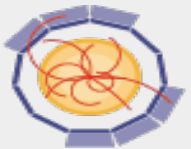


# Common Geometry Primitives library

## WP3

Gabriele Cosmo (CERN EP/SFT)  
*for the VecGeom team*

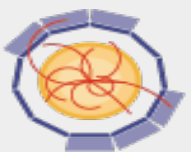
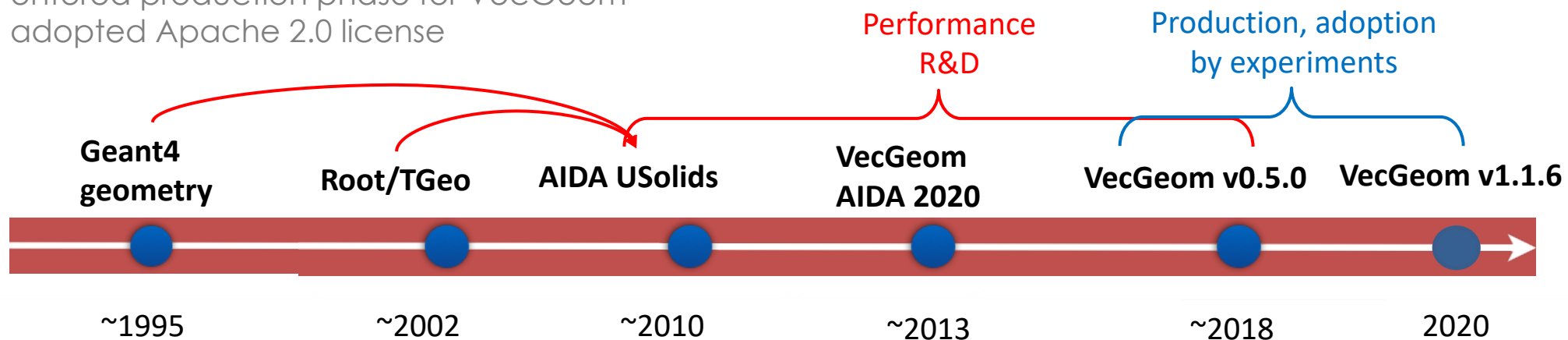


# Outline

- VecGeom evolution & achievements
- Recent developments
  - Last added shapes
  - Prototype interface with VecGeom navigator
- Current activities & perspectives
- Conclusions

# VecGeom Evolution

- AIDA project aiming to unify Geant4 and Root geometry algorithms
  - merge code base
  - pick best implementation and increase code quality
  - improve performance and increase long term maintainability
- Extended scope in **VecGeom**
  - encompass parallelism/vectorization
  - multi-architecture/multi-platform support
  - provide advanced navigation features
- Old initial USolids implementation phased out in 2018
  - entered production phase for VecGeom
  - adopted Apache 2.0 license



**AIDA** 2020

23 April 2020

WP3/Common Geometry Primitives Library

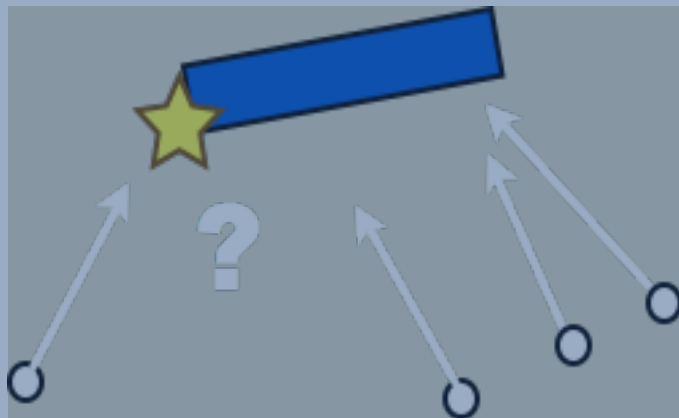
3



# VecGeom: targeting vectorisation

## *Vector signatures*

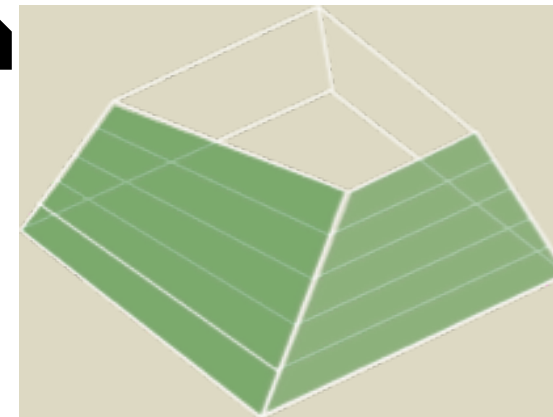
“parallel” collision detection



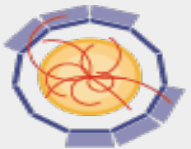
*Multi-particles queries*

## *Internal algorithm vectorization*

internal loop over lateral planes for distance calculation

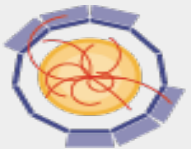
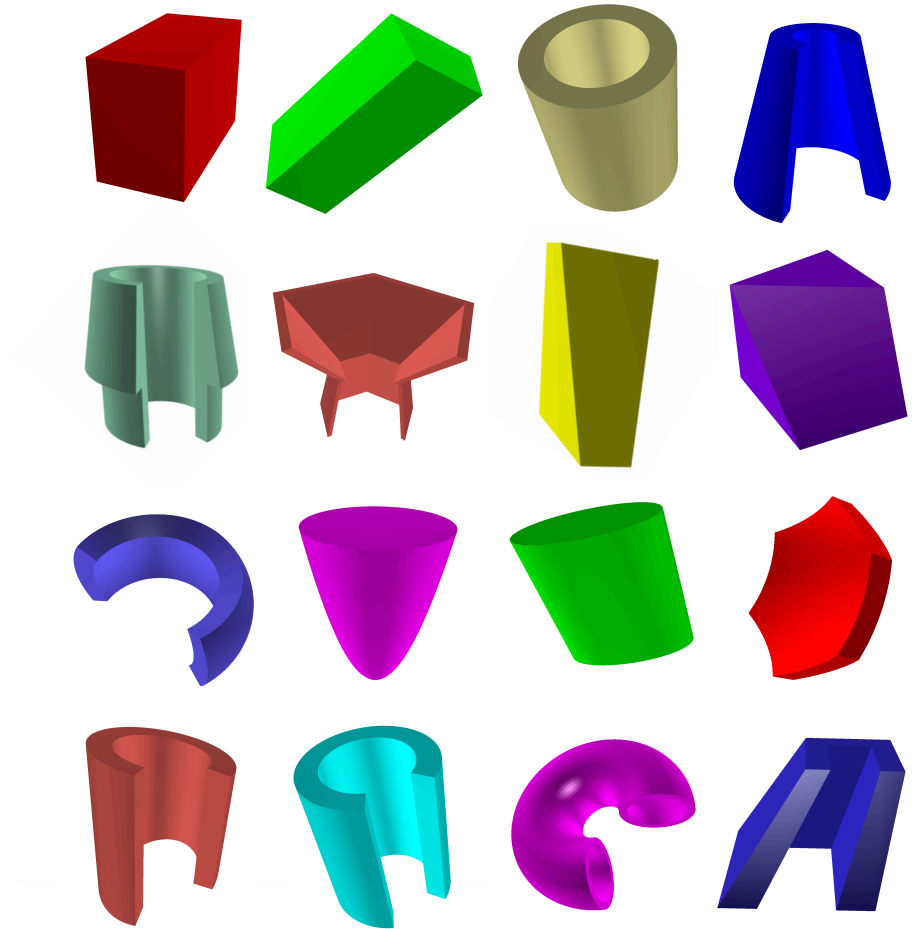


*Beneficial for current simulations*



# Implemented Shapes

- Available in VecGeom:
  - Box, Orb, Trapezoid (Trap), Simple Trapezoid (Trd), Sphere (+ sphere section), Tube (+ cylindrical section), Cone (+ conical section), Generic Trapezoid (Arb8), Polycone, Polyhedron (+generic)
  - Generic Polycone, Elliptical Tube, Elliptical Cone, Ellipsoid (+cut)
  - Paraboloid, Parallelepiped (Para), Hyperboloid, Ellipsoid, Torus (+ torus section), Scaled Solid, Boolean (addition, subtraction, intersection), Cut Tube, Simple Extruded Solid (SExtru), Tessellated Solid, Extruded Solid
  - Tetrahedron (Tet), Multi-Union
- Missing:
  - Twisted shapes (box, trap, tube)  
*complex and infrequent use*



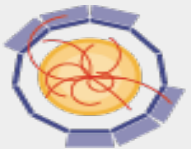
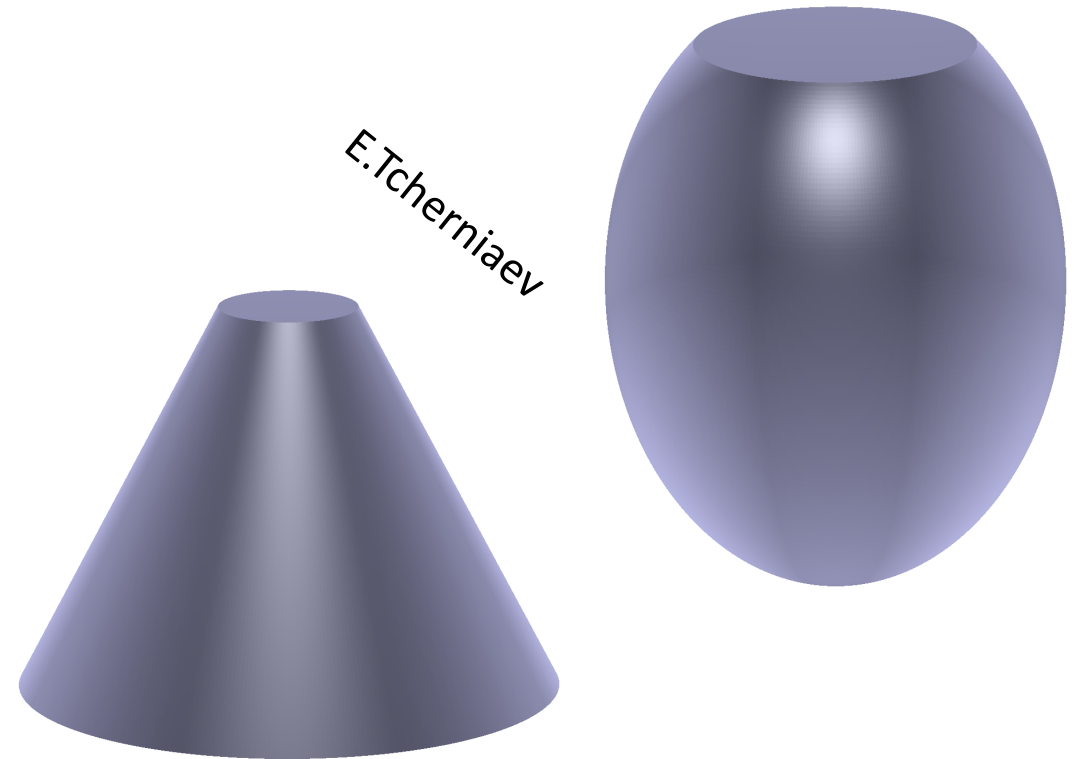
# VecGeom major advantages

- Improved performance in solids algorithms, in particular for complex/composed shapes
  - Polycone, polyhedron, tessellated shape, multi-union construct, extruded solid, ...
- Improved efficiency for solids topologies defined from specialization of primitives
  - E.g. Tube/cone/sphere/... sections/shells
- Efficient navigation algorithms exploiting SIMD acceleration
- Code portability across devices (CPU/GPU)
- Possibility to interface to kernels from industrial ray-tracing libraries
  - E.g. Use of Intel Embree with SIMD support

# Latest developments

# Last added shapes

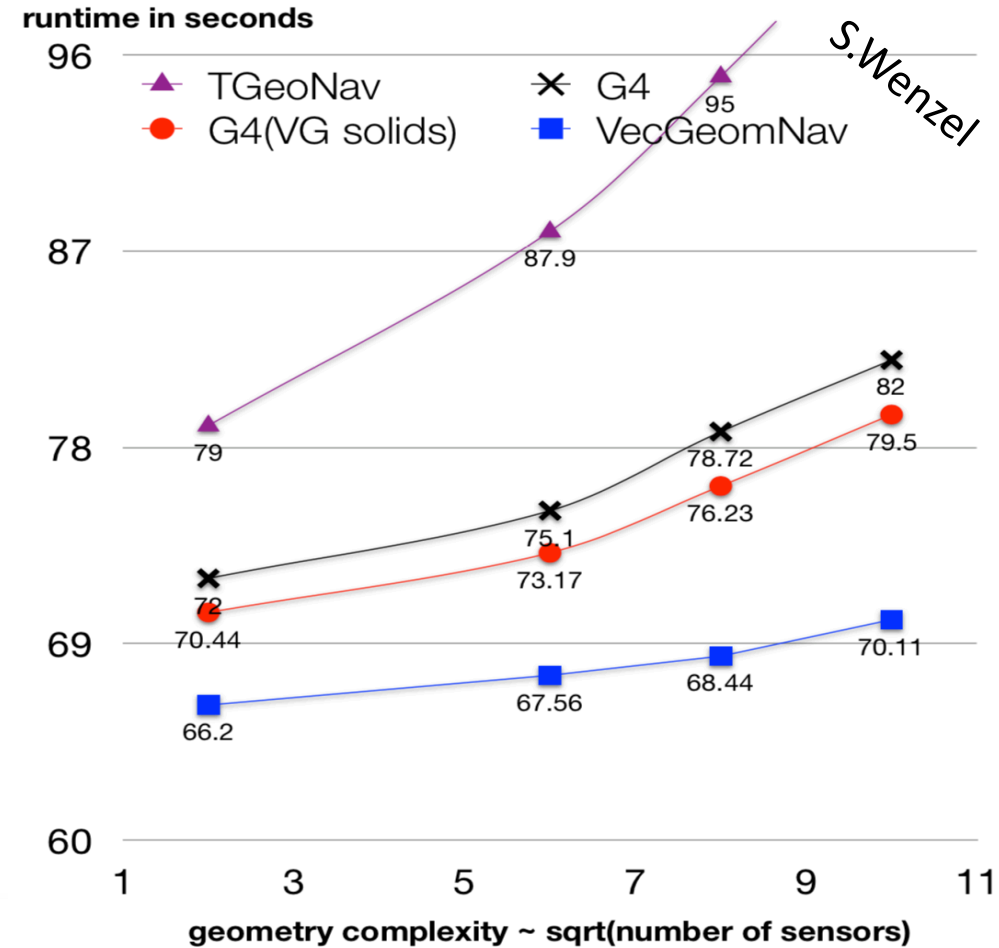
- Ellipsoid
    - A general triaxial ellipsoid with optional cut in Z
  - Elliptical Cone
    - A cone with elliptical cross-section
- *GDML schema now complete for all most common shapes*





# Prototype VecGeom navigation in Geant4

- VecGeom implements structures which benefit from SIMD (vectorized search/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
  - Implemented 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
    - Obtaining promising results on specific configurable test-case benchmark (layered geometry with configurable number of “sensors” shapes)
- See “S.Wenzel et al., A VecGeom Navigator plugin for Geant4, CHEP 2019 Proceedings in preparation”



# Current VecGeom version

- Version v1.1.6 of VecGeom containing all latest new features and fixes
  - New shapes: ellipsoid and elliptical-cone
  - Fixes for corner-case problems in tube and Booleans
  - Extended all shapes functionality with ability to generate polyhedral meshes
    - For use with visualization and geometry debugging sessions
- Reference version for latest Geant4 10.6.p01 release

# Building Geant4 to use VecGeom shapes

## a. Install VecGeom library

- Scalar mode:

```
cmake -DBACKEND=Scalar -DGEANT4=OFF -DCMAKE_BUILD_TYPE=Release \
  [...other optional VecGeom switches as needed...] \
  -DCMAKE_INSTALL_PREFIX=${VecGeomINSTALLDIR} ${VecGeomSOURCE}
make -j8 install
```

- Or .. Vector mode:

```
cmake -DBACKEND=Vc -DGEANT4=OFF -DCMAKE_BUILD_TYPE=Release -DVECGEOM_VECTOR=native \
  [...other optional VecGeom switches as needed...] \
  -DCMAKE_INSTALL_PREFIX=${VecGeomINSTALLDIR} ${VecGeomSOURCE}
make -j8 install
```

## b. Install Geant4

- `export VecGeom_DIR=${VecGeomINSTALLDIR}/lib/Cmake/VecGeom`
- Add `-DGEANT4_USE_USOLIDS=ALL` when configuring Geant4 with Cmake, to use all shapes currently being exercised
- OR, `-DGEANT4_USE_USOLIDS="box;trap"` for configuring to use/replace only specified shapes

**NOTES: Reasonably recent version of the gcc/clang compilers required. Windows VC++ currently not supported**

# Documentation

- VecGeom primitives can be transparently built through either Geant4 or Root modelers
  - Original APIs preserved
- Doxygen documentation can be generated and now covering most key classes in VecGeom
  - Available from VecGeom GitLab site

# Ongoing developments on navigation

- Finalisation of VecGeom navigation interface in Geant4
  - Optional choice (Beta version) in next Geant4 release
- GPU-based simplified simulation prototype
  - Ray-tracing using VecGeom CUDA backend on GPUs
  - Task part of Geant4 Task-Force for R&Ds
    - [http://cern.ch/geant4/collaboration/task\\_force\\_rd](http://cern.ch/geant4/collaboration/task_force_rd)

# Student projects & more...

- Enhancements to specialized navigators, neighbor volume detection
- Addition of replicas/divisions
- GDML writer & Root I/O persistency
- Improve error logging / diagnostics

# Summary

- VecGeom available as production-quality
  - Adopted by CMS experiment in production
  - Planned for use also in ATLAS for replacement of composed shapes (polycones/polyhedra)
  - Latest version v1.1.6 including most recent features - reference for Geant4 10.6 series
- Most primitives from the GDML schema now supported
  - Recently added: Ellipsoid, Elliptical Cone
  - Added ability to generate polyhedral meshes
- Ongoing work for interfacing Geant4 navigation with VecGeom
  - Based on first promising prototype
- Investigating use of VecGeom on GPUs as part of Geant4 R&D

# Contributors

- CERN-EP/SFT + AIDA 2020: G.Amadio, J.Apostolakis, G.Cosmo, A.Gheata, M.Gheata, P.Mato, W.Pokorski, E.Tcherniaev
- J.Martinez Castro, A.Miranda Aguillar (Mexico), P.Canal, G.Lima (FNAL), D.Savin (GSoC student), R.Sehgal (BARC), S.Wenzel (CERN-ALICE)
- Repository for VecGeom
  - <https://gitlab.cern.ch/VecGeom/VecGeom>
- JIRA issue tracking tool
  - <https://its.cern.ch/jira/projects/VECGEOM>