

Common Geometry Primitives library

WP3 – 3/6/2015

Gabriele Cosmo, PH/SFT

Outline

- Motivations & outlook
- Achieved status
- Synergy with VecGeom developments
- Proposed deliverables

Original motivations for a common solids library

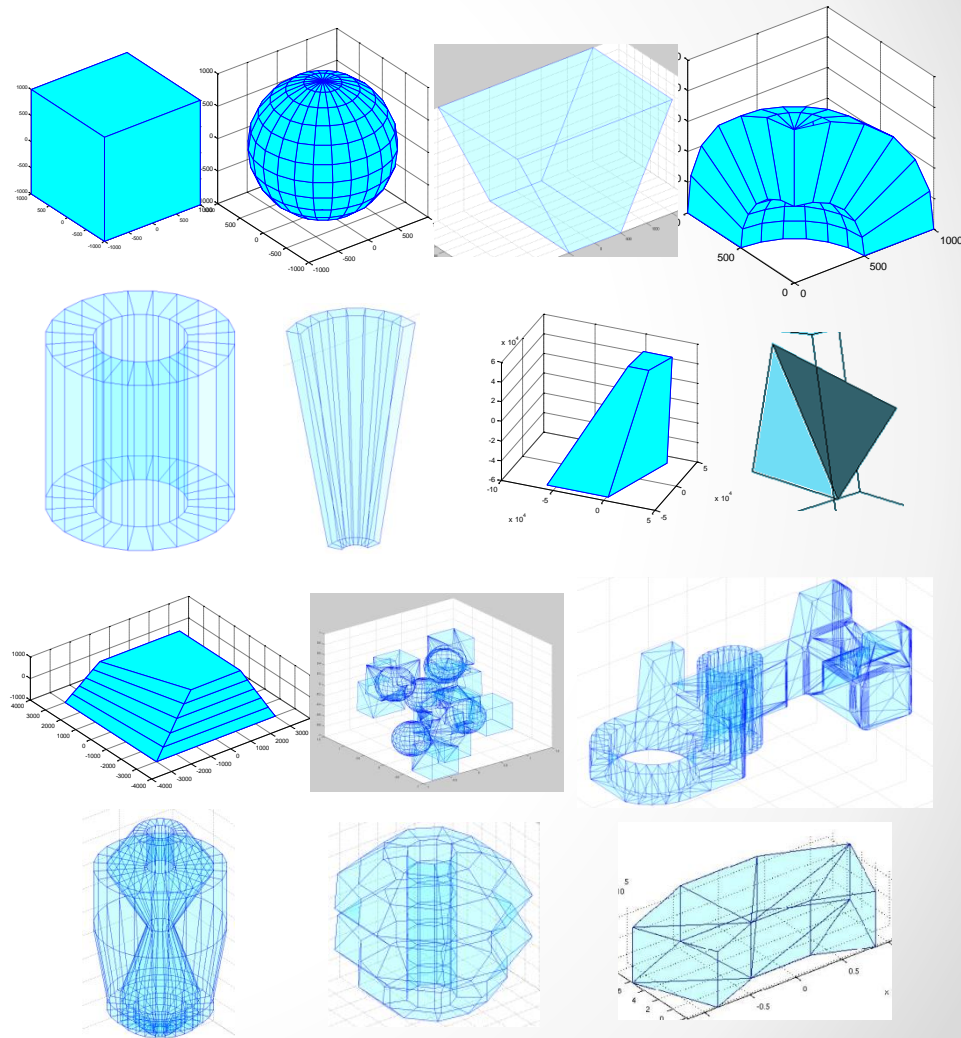
from AIDA

- Optimize and guarantee better long-term maintenance of ROOT and Geant4 solids libraries
- Create a single high quality library to replace solid libraries in Geant4 and ROOT
 - Starting from what exists today in Geant4 and ROOT
 - Adopt a single type for each shape
 - Significantly optimize (Multi-Union, Tessellated Solid, Polyhedra, Polycone)
 - Reach complete conformance to GDML solids schema
- Create extensive testing suite

Primitives implemented so far

from AIDA

- Box
- Orb
- Trapezoid
- Sphere (+ sphere section)
- Tube (+ cylindrical section)
- Cone (+ conical section)
- Generic trapezoid
- Tetrahedron
- Arbitrary Trapezoid
- Multi-Union
- Tessellated Solid
- Polycone
- Generic Polycone
- Polyhedra
- Extruded solid



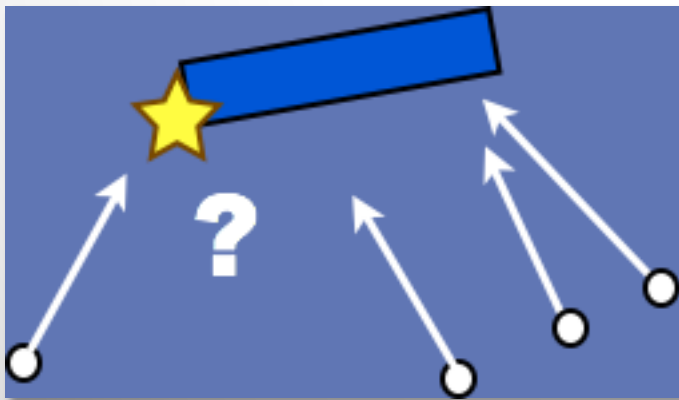
Status of USolids library

- Library in the current form distributed as optional module in the latest Geant4 release 10.1
 - Firstly introduced in Geant4 release 10.0 last year in reduced form
 - Now possible to use it also as external independent library
 - Validation of shapes on realistic detector geometries ongoing
- Testing suite further extended for performance/accuracy measurements
- Code available in the AIDA SVN repository
 - Using standard AIDA CMake setup for build/installation
- Documentation available from web
 - <http://aidasoft.web.cern.ch/USolids>

Ongoing Developments on Primitives

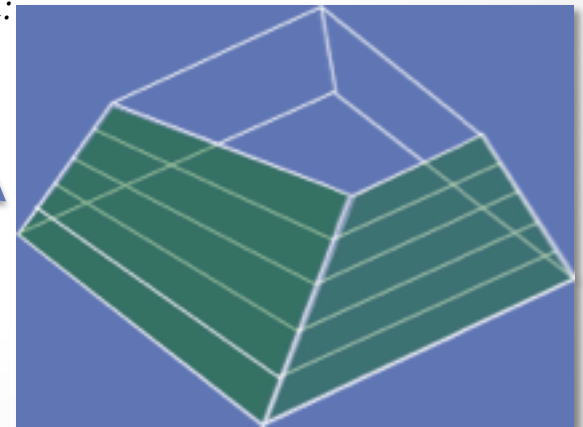
VecGeom

- Started as feasibility study of vectorization for geometry
 - Part of the development going on for the *Geant Vector Prototype*
 - Extending signatures of classes to enable use of vectorisation
 - Review algorithms on all developed shapes to efficiently apply vectorisation and strong code specialization
 - AIDA-2020 proposal
- Geometry primitives code development to be considered as long-term evolution of USolids
 - To replace/complement the current implementation in USolids
 - Developed back-to-back with USolids as independent library
 - Sharing same interfaces



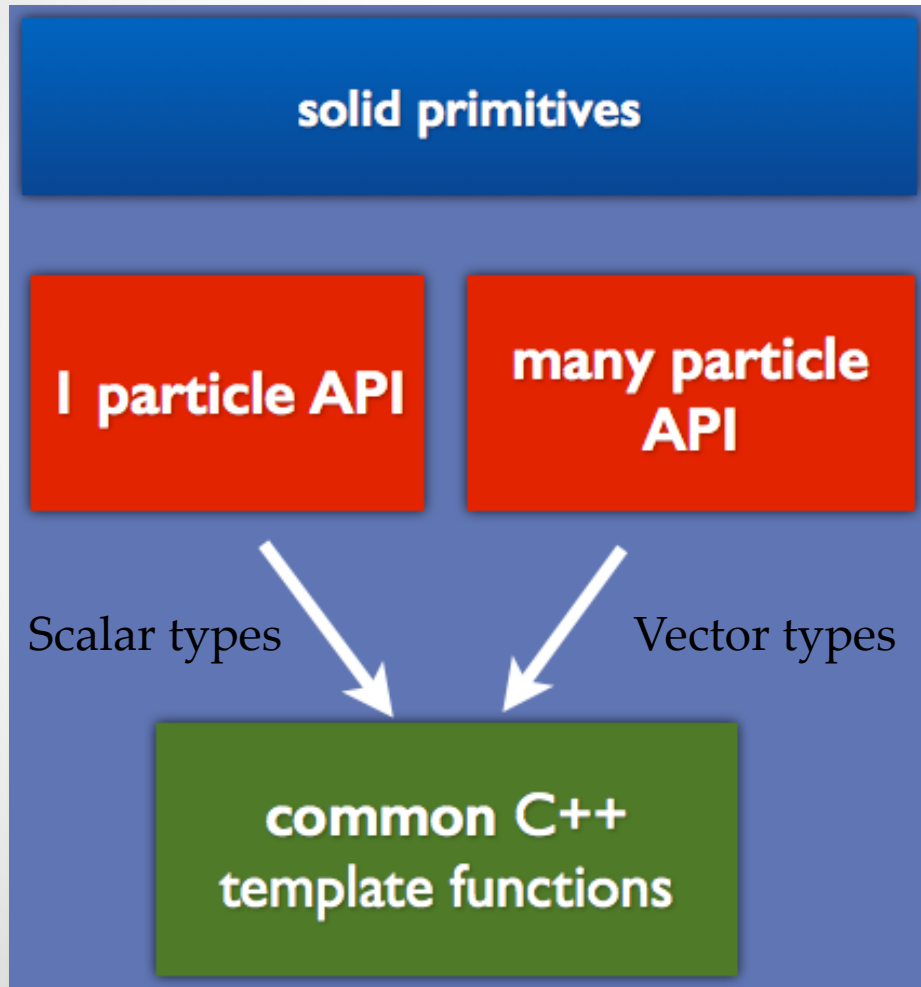
*Internal algorithm vectorization:
loop over lateral planes for
distance calculation*

*Vector signatures:
“parallel” collision detection*



VecGeom

The approach

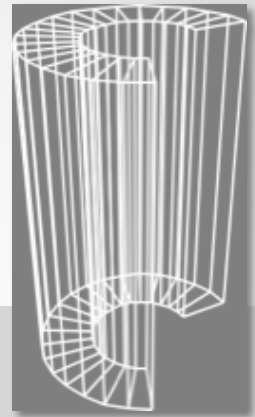


efficient SIMD vectorization achieved by using vector libraries (e.g. Vc) providing C++ approach to explicit vectorization

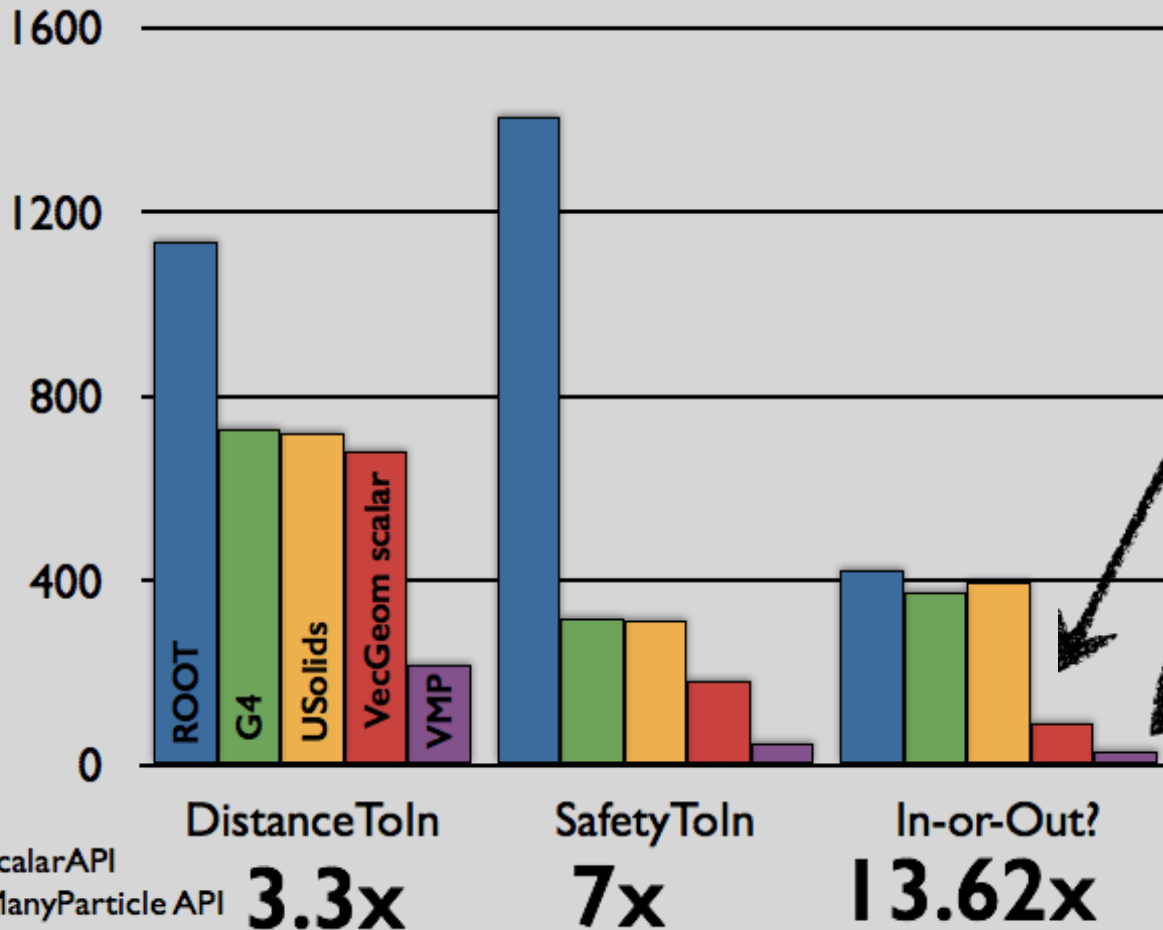
template C++ programming to reduce code multiplication due to proliferation of interfaces

Performance case study

Tube segment



time units



improved scalar performance

- improved algorithms (avoid atan2)
- template shape specialization

excellent SIMD vector performance

total speedup cmp to USolids

USolids / VecGeom

work in progress

- Missing USolids shapes from the standard set
 - Cut Tube, Torus, Ellipsoid, Elliptical Tube, Elliptical Cone, Hyperboloid, Paraboloid, Twisted shapes
 - Boolean compositions: Simple Union, Subtraction, Intersection
- Currently under development through VecGeom
 - Torus, Ellipsoid, Hyperboloid, Paraboloid
 - Boolean composite shapes with templated signature

Proposed deliverables

a) Extend/review Unified Solids Library

- Review algorithms of all implemented shapes for vectorisation using Single Instruction, Multiple Data (SIMD) instructions (from VecGeom package)
- Replace existing shapes with vectorised version, respecting defined interfaces and specifications for use in Geant4, Root and Geant Vector prototype
- Complete implementation of missing shapes from the standard set and assure full compatibility with GDML schema

b) Testing & Integration

- Extend testing suite to exercise the new vectorised algorithms
 - Creation of *Comparison Solid* for comparison of methods of two shapes during run-time
- Run prototype through the Geant4 simulation toolkit with examples and applications including realistic LHC detectors geometries
- Integrate USolids extensions for vectorisation in Geant4, ROOT and Geant Vector Prototype

Resources

- Assuming current resources sum up to ~0.5 FTE, adding up contributions from PH/SFT:
 - John Apostolakis
 - Gabriele Cosmo
 - Andrei Gheata
 - Tatiana Nikitina
 - Sandro Wenzel
- Additional 1 FTE for 1.5-2 years already allocated:
 - Mihaela Gheata (PH/UAI), AIDA PJAS

Thanks