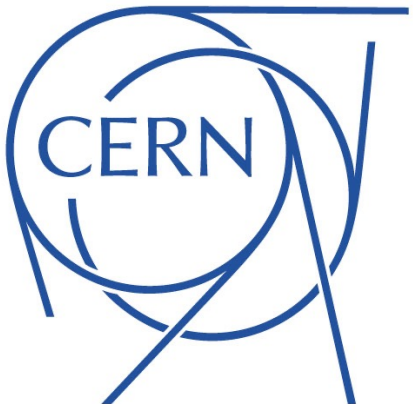


Geometry & Persistency

Recent & ongoing developments

John Apostolakis, CERN EP/SFT

for the Geant4 Geometry & Transportation Working Group



Contents

- Development and fixes in the last year
 - Introduced in release 11.0 and patches
- Features expected in Geant4 11.1
 - Currently under development and scheduled for inclusion in the next release

VecGeom update

Improvements in recent releases (up to release 1.2 – used by Geant4 11.1-beta)

Ongoing development

Current VecGeom version (1.2)

- Navigation
 - BVH: improved, sped up construction (and allows alternative tree building)
 - Navigator use **precision-dependent push**
 - Implementation of further functions for Geant4 navigation
- Configuration improvements
 - Config: Defaults to C++ 17, and chooses to use VecCore 0.8 (by default)
 - Modernisation of CMake target usage
 - Revised settings flags, adding VECGEOM_ prefix
- GPU related
 - GPU initialization optimizations in using/extending 'bulk copy' methods

VecGeom: ongoing development

- Investigation of 'Boundary represented' solids targeted for GPU
 - First steps in bounded surfaces approach (similar to ACTS 'detray') (Andrei, 4A)
 - unbounded surfaces approach based on Orange (Seth, 4A)
 - See the talks in the parallel session 4A
- Upcoming fixes
 - Precision mis-match in Cone/Polycone fix for [ATLAS stuck track issue](#)
 - Improved use of memory allocation for PolyCone/PolyGon parameters (in copying to GPU)
- Further enhancements to enable interim use of existing 'solids' approach for complex experiment geometries
 - Memory allocation improvement (as needed)
 - Simple 'two-plane' extruded solids (without Vector acceleration structures)

Fixes and improvements in release 11.0 & patches

Fixes in 10.7p03 and release 11.0

Volumes

- Added map for fast search based on name in solid, volume and region stores
 - Pointers to elements are stored in the map as buckets, grouping elements with same name
- Use G4Allocator to allocate nodes and proxies used for voxels optimisation
 - To reduce memory fragmentation
- Change G4PVDivision and G4ReplicatedSlice to inherit from G4PVReplica rather than G4VPhysicalVolume
 - Enables correct cloning in MT mode

10.7.p03

Navigation

- Introduced alternative G4VoxelNavigation class that can be enabled for all voxelised volumes in G4Navigator
 - Allows the use of VecGeom Navigation (beta)

Fixes in 11.0 patches

- Solids/Boolean:

- G4MultiUnion(11.0.p01), adds alternative signature for AddNode() with pointer to solid. Added 'const' qualification to transformation passed as argument. Addresses PR [#2457](#)
- G4UnionSolid (p02): add surface tolerance in Inside(p) check on Z. Minor optimisations.

- navigation:

- G4RegularNavigation: reset the zero step counter when a non-zero step is performed, to avoid aborted events. Based on [GitHub PR#38](#).

- management:

- Stores (Solid,Logical/Physical Volume): (optionally) extend getter methods to return last-found object (Part of 11.0.p02)
- Added protection in G4GeometryManager for Open/CloseGeometry() to ensure it is executed only by master thread. Addressing problem report [#2502](#). (Part of 11.0.p03)

Developments in 2022

Planned items for this year

VecGeom evolution and development

- Further adaptations, optimizations
 - Improved CUDA support and porting of solids that now require SIMD backend
 - Adoption of Gitlab CI & extended platforms support
 - Implementation of missing shapes/entities in GDML reader
- Prototype study on surface bounded volumes
 - for adaptation on GPUs
- Further items / details:
 - Support for single precision in data structures and navigation API
 - Handling of construction and run-time errors
 - Code simplification, removal of unused API/backends

Geometry & Transportation

- Validation/consolidation of interface with navigator based on VecGeom
 - Improve robustness of current interface/adaptor to VecGeom navigator, testing & code integration
- Separate safety computation and its state from navigator
 - Loose coupling of navigator in computation of safety distances from geometrical boundaries
- Prototype navigation indexing class
 - An integer index identifying touchables & associated transportation process
- Alternative BVH navigator and optimization structure
 - Navigation based on Bounded Volume Hierarchy (BVH) technique, either natively in Geant4 or through VecGeom

Magnetic Field

- Addition of QSS integration methods (Quantized State Simulation)
 - Alternative integration method which creates adapted polynomials and evaluates the limit of their validity
 - See update in [parallel 4A](#) (Thursday 09:00 CEST)
- Symplectic methods for field integrators (GSoC HSF project)
 - Second order Boris method implemented and tested
 - Fourth order 'improved' Boris method in development
 - See Divyansh's overview talk in [parallel 4A](#) (Thursday 09:00 CEST)
- Investigating [problem 2505](#) reported by CMS in MultiLevelLocator / ChordFinder
 - Restricting 'epsilon' integration accuracy – at least for Dormand Prince $\epsilon_{\max} < 0.001$
- Review accuracy of boundary crossing in field (ALICE and CMS requirement)

Agenda of Geometry Parallel 4A (Thursday)

09:00	Navigation Engines for GPUs <i>TD1</i>	<i>Andrei Gheata et al.</i> 09:00 - 09:35
	Symplectic integration - GSoC project update <i>TD1</i>	<i>divyansh tiwari</i> 09:35 - 09:50
10:00	Integration of QSS in Geant4 - an update <i>TD1</i>	<i>Lucio Santi et al.</i> 09:50 - 10:05
	BVH Navigation - integration in Geant4 <i>TD1</i>	<i>Dr Guilherme Amadio</i> 10:05 - 10:15
	Computation of cubic volume and surface area <i>TD1</i>	<i>Evgueni Tcherniaev</i> 10:15 - 10:25
	Review of open tickets <i>TD1</i>	10:25 - 10:30

Further details

Current VecGeom version

- Version v1.2 of VecGeom containing all latest new features and fixes
 - Config: Defaults to C++ 17
 - Revised settings flags, adding VECGEOM_ prefix
 - Can be used with latest Geant4 11.1-beta release
- Version v1.1.20 (30 March 2022) provided
 - Fix for CutTubs issue causing stuck tracks ([VECGEOM-540](#))
 - BVH navigation: restructure to allow alternative methods to construct tree
 - BVH: improved handling of concentric objects and sped up construction
 - GPU initialization optimisations – in ‘bulk copy’ methods:
 - Activate bulk copy of UnplacedVolumes in CudaManager
 - Added method for UnplacedPolyhedron and UnplacedPolycone.
- Version v1.1.19 (11 March 2022)
- VecGeom primitives can be transparently built through Geant4
 - Original APIs preserved

Current VecGeom version

- Version v1.1.19 (11 March 2022)
 - Modernisation of CMake target usage
 - Use C++11 in-class initialisers for AOS, SOA and Array classes.
 - [VECGEOM-597] Fixed uniformity issue in SamplePointOnSurface() for Tube.
- Version v1.1.18 (5 November 2021)
 - Moved included VecCore to 0.8
 - Implementation of further functions for Geant4 navigation
 - Use **precision-dependent push**
 - Improvement for single-precision mode
 - Improved cone performance (suppressed redundant surface checks)
- VecGeom primitives can be transparently built through Geant4
 - Original APIs preserved

VecGeom: upcoming fixes (master)

- Precision mis-match in Cone/Polycone fix for [ATLAS stuck track issue](#)
- Improved use of memory allocation for PolyCone/PolyGon parameters (in copying to GPU)

Further fixes (10.7-patch 03 – Nov 2021)

- divisions:
 - G4PVDivision & G4ReplicatedSlice inherit from G4PVReplica (not G4VPhysicalVolume), to fix cloning in MT mode.
 - Fixed bug in G4ParameterisationTrd for positioning calculation.
 - Provide operator<<() as a free function for G4UAdapter to support Geant4Py.
- navigation:
 - G4PhantomParameterisation: reduced warning messages in GetReplicaNo() function; issue warning only when difference is bigger than 'kCarTolerance'. Addressing problem report #2314.
- solids/Boolean:
 - In G4BooleanSolid, use G4RecursiveMutex in place of G4Mutex in GetPolyhedron() to avoid potential deadlocks in visualisation in recursive Boolean operations.
- solids/specific:
 - More accurate calculation of distance of point to triangle (G4TriangularFacet). Addressed prob. report #2401.
 - Added missing accessor in G4UTet.
- volumes:
 - Use same strategy for cloning solids for replicated volumes types in G4GeometryWorkspace (required for having proper treatment of divided volumes). Cleanup of unused verbose printouts.
 - G4PVReplica: Avoid deletion of rotation matrix for replication in Phi in destructor, as it is not necessary.

Summary

- Significant progress in VecGeom in several different areas
 - Build system, CUDA support, persistency, navigation, single-precision
 - Important evolutions expected soon in design and implementation to allow for enhanced support on GPU
 - First prototype of VecGeom navigator use in Geant4 provided in example
- Keeping improving and evolving the Geant4 geometry modeler
 - Faster lookup for solids and volumes in stores
 - Improved algorithms for computation of cubic volume in Boolean shapes
 - New templated steppers and templated Equation of Motion providing speedup in release 10.7
 - Fixes and consolidation in all areas