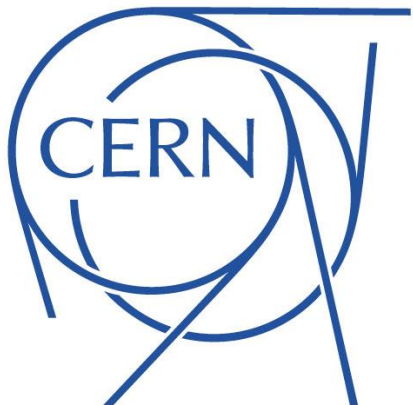


# Updates on solids modelling in VecGeom

Gabriele Cosmo, CERN EP/SFT

*for the Geant4 & VecGeom developers teams*



# Contents

- The VecGeom library
  - Status of implementation & validation of shapes in VecGeom
  - Newly introduced shapes and constructs
  - Usage of VecGeom shapes in Geant4 10.5

# Shapes implementation status

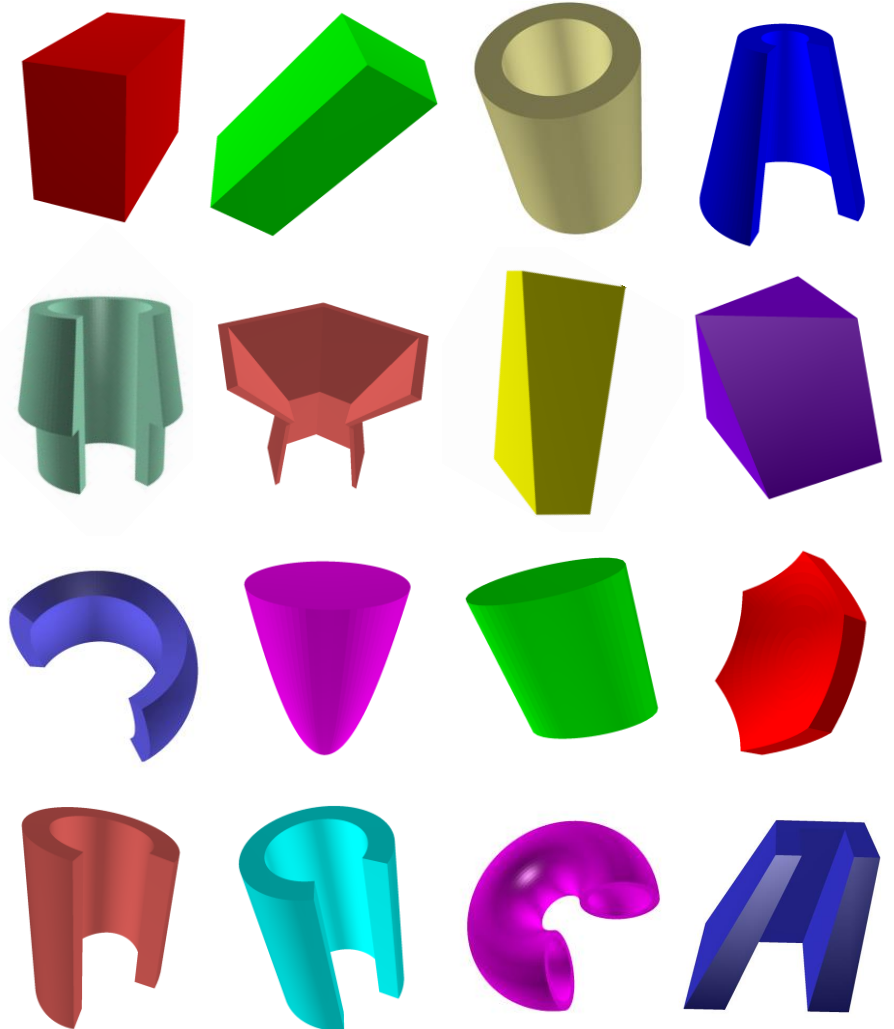
- Available in VecGeom:

- Box, Orb, Trapezoid (Trap), Simple Trapezoid (Trd), Sphere (+ sphere section), Tube (+ cylindrical section), Cone (+ conical section), Generic Trapezoid (Arb8), Polycone, Polyhedron
- Paraboloid, Parallelepiped (Para), Hyperboloid, Ellipsoid, Torus (+ torus section), Scaled Solid, Boolean (addition, subtraction, intersection), Cut Tube, Simple Extruded Solid (SExtru)
- Tetrahedron (Tet), Multi-Union, Tessellated Solid
- Extruded solid

*expressed as specialization of Tessellated Solid*

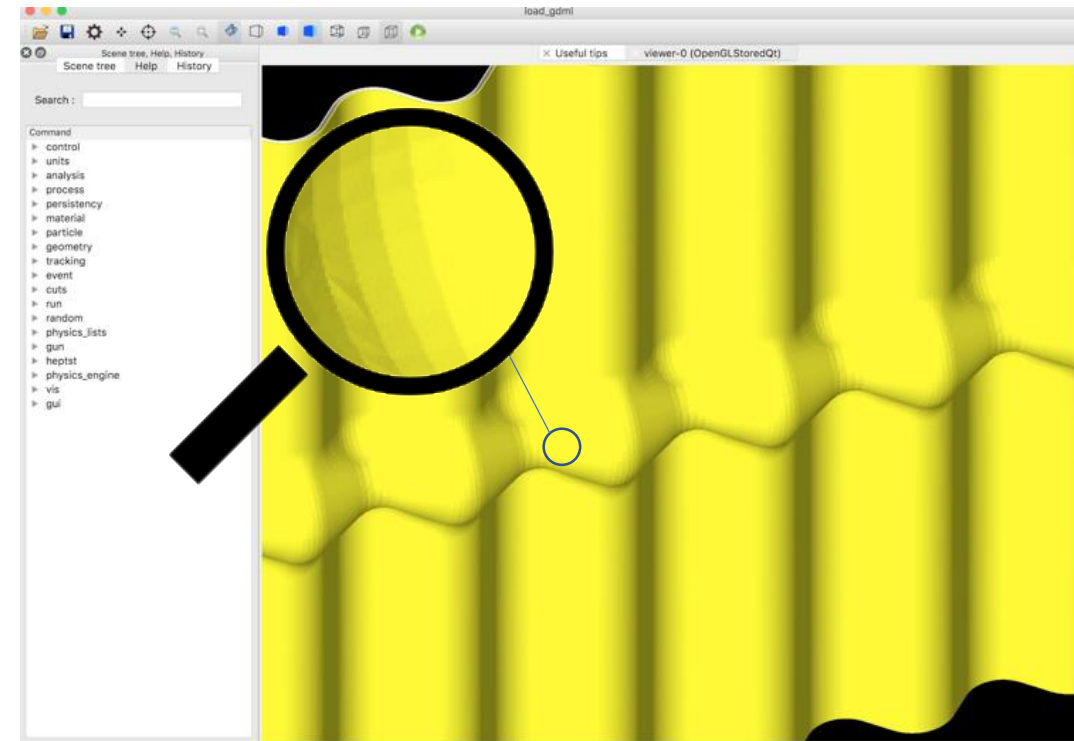
- Missing:

- Generic Polycone, Generic Polyhedron  
*can be expressed as composition of cones / polyhedron sections*
- Elliptical Cone, Elliptical Tube  
*can be composed through scaling*
- Half-spaces/planes
- Twisted shapes (box, trap, tube)  
*complex and infrequent use*



# Tessellated solid

- Surfaces are represented as meshes of connected facets (ideal for importing from CAD models)
  - Use of vectorization natural for looping over facets
- A more advanced approach is to pre-select the facets to be checked
  - In most cases not all facets have to be checked. Can we select the subset of candidates also in vectorized mode?
  - In Geant4 voxelization is used for this purpose

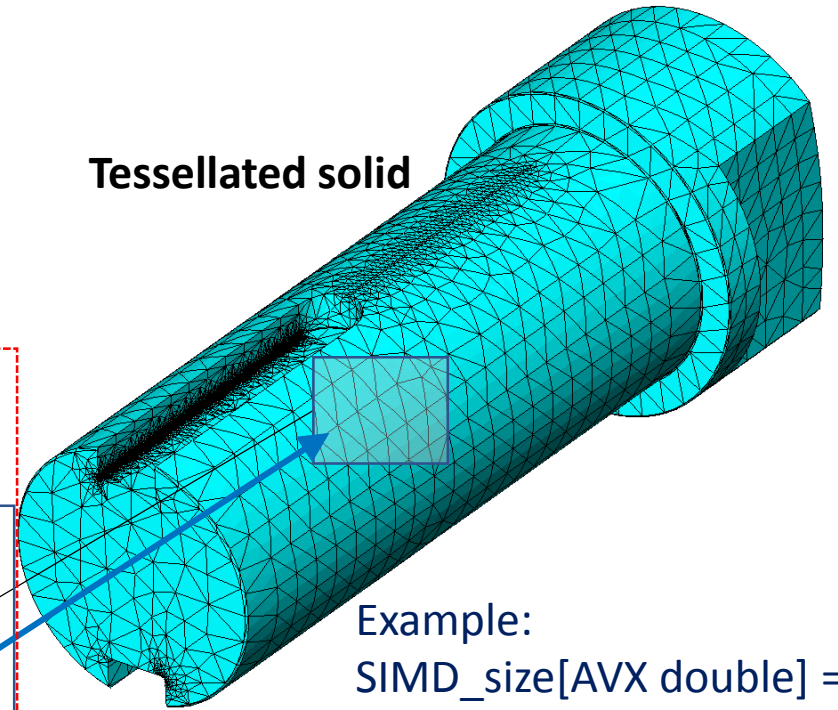


*The LHCb RF-foil loaded from GDML as tessellated solid (164k facets) and visualised in Geant4 through VecGeom*

# Multi-level vectorization

Computing the distance to the tessellated solid

Tessellated solid



Group neighbor triangles in **clusters**, 4 per cluster. Store data in `vecCore::Double_v`

Make groups of **bounding boxes** of clusters, 8 per group

Continue grouping by 8 and make "super" bounding boxes  
-> bounding volume hierarchy (BVH)

Vectorize in float computation of distances to super-boxes to select only hit candidates

e<sup>-</sup>

Repeat the same with the content of the boxes being hit, until we get the candidate clusters

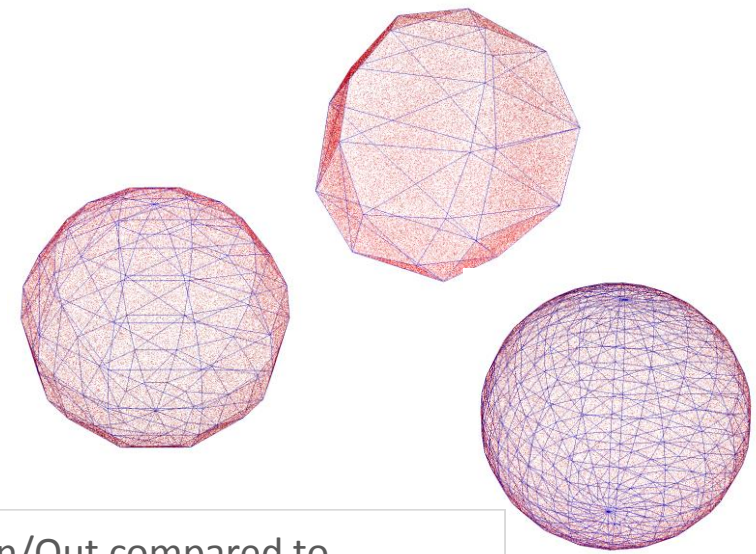
Example:  
SIMD\_size[AVX double] = 4  
SIMD\_size[AVX float] = 8

Vectorize in double computation of distances to triangles in each cluster

M.Gheata

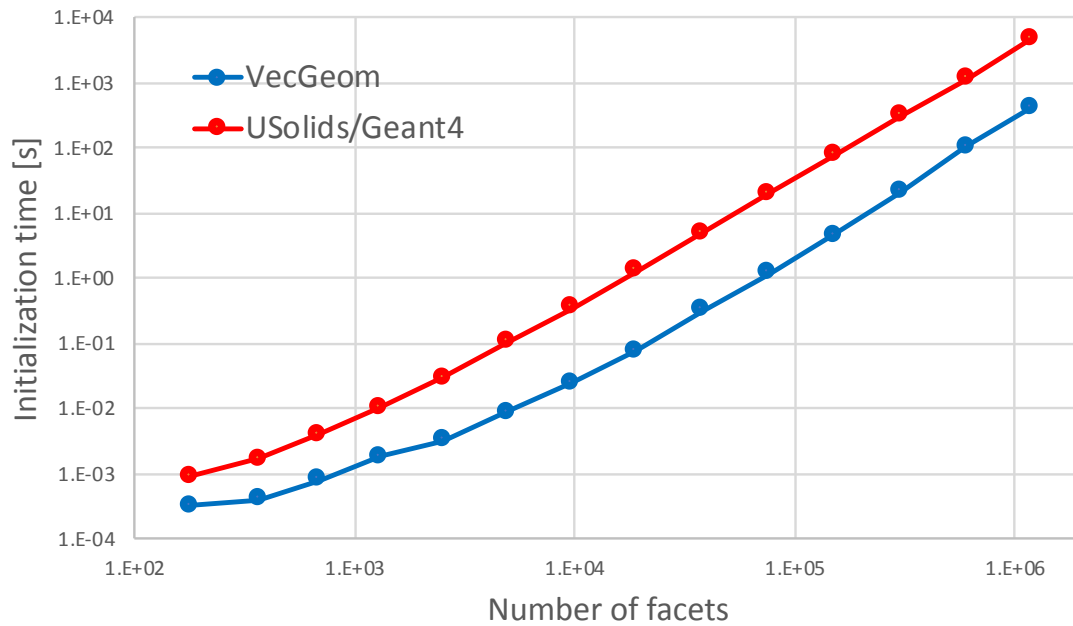
# Tessellated solid performance

- $O(10)$  speed-up compared to Geant4 in both initialization and run time for up to 100K facets
- Next step: improve of scalability for very large number of facets/components
  - Vectorized approaches are available in industry ray tracers (such as [Intel Embree](#))
  - Trading precision for speed in some calculations (isotropic safe distance)

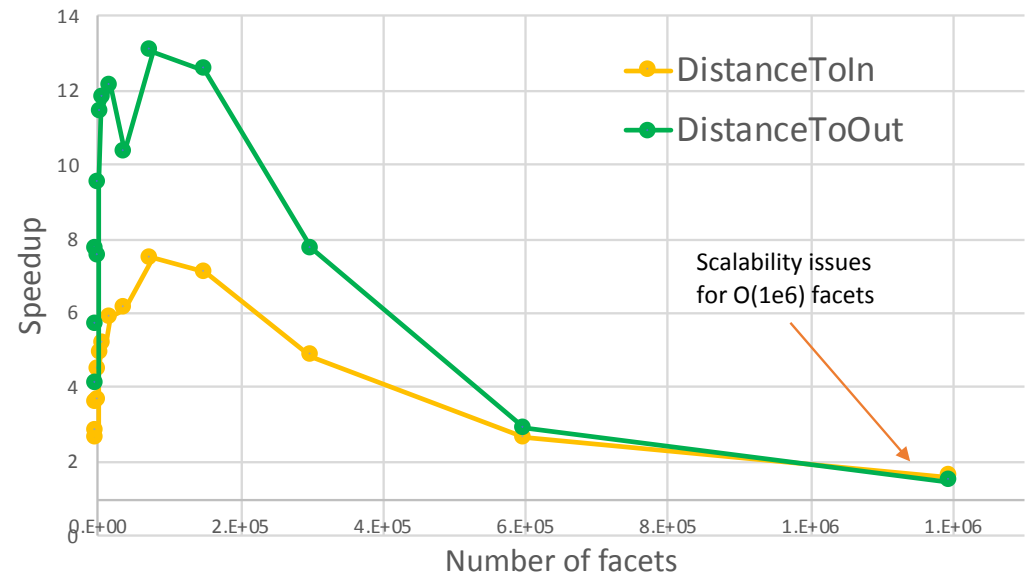


AVX double

Initialization time scalability



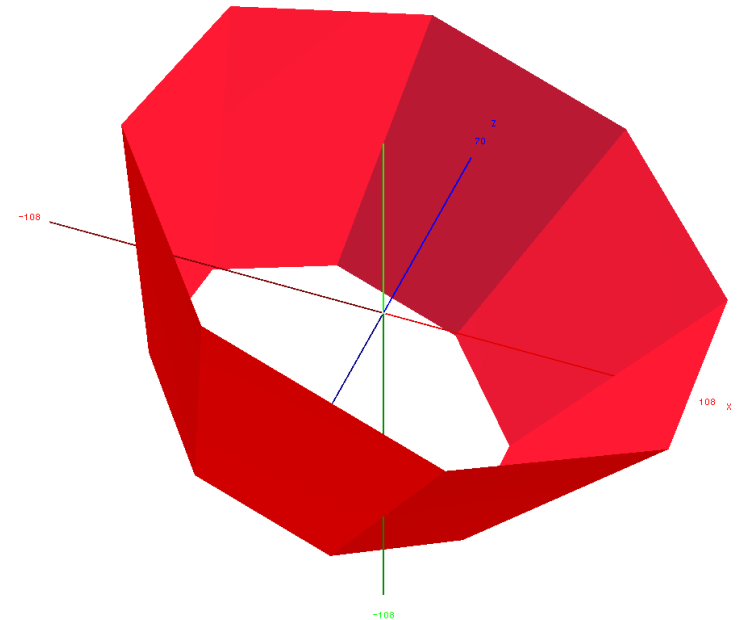
Speed-up of DistanceToIn/Out compared to previous USolids/Geant4 version



M.Gheata

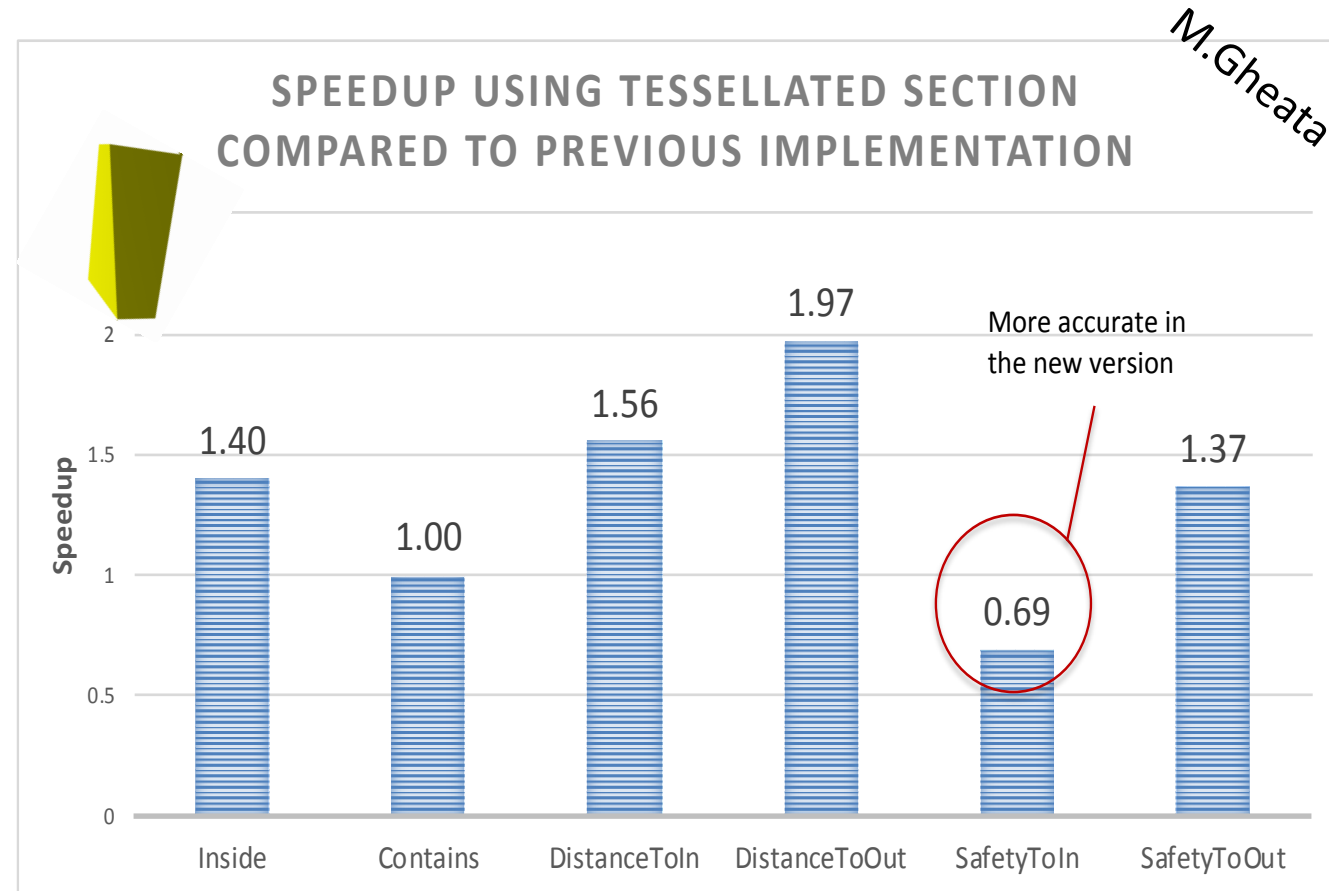
# Extending multi-faceted approach to other solids

- Generalized to more multi-faceted solids
  - Trapezoids, polyhedra, extruded solids
- Created a new SIMD helper class
  - Representing a surface made of quadrilateral tile: organized in clusters of size = vector length
  - Delimited by two Z planes
  - Using explicit vectorization on tiles based on VecCore



# New Generic Trapezoid (Arb8)

- No benefits from using tessellated sections measured for simple solids (box, simple trapezoids)
  - Faster without loops
- Other solids already vectorized in the scalar case -> marginal gain
  - Polyhedron, Trapezoid
- Important improvements observed for previously non vectorized cases
  - Generic trapezoid (Arb8)
  - Extruded solid

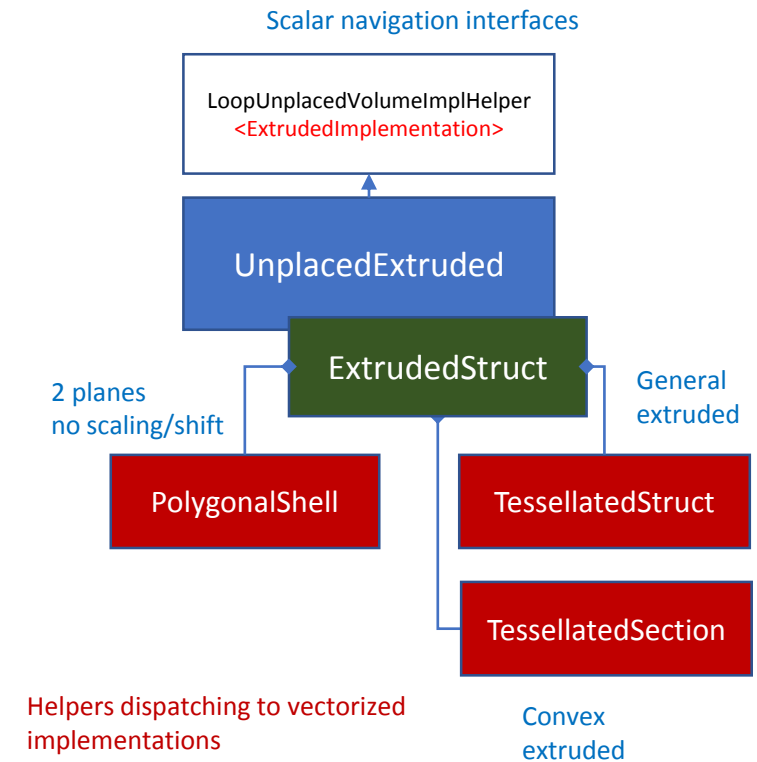


# Extruded Solid

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

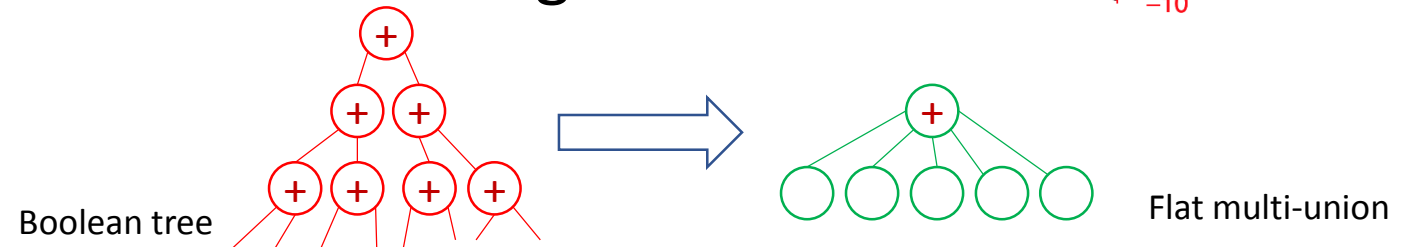
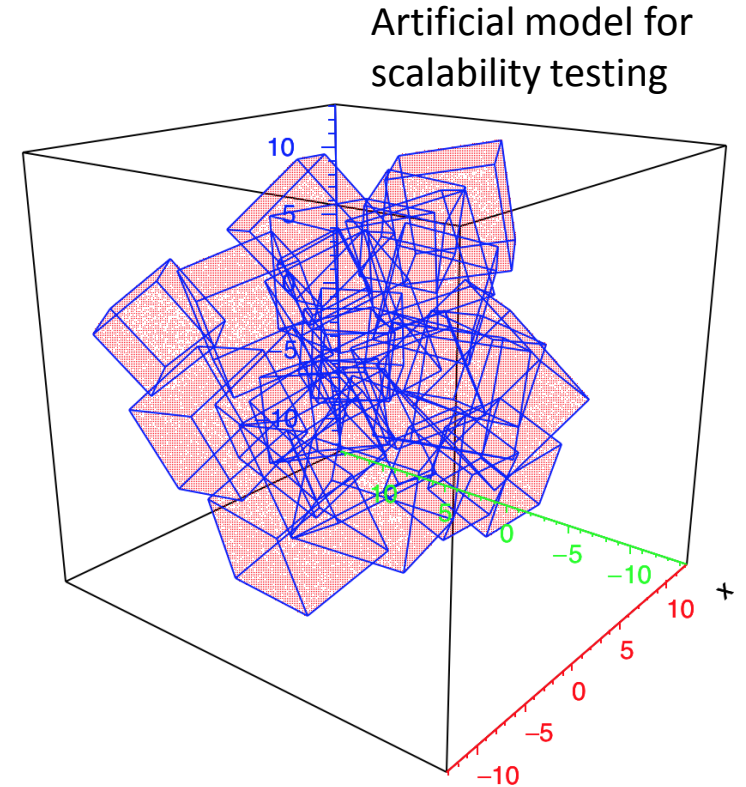


M.Gheata



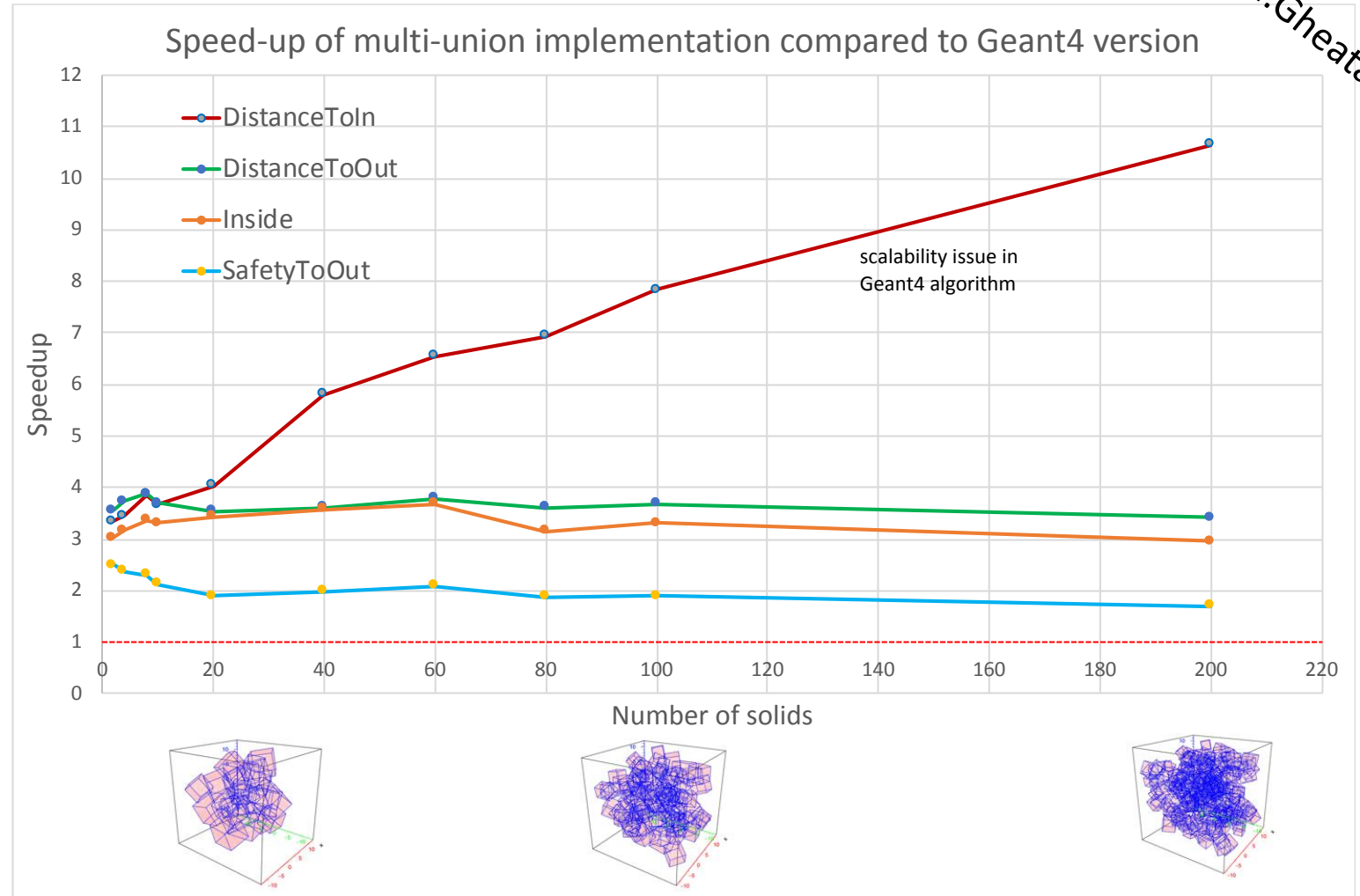
# Multi-union structure

- Boolean unions are represented as binary trees
  - Pathologically slow in simulation - too many individual checks
- Multi-unions representing volumes at same level with same material
  - Optimization structures to limit selection of candidates
  - Implementation in Geant4 using voxelization helper
- Re-implemented in VecGeom based on Bounding Volume Hierarchies (BVH)
  - Vectorized search of candidates



# Benchmark of Multi-Union in VecGeom

Large speed-up compared to the original Geant4/USolids implementation for up to several hundred components

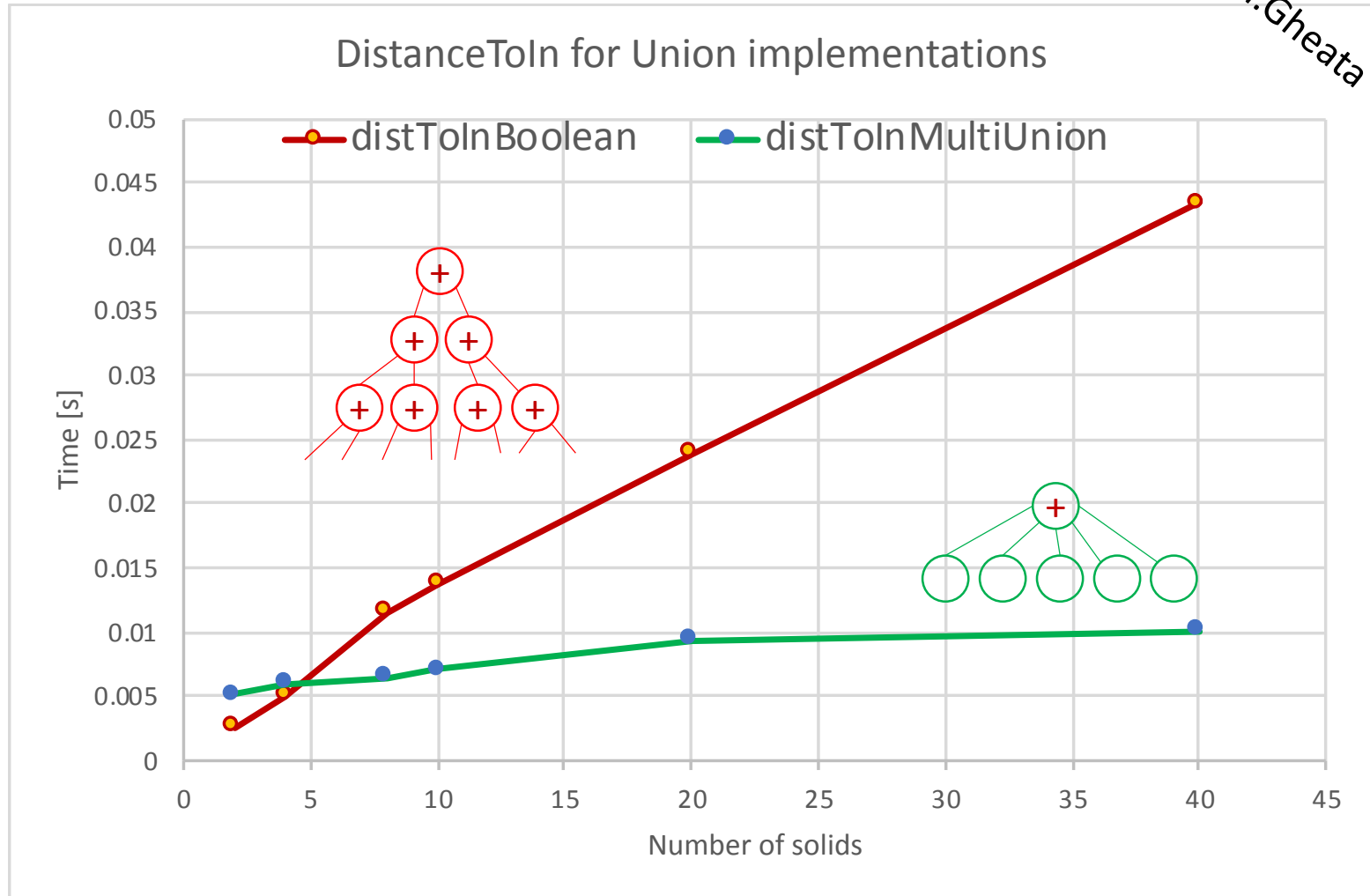


M.Gheata

# Replacing Boolean union with multi-union

M.Gheata

- Implemented VecGeom automatic conversion of Boolean volumes to the new multi-union structure
  - Far better performance and scalability for large number of components



# Latest VecGeom releases and features

- Consolidated version of VecGeom v00.05.01
  - To be used with Geant4 10.4 series
- New VecGeom version v01.00.00
  - Removed deprecated USolids module and interfaces
  - Cleaned up code, tests and package configuration
  - VecGeom solids called directly from Geant4 wrappers
  - To be used with Geant4 10.5-beta
  - No code migration required for users
- Ongoing implementation of a templated factory for automatically building specialization of shapes
  - Shapes featuring this so far:
    - Tube, cone, polycone, trap, trd, hype, extruded solid (convex/concave/simple-extru)

# 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 \  
  -DVC_ACCELERATION=OFF -DVECGEOM_VECTOR=avx2 \  
  [...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**

# Expecting in Geant4 10.5

## Geometrical primitives

- Removed USolids module and deprecated interfaces
- Improved implementation for Generic Trapezoid
- New wrapped shapes:
  - Hyperboloid, Tessellated Solid, Extruded Solid, Tet, Multi-Union structure

## Navigation

- Prototype study of specialized navigator with VecGeom
  - Applicable to placed/static geometries only
  - First attempt based on 'smart' cloning of navigation histories
  - Evaluation of performance