Common Geometry Primitives library

WP3

Mihaela Gheata, CERN EP/SFT

for the VecGeom team

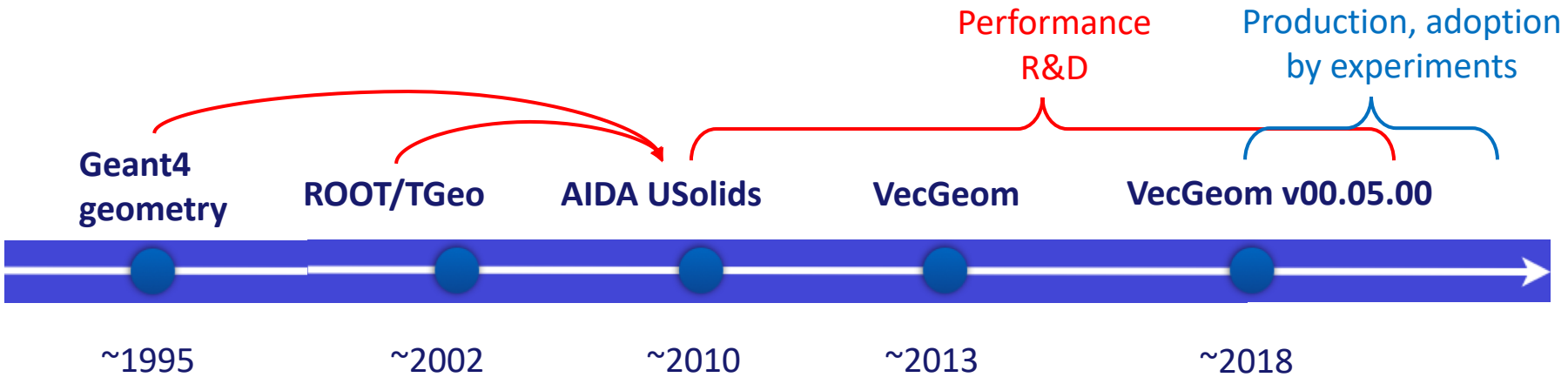
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 654168.

Content

- Evolution
- Status
  - Code restructuring
  - Solids optimisations
  - VecGeom in experiments
- Directions and work plan
- Conclusions

Evolution

- AIDA project aiming initially to unify and modernize geometry algorithms
- Scope extended to encompass parallelism/vectorization and multi-architecture/multi-platform support -> **VecGeom**
- 2018 marks phasing out the initial USolids implementation while entering the production phase for VecGeom
  - Adopted Apache 2.0 license
- Next: integration in ROOT & Geant4 as complete alternative to native navigation

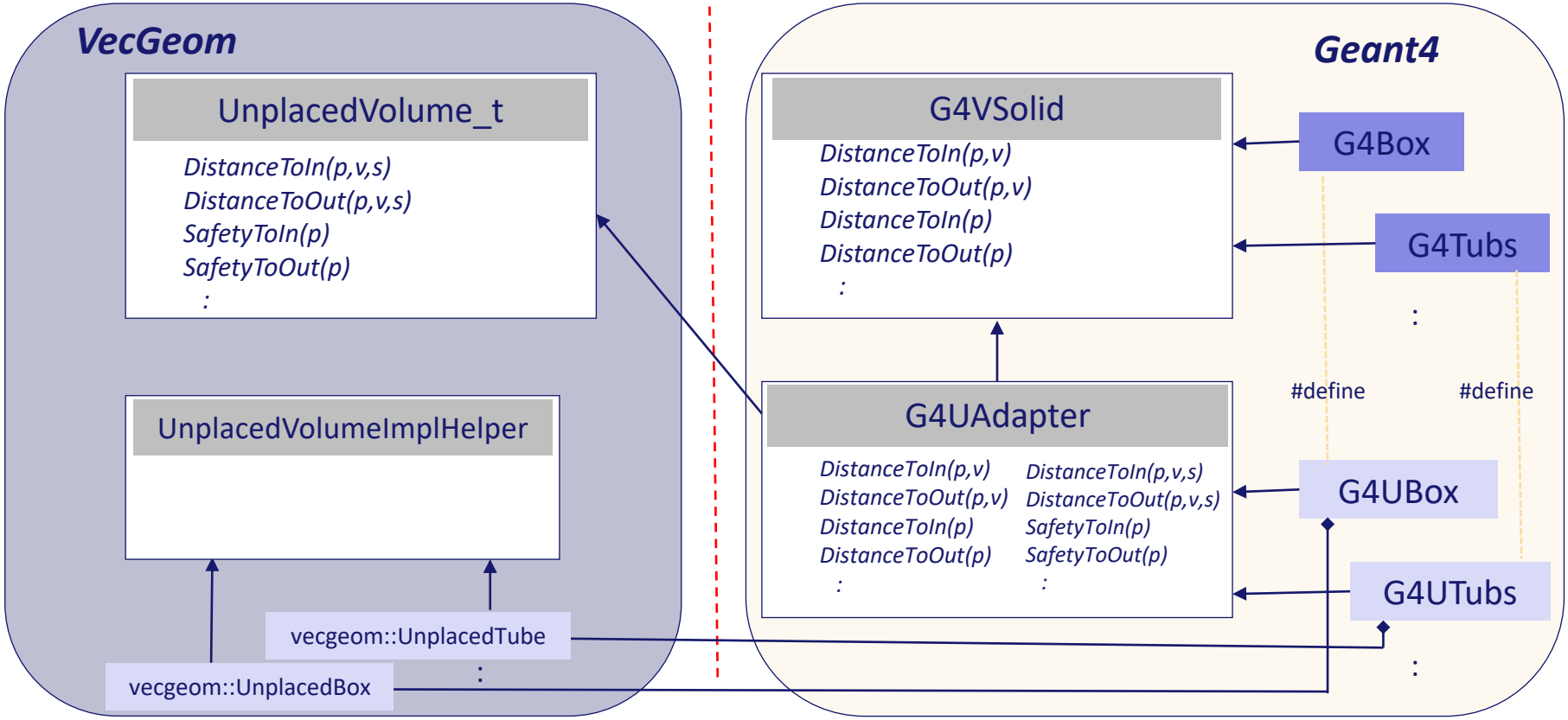


Code restructuring

- Externalized VecCore that became a common vectorization abstraction layer used in other areas (GeantV, ROOT, experiments)
  - VecCore maintained now under root-project GitHub umbrella:

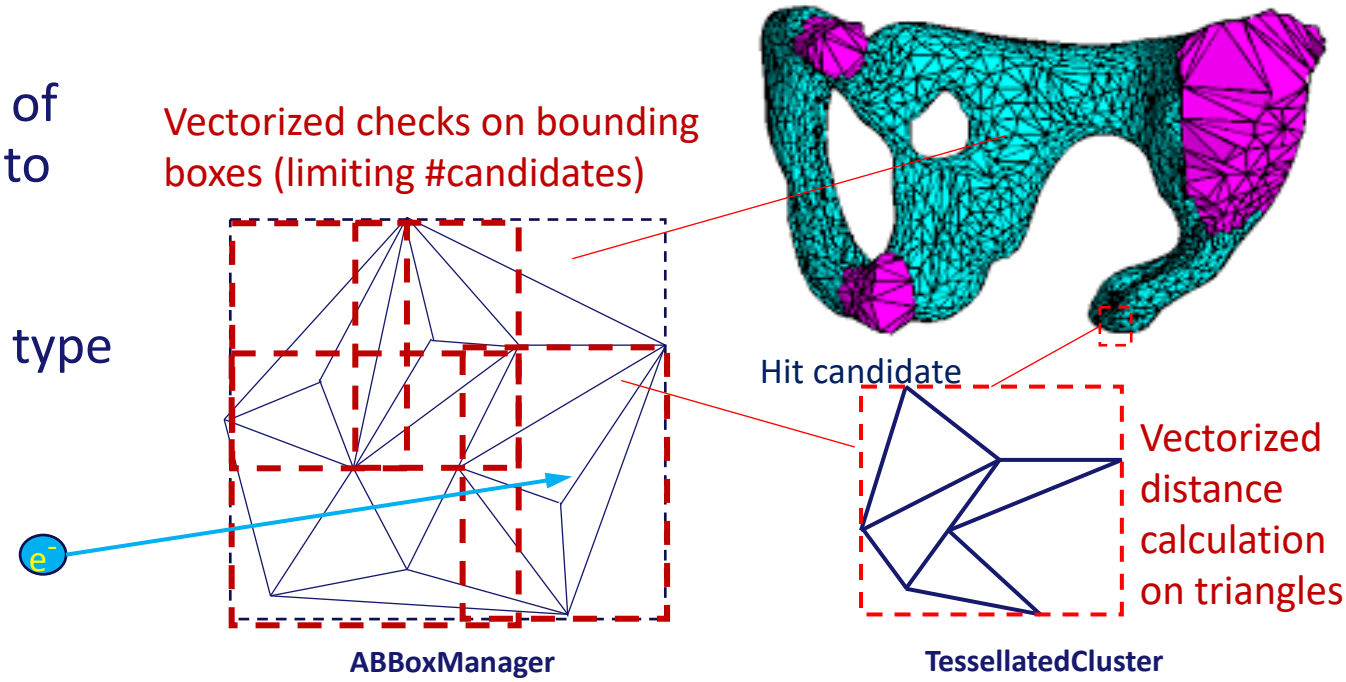
  https://github.com/root-project/veccore
- Completed migration of interfaces to use VecCore and templates on data type (scalar/vector) for all navigation algorithms
  - Mostly internal changes to increase flexibility
- Added possibility to use solid specializations via a common factory mechanism
  - Provide to the user the best specialization of a solid for a given set of parameters
- Phased out USolids implementation
  - All functionality directly available in VecGeom or Geant4 with better performance

New interface to Geant4



Solids development: Tessellated solid

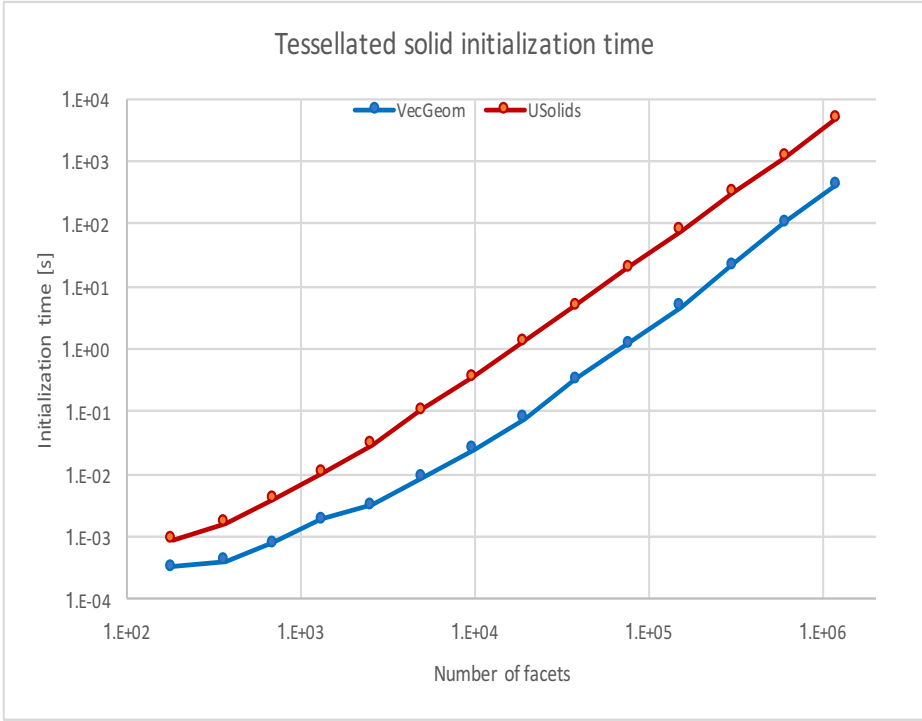
- Complete re-write of algorithms aiming to vectorize in single particle mode on facets of the same type (triangles)



- 2-level vectorization
  - Find hit candidates by fast vectorized checks on bounding boxes of clusters of triangles
  - Optimize checks on selected clusters by vectorizing on triangle components

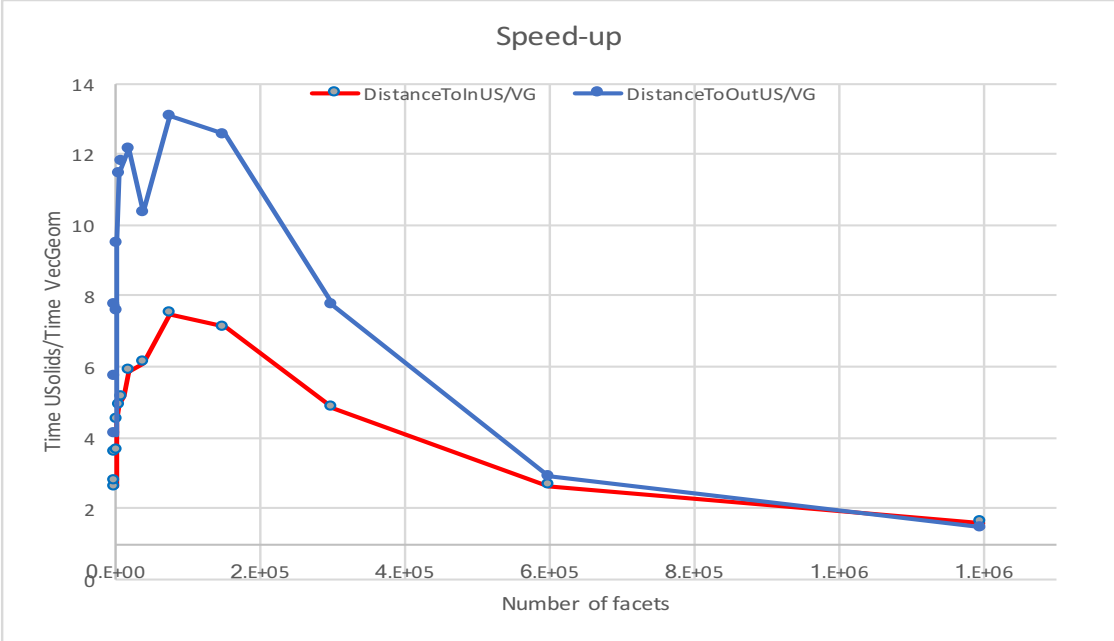
Tessellated Solid: Initialization performance

- Initialization time bound by clustering (VecGeom) and voxelization (USolids/Geant4)
- In VecGeom the clustering of bounding boxes by the helper is $O(N^2)$
  - Still ~x10 faster than the old algorithm
  - Can be largely improved
- New clustering algorithm needed. Time becomes prohibitive for large structures (>10^6 facets)



Tessellated Solid: Navigation performance

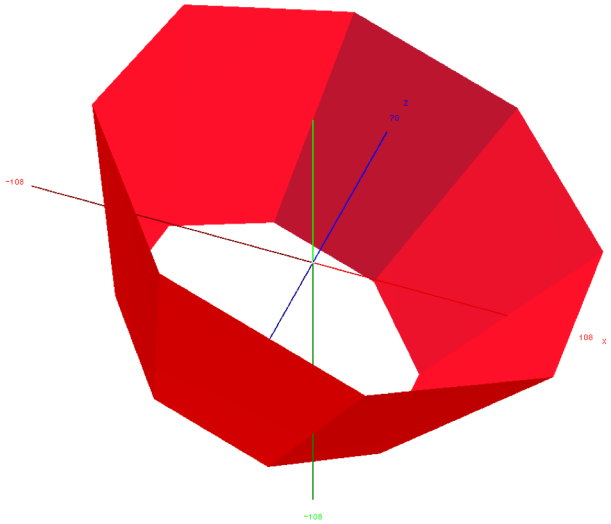
- Large performance improvement for distance computation for up to $O(10^5)$ facets
- Log(N) complexity in USolids approach catches up linear complexity in new algorithm for ~10^6 facets
- Improvements possible for very large number of facets
  - 2 levels of bounding boxes now -> too many candidates
  - Number of levels can be increased



Speed-up for DistanceToIn/DistanceToOut computation compared to USolids

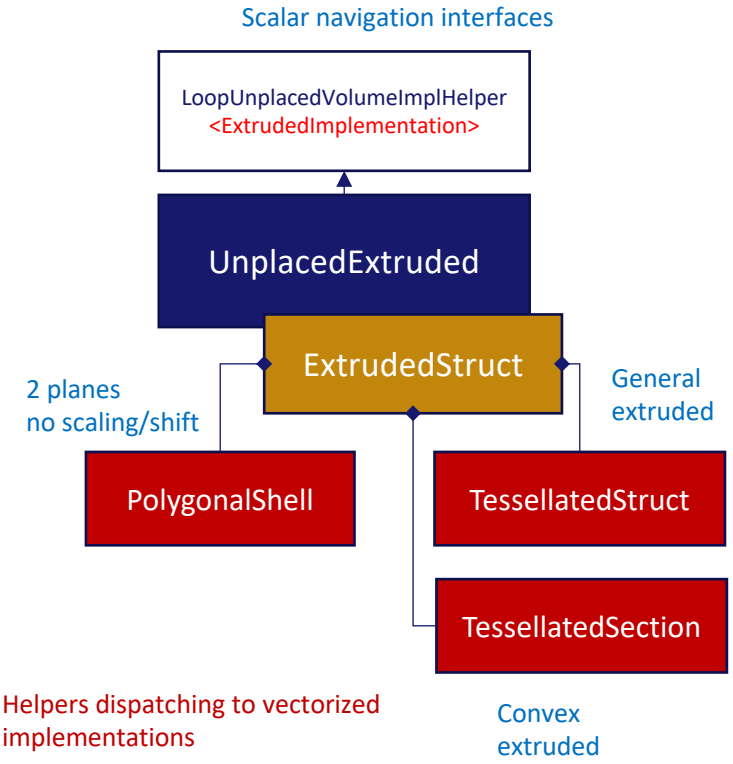
Extending multi-faceted approach

- Implementation of facet-based solids (trapezoids, polyhedra, extruded solid) using a helper based on TessellatedCluster
- Implemented new SIMD helper class TessellatedSection
  - Representing a convex surface made of quadrilateral tiles, organized in clusters of size = vector length
  - Delimited by two Z planes
  - Using explicit vectorization on tiles using VecCore types
- Can represent sections of multi-faceted solids
  - Polyhedra, extruded solids, trapezoids
  - Tests ongoing



Solids development: Extruded Solid

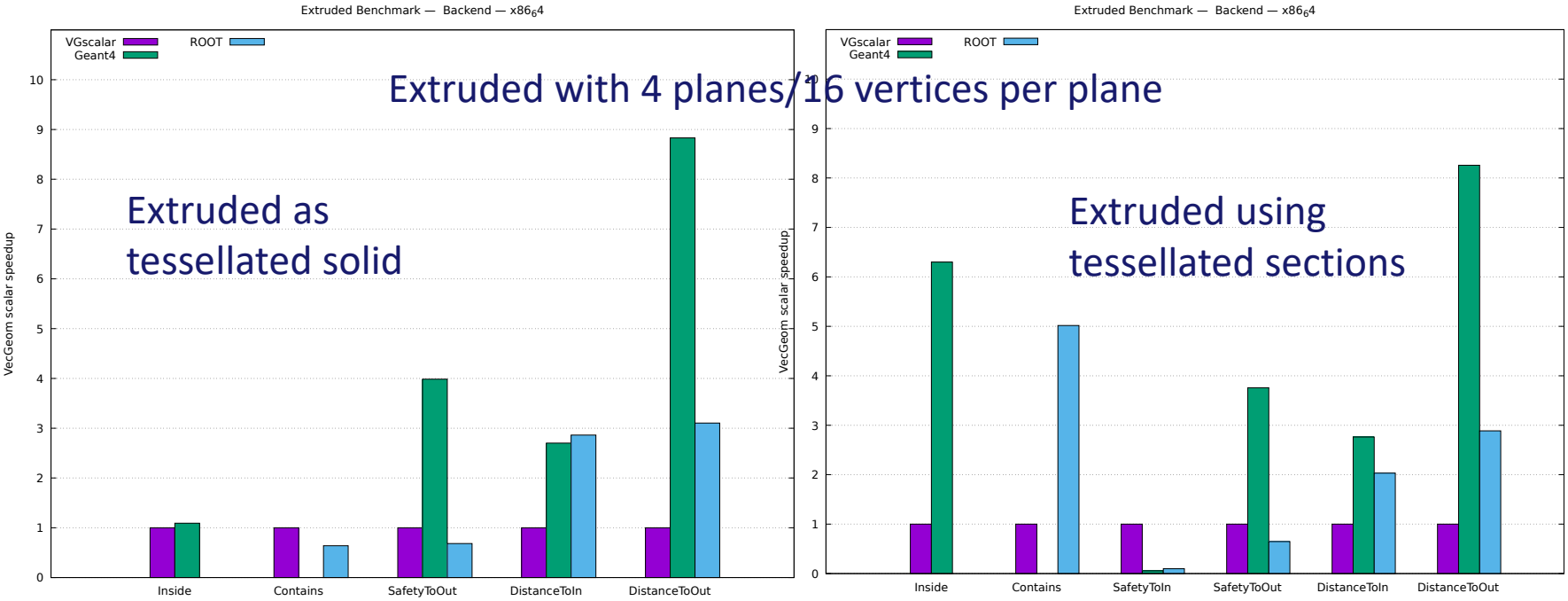
- Initially implemented as a Tessellated Solid for the general case (arbitrary polygon, arbitrary number of Z planes)
  - Performance gain due to improved algorithm
- Specialized to simple extruded implementation (Sextru) for 2 planes right prism
- Specialized to using tessellated sections for the convex case with more Z planes
- Planning to use the new factory mechanism for selection of the best implementation



Performance for convex extruded solid

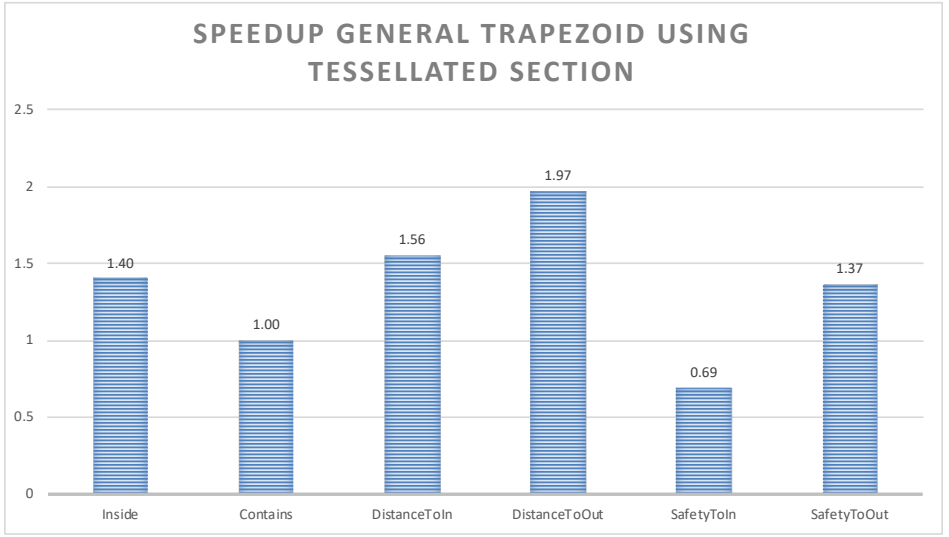
- Initial implementation based on tessellated solid not so fast for Contains/Inside
- Performance increased by large factor after using TessellatedSection

Speedup compared to ROOT/Geant4



New general trapezoids

- For simple solids (box, trd, parallelepiped) not expecting gains using tessellations
  - Gains in scalar mode obtained by scalar optimizations (usage of vectorized Min/Max, better use of temporaries)
- For the solids already vectorized in the scalar case the improvement is very limited
  - 5-10% in some methods of Polyhedron, Trapezoid
- Large improvements observed for previously non vectorized cases (general trapezoids)



Speed-up of general trapezoid compared to previous implementation

VecGeom in experiments

- In CMS, tests of Geant4 10.4 with VecGeom started early in 2017, during the development process
  - Coordinated work between the VecGeom, Geant4 and CMS teams
  - Observed 7-13% improvement in CPU performance with similar memory usage when using Geant4 10.4 + VecGeom
  - Decided to use VecGeom for 2018 productions
- LHCb started also testing VecGeom in fall 2017
  - No speed-up measured due to usage of very simple solids
- The Fermilab Mu2e experiment will perform precision measurements of the neutrino- less muon-to-electron conversion rate to search for new physics
  - Tests of Geant4 10.4, including VecGeom v00.05.01 started in January 2018
- ALICE planning to use full VecGeom features in Geant4 simulation
  - Needs VecGeom navigator interfaced from Geant4

WP3 deliverables achieved

Grant Agreement No: 654168

AIDA-2020

Advanced European Infrastructures for Detectors at Accelerators

Horizon 2020 Research Infrastructures project AIDA-2020

DELIVERABLE REPORT

IMPLEMENTATION OF EXTENSIONS IN USOLIDS

DELIVERABLE: D3.1

Document identifier:	AIDA-2020-D3.1
Due date of deliverable:	End of Month 32 (December 2017)
Report release date:	20/12/2017
Work package:	WP3: Advanced Software
Lead beneficiary:	CERN
Document status:	Final

Abstract:

The Unified Solids (USolids) [1] interfaces have now been fully integrated within the VecGeom [2] library with extended vector signatures and implementing most of the shapes defining the standard set in the GDML schema [3]. This makes it possible in Geant4 [4, 5] version 10.4 to wrap calls to VecGeom directly and in a transparent manner, with minimal performance penalty. Preliminary tests done with both CMS and LHCb simulation applications using the new approach show overall performance gains of 5-8% compared to the native Geant4 primitives. Developments for new shapes such as the tessellated and extruded solids were focused on providing internal vectorization of the algorithm for the fast detection of facets intersections, with considerable performance gains compared to previous implementations. The global vectorised navigation interfaces of VecGeom are now completed and are used by the GeantV [6] prototype.

Next steps

- VecGeom navigation interfaced with Geant4 and ROOT
  - Geant4: first implementation as derivation from G4Navigator
  - Investigating also a templated approach for using VecGeom navigator
- Complete the set of remaining less-used solids
  - Multiple union, twisted solids, planar half-space used in Boolean solids
  - New version of more robust torus in development
- Finalizing solids factory implementations to handle all specializations
- Continue testing on realistic geometry setups
  - Robustness tests
  - Exact tracking comparison with original Geant4 implementation
  - Include new snapshots from LHC detectors & feedback from experiments
- VecGeom persistency will be implemented for GDML and ROOT formats
  - GSoC and summer student projects this year
- Write a first version of a user guide

Summary

- VecGeom now available as first production-quality release v00.05.00
  - Tested by several experiments, adopted by CMS in production for 2018
- Interface re-factoring completed
  - Based on external VecCore, used by other packages than VecGeom
  - USolids implementation phased out: all functionality now available in VecGeom
  - Factories to handle specialized solids
- New solids/performance improvements for existing ones
  - General trapezoid, but also simple solids
  - Fast version of tessellated and extruded solids now available
- Ongoing work for interfacing Geant4 navigation with VecGeom
  - Navigation interface re-design in Geant4
  - Documentation

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), 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*