexlib.lal.in2p3.fr and Barrand, G., 2014. J. Physics Conf. Series 513 022002.
Garnier, L., "Qt driver tutorial", http://geant4.in2p3.fr/spip.php?article60&lang=en.
Gayer, M. et al, 2012. "New Software Library of Geometrical Primitives for Modeling of Solids used in Monte Carlo Detector Simulations", J. Phys.: Conf. Ser. 396 Part 5, 052035.
Heikkinen, A., Stepanov, N., Wellisch, J.P., 2003. "Bertini intra-nuclear cascade implementation in Geant4", Computing in High Energy Physics 2003, La Jolla, California, USA.
Hrivnacova, I., 2014. "Analysis Tools in Geant4", J. Physics. Conf. Series 513, 022014.
Incerti, S. et al., 2010. Medical Physics 37, 4692.
Ivanchenko, V. N., Kadri, O., Maire, M., Urban, L., 2010. "Geant4 models for simulation of multiple scattering", J. Phys., Conf. Ser. 219 Part 3, 032045.
Ivanchenko, V. N. et al., 2011. "Recent improvements in Geant4 electromagnetic physics models and interfaces", Proceedings of SNA + MC2010: Joint international conference on supercomputing in nuclear applications + Monte Carlo 2010 Tokyo
Ivanchenko, V. N. et al., 2014. "Geant4 Electromagnetic Physics for LHC Upgrade J. Phys., Conf. Ser. 513 Track 2, 022015.
Karamitros, M. et al., 2014, J. Comput. Phys. 274, 841
Kimura, A., "gMocren - a volume visualizer for Geant4 medical simulation", http://geant4.kek.jp/gMocren.
Mancusi, D. et al., Phys. Rev. C , 2015, in press.
"Evaluated Nuclear Structure Data Files".
Paltani, S., University of Geneva, Switzerland, 2014.
Pandola L. et al., 2014, arXiv:1410.2002 (physics.med-ph)
Perl, J., "HepRep: a Generic Interface Definition for HEP Event Display Representables", http://www.slac.stanford.edu/~perl/heprep.
Piparo, D., Innocente, V., Hauth, T., 2014. "Speeding up HEP experiment software with library of fast and auto-vectorisable mathematical functions", J. Phys., Conf. Ser. 513, 022013.
Quesada, J.M. et al., 2010. "Recent Developments in Pre-Equilibrium and De-Excitation Models in Geant4", Proceedings of SNA + MC2010: Joint international conference on supercomputing in nuclear applications + Monte Carlo 2010 Tokyo
SAID, Scattering Analysis Interactive Dial-up, George Washington University, 2013.
Schaelicke, A. et. al., 2011. "Geant4 electromagnetic physics for the LHC and other HEP applications", J. Phys.: Conf. Ser. 331 Part 3, 032029.
Wright, D.H. and Kelsey, M., to be submitted 2015.