

GEANT4 10.6 highlights

kernel modules

Gabriele Cosmo, CERN EP-SFT
for the [Geant4 Collaboration](#)



Outline

- Features and fixes introduced in release 10.6
 - Kernel modules
 - EM Physics (see talk by V.Ivantchenko)
 - Hadronic physics (see talk by A.Ribon)

➤ *Detailed release notes:*

- <http://cern.ch/geant4-data/ReleaseNotes/ReleaseNotes4.10.6.html>

➤ *List of planned features for 2019:*

- http://cern.ch/geant4/support/planned_features

Geometry

Geometrical primitives

- Updated VecGeom library, VecGeom v1.1.5
 - Selection for enabling use made at configuration
 - <https://gitlab.cern.ch/VecGeom/VecGeom/tree/v01.01.05>
 - Introduced generation of polyhedral meshes for all shapes
 - First implementation of dedicated GDML reader for persistency
 - Added new missing shapes (ellipsoid, elliptical-tube, elliptical-cone)
 - All shapes (excepts for twisted) now available for replacement
- Revised algorithms for overlaps checking
 - Speedup and improved diagnostics

Geometry

Navigation, Field

- Added hooks for enabling partial or complete replacement of navigation algorithms in Geant4
 - Allowing interfacing with Flair/Moira for use of Fluka geometries with Geant4
 - First prototype implementation of a navigator based on VecGeom
- Reviewed treatment of looping particles in field propagation
 - Enhanced diagnostics and settings, fully under user control
- Enabled default use of interpolation for intersection calculation in field propagation
 - Selection between the new interpolation-capable integration scheme, for shorter steps, and a helix-based scheme chosen for steps larger than 2π times the curvature radius at the initial location
- C++11 revision of geometry code

Global, Biasing, UI

- Global:
 - Reviewed and optimised G4PhysicsVector (added additional Value() method option to propagate down the known log-energy value), avoiding log() calls when log-vector is used and providing CPU run-time boost
 - Added functionality in G4DynamicParticle to provide log-kinetic energy value, computed only on demand; reviewed EM processes to select the target atom by making use of the already known log-energy value in the log-vector access. Providing additional measurable CPU speedup
- Generic processes:
 - Introduced ability to perform leading particle biasing
- UI commands:
 - Enhanced UI commands to now return proper return code when they are not successful

Materials, Analysis & Digits/Hits

- Materials:
 - New class G4DensityEffectCalculator, providing on-the-fly computation of the density effect correction using "exact" formulas
- Analysis:
 - Added new factory function, allowing to create the analysis manager of the type selected via a string argument and so to choose the output type at run time
 - Switched the default Root n-tuple merge mode to new column-wise mode with preserving rows and with possibility to override the default
- Digits/Hits:
 - Added G4TScoreNtupleWriter and G4TScoreNtupleWriterMessenger classes, implementing ability to store hits collections (G4THitsMap) with Geant4 analysis tools

Examples

- New example *GB07* (in extended/biasing)
 - Demonstrating the usage of the new leading particle biasing functionality, for generic biasing
- New example *Hadr08* (in extended/hadronic)
 - Demonstrating the possibility of emulating hadronic-model-per-region capability (not foreseen by the hadronic framework) via the use of generic biasing with usual, unbiased weights of 1.0
- Extended GFlash examples
 - New example *gflash2* with envelope in parallel world
 - New example *gflash3* with sensitive-detector in parallel world
 - New example *gflasha* with histogramming of shower profiles
- New DNA examples
 - *dnadamage1*: DNA simulation of damage on a chromatin fiber
 - *microprox*: DNA computation of proximity functions in liquid water

Visualization/Interfaces & Data sets

- Visualization & Interfaces:
 - New set of UI commands for view interpolation and centering in visualization
 - New additional cloud drawing style in visualization
 - Using kernel algorithms for generating points on the surface of volumes
 - By-passing polyhedral representations
 - Solid being visualised by a polymarker of dots
- Data sets:
 - New versions: `G4EMLOW-7.9`, `G4PARTICLEXS-2.1`, `G4NDL-4.6`, `G4RadioactiveDecay-5.4`, `G4PhotonEvaporation-5.5`

Configuration & Externals

- Cmake:
 - CMake v3.8 or higher now required and enforced
 - Pre-processor flags are promoted to fixed `#define` statements in a generated header. User code relying on these macros should include the `G4Types.hh` header to make them available
 - Preprocessor macros `G4UI_USE` and `G4VIS_USE` are removed
 - Deprecated support for Qt4 for UI/Visualization
- CLHEP:
 - Version 2.4.1.3 required

Platforms for 10.6

- Linux, gcc-4.9.3, 5.4.X, 6.3.X, 7.4.X, 8.3.X, 9.2.X 64 bits
- MacOSX 10.15, llvm/clang-8.0 (XCode 11.x), 64 bits
- Windows 10, Visual C++ 14.23 (Visual Studio 2019)
- Also tested (sequential/MT):
 - Linux CentOS7, icc-19, clang-5.0/8.0
 - Linux Ubuntu 18, gcc-7.4
 - MacOSX 10.13/10.14, clang-7.0

Thanks!