Geometry & Persistency
Recent & ongoing developments

Gabriele Cosmo, CERN EP/SFT

for the Geant4 Geometry & Persistency Working Groups

Contents

- Development and fixes in the last year
  - Introduced in release 10.7 and patches
- Features expected in Geant4 11.0
  - Currently under development and scheduled for inclusion in the next release

VecGeom updates

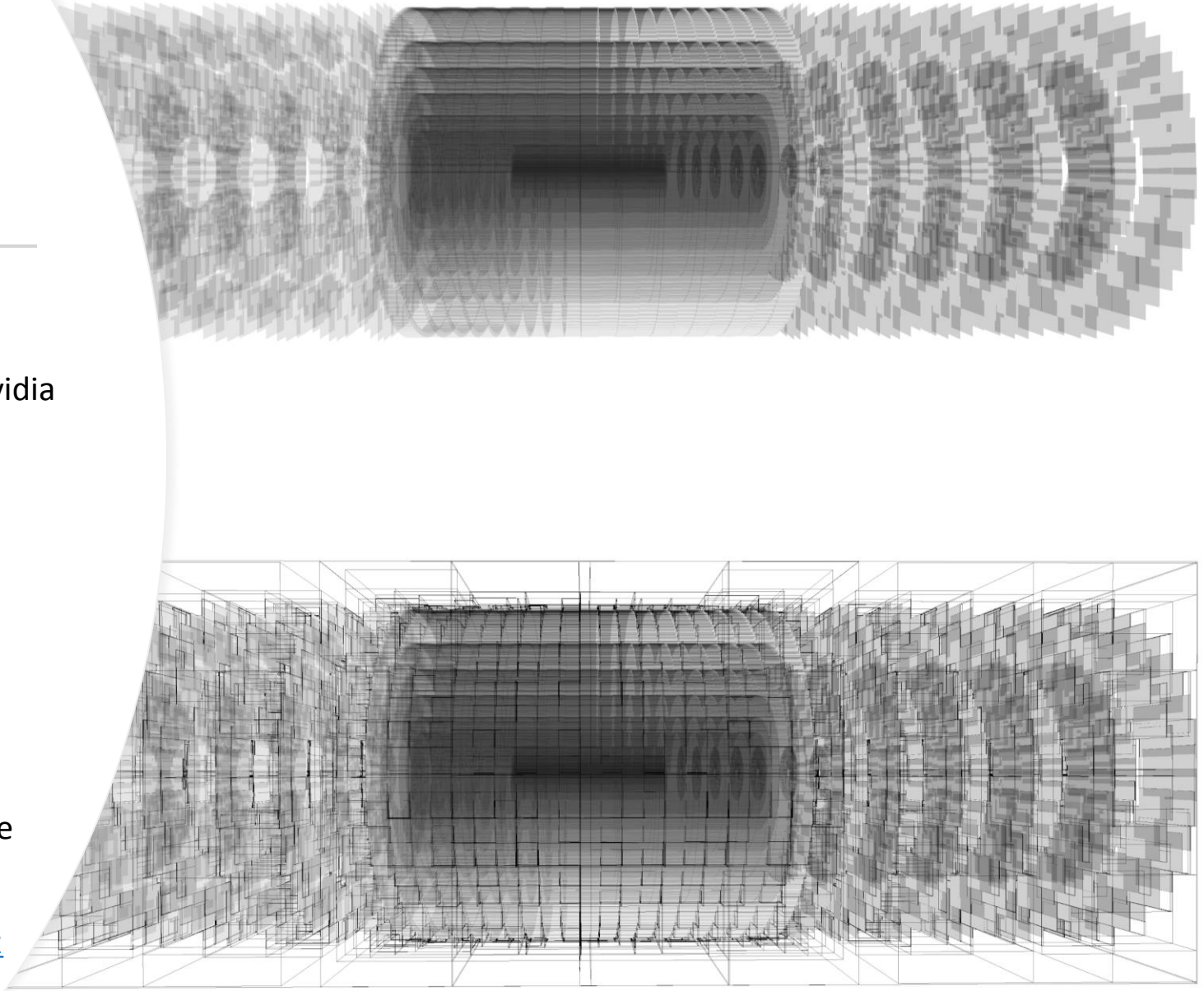
Detailed information in parallel session of last Tuesday:
https://indico.cern.ch/event/1052654/sessions/408407/#20210913

Current VecGeom version

- Version v1.1.17 of VecGeom containing all latest new features and fixes
  - 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; many bug fixes
  - Simplified/unified use of C++ compiler flags; C++17 support; modernised CMake
  - Can be used with latest Geant4 11.0-beta release
- VecGeom primitives can be transparently built through Geant4
  - Original APIs preserved

BVH acceleration for navigation

- Acceleration structure for reducing the number of candidate checks
  - Used natively in RTX hardware, available via Nvidia Optix
  - Implementation available in VecGeom (HybridNavigator2)
    - SIMD, but not GPU friendly
- New BVH navigator available for both CPU and GPU
  - Binary tree structure of axis-aligned bounding boxes
  - Important performance gain compared to “looper” version
    - Speedups: CPU: ≈38.24x, GPU: ≈58.2x for trackML, ~2x for cms2018 in AdePT GPU examples
  - Only up to 30% slower on CPU compared to the AVX2-accelerated HybridNavigator

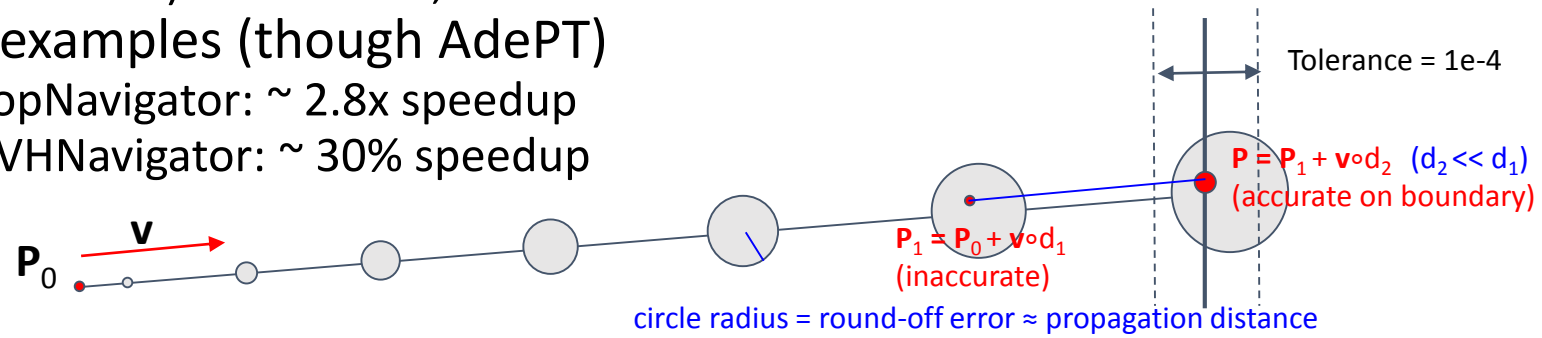


G. Amadio, SFT R&D meeting, 23 Mar 2021

Single-precision VecGeom

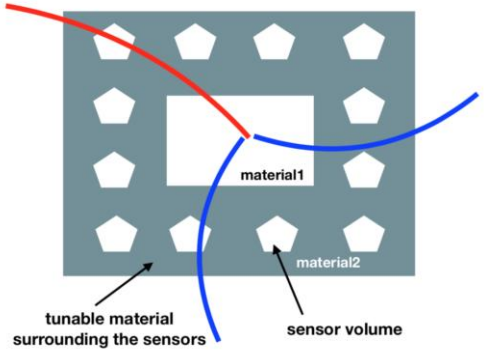
See A.Gheata and M.Kostelnik talks @ geometry parallel

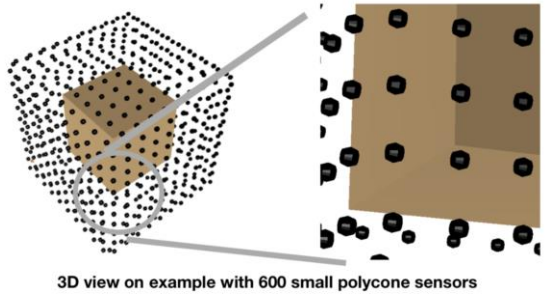
- Make VecGeom a single-precision library on-demand and understand implications on particle transport code (-DSINGLE_PRECISION=ON)
- Tedious work of fixing boundary tolerance algorithm inconsistencies
  - In particular reducing propagation-related rounding errors
- Propagate from an inaccurate point close to boundary rather than from an accurate point far away
- RaytraceBenchmark example (using BVHNavigator)
  - Reading a GDML file and modeling reflections/refractions and specularity
    - Very simple geometry: ~ 7.5% speedup
    - Complex geometry (trackML): ~ 44% GPU, ~ **13.6% CPU!**
  - Physics-enabled GPU examples (though AdePT)
    - Exa9 + trackML + LoopNavigator: ~ 2.8x speedup
    - Exa11 + trackML + BVHNavigator: ~ 30% speedup

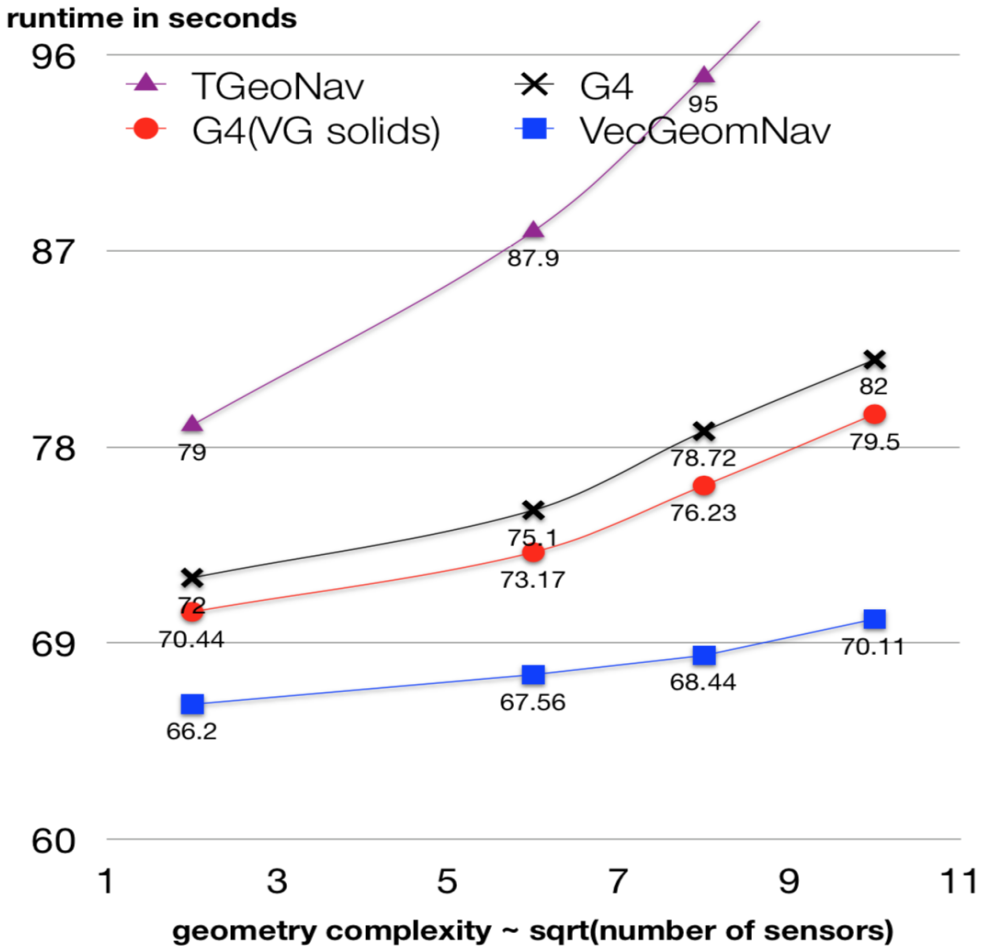


Prototype VecGeom navigation in Geant4

- VecGeom implements structures benefiting from SIMD (vectorized traversal of structure)
  - Increasing size of vector registers on future hardware will automatically make algorithm faster
- Other advantages like strong solid specialization would be made available, especially important for simple solids
- Prototype interfacing navigation:
  - Simultaneous existence of Geant4 and VecGeom geometry with necessary synchronization/translation of states/objects
  - Reusing existing logic to handle the navigation state needed by the Geant4 engine for non-voxelised geometries
  - Custom safety calculation - selectable on per (mother logical) volume (at runtime)
    - Innovative method to calculate safety - using pre-computed lists of candidate volumes in the 'subtraction' volume of mother minus all daughters (requires substantial computation for mother volumes with many daughters)
  - Obtaining promising results (~10% speedup vs. Geant4 navigation) on specific configurable test-case benchmark (layered geometry with configurable number of "sensors" shapes)




3D view on example with 600 small polycone sensors



*See "S.Wenzel et al., A VecGeom Navigator plugin for Geant4, EPJ Web Conf., **245** (2020) 02024"*

New extended example (Beta version)

See J.Apostolakis talk @ geometry parallel

- In release 11.0-beta a new example is available
  - Demonstrating how to enable VecGeom navigation in Geant4
  - examples/extended/geometry/vecGeomNavigation
- Imports a GDML file into Geant4
  - Converts the geometry to create the equivalent VecGeom setup
  - Navigates in either both setups (Geant4 or VecGeom) depending on configuration
  - In 'comparison' mode it reports differences verbosely
- Automatically imports a selected version of the VecGeom/G4VecGeomNavigation package, compiles and links it
  - Temporary solution until helper classes can be properly integrated in Geant4
- Next steps:
  - Improve robustness: identify cause(s) of differences between Geant4 & VecGeom
  - Extend test cases & Performance testing for mid and high complexity setups

Updates in Geometry

Solids

- Optimised implementation of InitializeThetaTrigonometry() in G4Sphere replacing call to std::tan() with pre-calculated quantities — 10.7
  - Addressing problem report #2289
- In G4Sphere, fixed bug in GetSurfaceArea() and revised implementation of GetPointOnSurface() — 10.7.p02
- Revised method GetPointOnSurface() in G4Polycone, G4GenericPolycone and G4Polyhedra — 10.7
  - Added protected auxiliary methods; improved speed and precision
- Optimisations in G4Polycone and G4Polyhedra utility classes — 10.7.p01
  - Reduced cases of bad speculation based on profiling analysis
- Fix in the calculation of normal in G4UAdapter::DistanceToOut() — 10.7
- G4Trap, G4Para: removed spurious character in printout in StreamInfo() — 10.7.p01
  - Addressed problem report #2318
- Fixed calculation of bounding box position in G4Voxelizer::BuildVoxelLimits()
  - Resolving cases of skipped components in G4MultiUnion structures
  - Addressing problem report #2301 — 10.7.p02

More on Solids

- Fixed setting of 'endPhi' for the case of negative 'phiStart' in G4Polyhedra and G4Polycone — 10.7.p02
  - Also addressing problem report #2362 — 11.0.beta
- Simplified implementation of CreatePolyhedron() in G4Polyhedra, G4Polycone, G4GenericPolycone and related wrappers
- Fixed use of conversion factor in G4UPolyhedra wrapper for "generic" polyhedra constructs — 10.7.p02
- Revised GetCubicVolume() method in G4UnionSolid and G4SubtractionSolid to use cubic volume of constituent volume(s) and an intersection solid — 11.0.beta
  - Providing speed improvement for most cases
- Added accessors to retrieve/calculate original values for alpha, theta, phi in G4Trap, G4Para and wrappers — 11.0.beta
- More accurate calculation of distance from point to triangle in G4TriangularFacet
  - Addressing problem report #2401 — 10.7.ref07

Volumes

- Added map for faster search based on name in G4SolidStore, G4RegionStore, G4logicalVolumeStore and G4PhysicalVolumeStore — 11.0.beta
  - Pointers to elements are stored in the map as buckets, grouping elements with same name
- Added creation order index in G4LogicalBorderSurface — 10.7.p01
  - Addressing problem report #2311
- G4GeometryWorkspace: added protection for thread contention in function DestroyWorkspace() — 10.7.p01
- G4ReflectionFactory: added Clean() method for clearing the internal maps on user request and removed problematic Reset() function — 11.0.beta

Navigation

- Enabled use of alternative G4VoxelNavigation class in G4Navigator for voxelised volumes — 11.0.beta
- G4ReplicaNavigation: reduced step correction in ComputeStep() — 10.7.p01
  - Addressing problem report #2302
- G4PathFinder: moved debug printout within debug block in Locate() — 10.7.p01
- G4RegularNavigation: removed debug printout and substituted warning printout with proper G4Exception — 10.7.p01
  - Addressing problem report #2196

Field propagation

- Introduction of templated steppers to be used with G4InterpolationDriver
  - Stepper type templated on the Equation of Motion
  - Equation of Motion templated on exact Field Type
  - Avoid virtual function call & can inline

10.7

More in J.Apostolakis talk @ geometry parallel

➢ *New QSS integration methods and stepper expected for integration in release 11.0*

Updates in Persistency

GDML

- Added new method SetOutputFileOverwrite(G4bool) to the G4GDMLParser, allowing to set the flag to enable overwriting of the output GDML file — 10.7
  - Addressing problem report #2288
- Fix for proper treatment of material properties reading for skinsurface and bordersurface elements — 10.7.p01
  - Adopt the key of the property vector map as 'name + ref'.
  - Addressing problem report #2305
- Fixed parsing of regions in G4GDMLParser::ImportRegions() — 10.7.p01
  - Allow for proper deletion of meta-data lists in destructor
- Fixed reading of units for GenericTrap in G4GDMLReadSolids — 10.7.p01
  - Addressing problem report #2317

More on GDML

- Removed hardcoded string size in G4GDMLWrite causing truncation of values in exported GDML
  - Adressing problem report #2336

10.7.p02

- Fixed case of writing degenerated names (position, rotation and nodes) for multiUnion and assembly entities
  - Addressing problem report #2342

10.7.p02

- Fixed problem of not complete writing of assemblies to GDML file
  - Addressing problem report #2343

10.7.p02

- G4GDMLReadSolids: updated optical surface properties to reflect introduction of DAVIS interface
  - Addressing problem report #2371

10.7.p02

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