Geant4 11.0 highlights
kernel modules

Gabriele Cosmo, CERN EP-SFT
for the Geant4 Collaboration

Outline

- Features and fixes introduced in release 11.0
  - Kernel modules
  - EM Physics (see talk by V.Ivantchenko)
  - Hadronic physics (see talk by A.Ribon)

➢ *Detailed release & patches notes:*

- http://cern.ch/geant4-data/ReleaseNotes/ReleaseNotes.11.0.html
- https://cern.ch/geant4-data/ReleaseNotes/Patch4.10.7-2.txt
- https://cern.ch/geant4-data/ReleaseNotes/Patch4.10.7-3.txt

➢ *List of planned features for 2021:*

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

Geometry

Geometrical primitives

- Updated VecGeom library, VecGeom v1.1.18
  - Selection for enabling use made at configuration
    - https://gitlab.cern.ch/VecGeom/VecGeom/tree/v01.01.18
  - New ability to run in single-precision on-demand
  - New navigation algorithm with BVH acceleration (for both CPU and GPU)
  - New CUDA manager: speedup bulk copy CPU<>GPU of volumes and transforms
  - Extended GDML reader supporting auxiliary information
  - Simplified/unified use of C++ compiler flags; C++17; modernised CMake
  - Bug fixes
- New example enabling VecGeom navigation in Geant4 (***β*** version)
  - examples/extended/geometry/vecGeomNavigation

Geometry

Navigation, Volumes, Transportation

- Added map for faster search based on name in solids, volumes and regions stores
  - Pointers to elements are stored in the map as buckets, grouping elements with same name
- Enabled use of alternative G4VoxelNavigation class in G4Navigator for voxelised volumes
- Enabled use of G4Allocator to dynamically allocate nodes and proxies for the voxels optimisation structure
  - Helping in reducing memory fragmentation
- Make G4PVDivision and G4ReplicatedSlice inherit from G4PVReplica rather than G4VPhysicalVolume
  - Allowing proper cloning in MT mode

10.7.p03

Materials, Analysis & Digits/Hits

- Materials:
  - Revised and updated G4Exceptions in G4Material, now providing more complete information and better diagnostics
  - New class G4OpticalMaterialProperties, a store for optical material properties and allow use of predefined optical material properties
- Analysis:
  - New functions in G4VAnalysisManager allowing resetting and deleting all allocated analysis objects and clearing their collections
  - Added support for n-tuple columns of string vectors
  - Implemented support for file system directories with Csv output
  - Major revision in analysis classes. Migration to G4ThreadLocalSingleton in all specific analysis manager and reader classes
- Digits/Hits:
  - Extended cylindrical scoring mesh to tube and tube segment

Run, Tracking, Global

- Run:
  - Tasking system, based on PTL (Parallel Tasking Library) v2.0.0 now set as default parallelism scheme for multi-threading
  - New G4SteppingVerboseWithUnits class, an alternative to G4SteppingVerbose with printout of proper units. Simplified way of defining a user-specific stepping verbose, that is now common to all sequential, MT and tasking modes
- Tracking:
  - New G4VTrackingManager class, an interface for custom tracking managers specialized for one or a small number of particle types.
- Global:
  - Simplified G4String interfaces and implementation with C++11/17; new G4StrUtil namespace with functions for simple string manipulation
  - Major update the G4PhysicsVector and related classes; new public access methods; removed obsolete unused methods; introduced possibility to use different Spline methods

Visualization/Interfaces

- New ToolsSG (TSG) visualisation package based on g4tools, providing four new (mutually exclusive) visualisation drivers: TOOLSSG_X11_GLES, TOOLSSG_WINDOWS_GLES, TOOLSSG_XT_GLES and TOOLSSG_QT_GLE
  - Allows for in-app plotting with visualization and related UI command
- New VtkQt and VtkNative visualisation drivers
- Added G4Mesh, a light class that encapsulates and validates visualisation of a nested parameterisation
- Extended G4UIWin32 driver with new features
- Revised all visualisation models, to have G4PhysicalVolumeModel taking care of the transformation of any primitives that it generates
- Removed obsolete network visualisation options for VRML driver and FukuiRenderer driver
- Removed obsolete GAG/Gain/MOMO and HepRep(WIRED) modules/driver
- Removed deprecated functions in visualization
- Qt drivers require the Qt-5 platform. Qt-6 is not yet supported

Data sets

- New data set versions:
  - **G4EMLOW-8.0, G4PARTICLEXS-4.0**
- In order to use ParticleHP for charged particles (protons, deuterons, tritons, He3 and alphas), an optional data set is required, and can be optionally downloaded in addition:
  - **G4TENDL-1.4**

Configuration & Externals

- Cmake:
  - C++17 is the minimum ISO C++ Standard required to compile Geant4 and all applications using it
  - Multithreading support is now enabled in the build of Geant4 by default
  - CMake 3.16 is the minimum version required to build Geant4
- CLHEP - Version 2.4.5.1 required
  - Added units: minute, hour, day, year and millielectronvolt
  - Added constants: Bohr_magneton and nuclear_magneton
  - New random engine RANLUX++ and related helper classes: an extension of RANLUX with much higher luxury level at better performance
  - Fixed seeding in Ranlux64Engine

Extented examples

- **pythia/py8decayer** - demonstrating how to outfit Pythia8-based decay features to resonances in Geant4 where decay tables are not implemented by default. Showing also how to replace existing Geant4 decay tables to such resonances as tau+/- or B+/- with the Pythia8-based ones
- **geometry/VecGeomNavigation** - demonstrating integration of VecGeom navigation, based on the prototype package G4VecGeomNav, being integrated in the example at build time
- **hadronic/Hadr05** - demonstrating how to collect energy deposition in a sampling calorimeter and how to survey energy flow
- **hadronic/Hadr10** - exercising the decay of tau leptons, as well as charmed and bottom hadrons
- **parameterisations/Par04** - demonstrating how to use the Machine Learning (ML) inference to create energy deposits as a fast simulation model using ONNX runtime and LWTNN libraries
- **runAndEvent/RE07** - demonstrating how to register specialized tracking managers for a particle or a set of particles
- **dna/AuNP** - simulating the track-structure of electrons in microscopic gold volume. Also simulating that in liquid water medium surrounding the gold volume
- **dna/scavenger** - showing how to activate the scavenging process in chemistry using the deterministic treatment of the IRT model
- **visualization/movies** - New example illustrating how to make a movie with UI commands

Advanced examples

- **CaTS** - New application implementing a flexible and extendable framework for the simulation of calorimeter and tracking detectors. Also demonstrating how to use Opticks for the creation and propagation of optical photons

Platforms for 11.0

- Linux CentOS8/Stream
  - gcc-8.3.1 to 11.2, 64 bits (Intel or AMD)
- macOS 12.1 Monterey
  - Apple Clang-13 (XCode 13.x), 64 bits (Intel or Apple Silicon)
- Windows 10
  - Visual C++ 14.29 (Visual Studio 2019)

❖ Also tested (sequential/MT):
  - Linux CentOS7, icc-2021, clang-9/10/11
  - Linux Ubuntu 20, gcc-9.3
  - macOS 10.15 Catalina, Apple Clang-12
  - macOS 11.6 Big Sur, Apple Clang-12

Thanks!