GEANT4 11.3 highlights

kernel modules

Gabriele Cosmo, CERN EP-SFT

for the Geant4 Collaboration

Outline

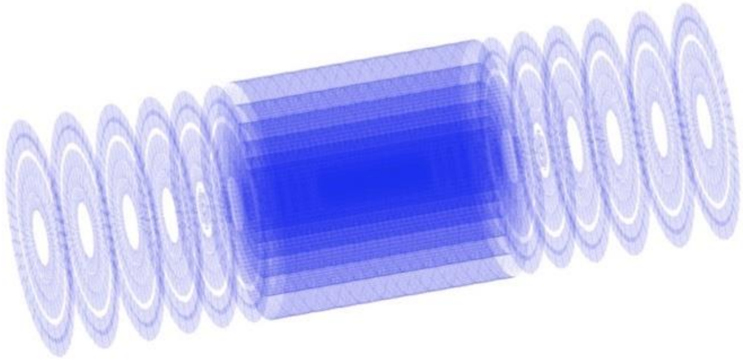
- Features and fixes introduced in release 11.3
  - Kernel modules
  - EM Physics (see talk by V.Ivantchenko)
  - Hadronic physics (see talk by A.Ribon)

- *Detailed release notes:*
  - **https://cern.ch/geant4-data/ReleaseNotes/ReleaseNotes.11.3.html**

- *List of planned features for 2024:*
  - **https://cern.ch/geant4/planned-features-2024**

Geometry

VecGeom

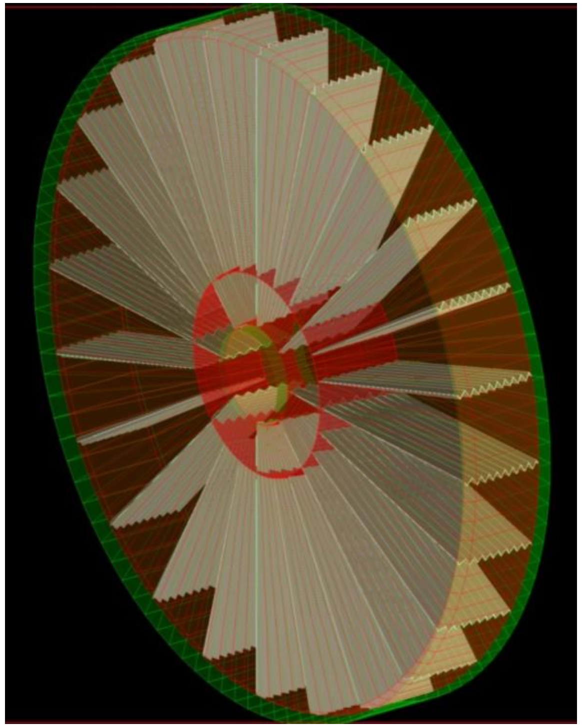


- Updated VecGeom library, VecGeom v1.2.10
  - Selection for enabling use made at configuration
    - https://gitlab.cern.ch/VecGeom/VecGeom/tree/v01.02.10
  - Added logger classes for warnings and printouts (contributed by Celeritas)
  - Bug fixes to Polycone, Trapezoid and Torus shapes
  - Fixes in CI settings and porting
- New 2.0.0 release candidate, VecGeom v2.0.0-rc3
    - https://gitlab.cern.ch/VecGeom/VecGeom/tree/v2.0.0-rc3
  - Simplified code; removed unused interfaces and backends (vector API, unused specialisations)
  - Targeting final release v2.0.0 with portable code across devices
- Surface Model geometry development
  - All common solids now supported; added BVH acceleration support; overlaps handling; mixed precision; actively working on optimisation

Geometry

Navigation, Volumes, Solids, Magnetic Field

- Geometry optimisation using multithreading
  - New option, reducing time spent for voxelisation over logical volumes of complex geometry setups by adopting multithreading/tasking
- Added new classes for field configuration
  - New UI commands to allow users to control parameters for field transportation and the accuracy of intersection of curved trajectories with volumes
- Revised implementation of G4GenericTrap
  - Now used in the implementation of the EMEC structure in ATLAS, under validation and providing ~19% speedup vs. original custom solid
- Improved/optimised implementation of GetCubicVolume() in Boolean solids



Materials, Particles & Analysis

- Materials:
  - Extended diagnostics and output from material scanner
- Particles:
  - Added new class, G4ChargedUnknownParticle, similar to G4UnknownParticle for charged unknown particles
  - Part of ATLAS and LHCb request to assign automatically ionisation and multiple scattering processes to charged primary particles with valid PDG code but not known to Geant4
- Analysis:
  - Added support for accumulable collections in analysis module

Run, Parameterisations

- Run:
  - First implementation of task-based sub-event level parallelism: event split in sub-events, with automatic merging of the hits at the end of the event – demonstrated in example runAndEvent/RE03
    - Added sub-event parallel mode as an option in G4RunManagerFactory and new UI command for allowing trajectory merging
    - Useful for splitting of large events, e.g. heavy-ion collisions
    - “Phase I”: all tasks have the same physics processes and see the same detector geometry
  - In “Phase II” (i.e. next year release): each task can have only the necessary physics processes and see limited detector geometry, needed for that particular task
    - Useful for heterogeneous simulation
  - The introduction of sub-event parallelism does not change the threading model, and is optional
- Parameterisations:
  - New model of ionisation losses for e+/- ionisation channeling
  - Added new process G4CoherentPairProduction for the simulation of coherent pair production by high energy gamma in an oriented crystal; using new channeling dataset accessed through G4CHANNELINGDATA environment variable

Visualization/Interfaces

- Qt6 now fully supported
  - Restored OpenGL as default backend for Qt6. Fixed open issues when using Qt6 with supported drivers
- Improved plotting with /vis/plot if TOOLS_USE_FREETYPE flag is set
  - Set GEANT4_USE_FREETYPE=ON in CMake
- Improved identification of scene tree models in visualisation
  - Now suppressing the scene tree if the number of touchables exceeds 10000 (equivalent to about 20 MB)
- Updated visualisation manager to correctly merge/handle asynchronous event merging

Data sets

- New data set versions:
  - **`G4EMLOW-8.6.1, G4PARTICLEXS-4.1, G4ENSDFSTATE-3.0, G4PhotonEvaporation-6.1, G4RadioactiveDecay-6.1.2, G4CHANNELLINGDATA-1.0`**
- New optional data sets:
  - **`G4NUDEXLIB-1.0, G4URRPT-1.1`**

Configuration & Externals

- Configuration/General
  - Support running example build-and-run tests in XCode on macOS and added support for use of sanitizers with XCode
  - Fixed spurious compilation warnings on gcc when LTO settings are enabled
  - Removed configuration/use of no longer supported TiMemory profiling
  - Bump minimum VecGeom version to v1.2.9; recommended v1.2.10 or higher
  - CMake 3.16 is the minimum version required to build Geant4
- PTL
  - Synchronised with version PTL-3.0.1. Retaining patches for Geant4 source layout in configuration

Extended & Advanced examples

- The example **RE03** in **extended/runAndEvent** has been extended to demonstrate the ability to run in sub-event parallel mode
- New **extended/exoticphysics/channeling** examples category: created **ch0** example out of the original example and added new set of examples
  - **ch1**: an easy demonstration of the minimum requirements necessary to integrate the G4ChannelingFastSimModel and the G4BaierKatkov model into a project in order to simulate the physics of channeling and channeling radiation/coherent bremsstrahlung
  - **ch2**: an enhanced version of ch1, providing the user with the full functionality of both demonstrated models, with parameters set up via a macro
  - **ch3**: demonstrating the minimum requirements necessary to integrate the G4CoherentPairProduction process into a project, along with the demonstrated models, to simulate the physics of electromagnetic showers in an oriented crystal
- New **dna** category for Geant4-DNA advanced examples, including:
  - **moleculardna** example, moved from extended category
  - New **cellularPhantom** application, showing how to simulate the irradiation of a 3D voxel phantom containing biological cells, created from a confocal microscopy 24-bit RGB image
  - Updated **dsbandrepair** example, now using EmParameters to control chemistry time step model

Platforms for 11.3

- Linux Alma9/RedHat Linux Enterprise 9
  - gcc-11.5 to 14.2, 64 bits (Intel or AMD)
- macOS 15 Sequoia
  - Apple Clang-16 (XCode 16.x), 64 bits (Intel or Apple Silicon)
- Windows 11
  - Visual C++ 14.4 (Visual Studio 2022), 64 bits

❖ Also tested (sequential/MT, 64 bits):

- Linux CentOS9/alma9, icx-2024.2, LLVM/clang-16/17/19
- Linux Ubuntu 22/24, gcc-11.4/13.2
- macOS 13 Ventura/14 Sonoma, Apple Clang-15
- Windows 10, Visual C++ 14.36 (Visual Studio 2022)

Thanks!