

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Integration of VecGeom into Geant4

Guilherme Lima (Fermilab) for the VecGeom Group

Geant4 Collaboration Meeting Fermilab – September 30, 2015

Outline

- Note: VecGeom has already been introduced by Sandro in previous presentation
- In this talk:
 - VecGeom integration
 - what is it?
 - why is it needed?
 - how is it done?
 - Results
 - callgrind on a simple geometry
 - full CMS detector
 - How to use VecGeom shapes with Geant4
 - Current status
 - Summary

VecGeom integration with Geant4

• What is it?

It means that we can run "any Geant4 job" using solids and algorithms from the VecGeom library

- ...uses VecGeom shapes when available, if not uses the USolids versions then the Geant4 ones
- no changes are needed in user/application code other than turning on a few compilation switches
- VecGeom shapes: only scalar algorithms are used no parallelized SIMD processing of tracks!



VecGeom integration with Geant4

• Why is it needed?

- It allows direct, high-level (physics) comparisons, for validation of solids and their algorithms, and for optimization purposes (Geant4 / USolids / VecGeom)
- use of well tested Geant4 tools for testing, logging, debugging, validation, visualization and so on
- make VecGeom shapes immediately and transparently available to any Geant4 applications
- *expected* to provide performance gains with respect to both USolids and Geant4 shapes – to be verified!
- promotes widespread adoption, feeding back into further developments and contributions



VecGeom integration with Geant4

• How is it done?

- VecGeom was designed to be USolids-compatible
- Existing USolids interface was used, e.g.
 G4Box → G4UBox → UBox → (VecGeom) SimpleBox
- USolids shapes were *extended* to either use USolids source code, or to define USolids shapes as inheriting from the VecGeom shapes, depending on whether a pre-processor macro is defined from compilation command lines:

```
g++ [...] -DVECGEOM_REPLACES_USOLIDS [...]
```

- if that macro is not defined, current USolids shapes and algorithms are used
- all navigation is still performed and controlled by Geant4
- boolean operations with solids still managed by Geant4



Example: the integrated UBox

```
#ifndef USOLIDS UBox
#define USOLIDS UBox
                                          Macro that enables the use of VecGeom shapes
#ifdef VECGEOM REPLACE USOLIDS
#include "volumes/SpecializedBox.h"
#include "volumes/LogicalVolume.h"
#include "volumes/UnplacedBox.h"
#include "base/Transformation3D.h"
                                             Most shapes only needed this part.
class UBox: public vecgeom::SimpleBox {
 // just forwards UBox to vecgeom::SimpleBox
                                             Only two shapes needed any
 using vecgeom::SimpleBox::SimpleBox;
                                             extra declarations.
public:
 // add default constructor for tests
 UBox() : vecgeom::SimpleBox(new vecgeom::LogicalVolume(new vecgeom::UnplacedBox()),
                          &vecgeom::Transformation3D::kIdentity, this) {}
}:
//========= end of VecGeom-based implementation
#else
#include "VUSolid.hh"
                                             Existing Usolids used when flag is not enabled
class UBox : public VUSolid
 /// ... existing UBox declarations ...
};
//========= end of USolids-based implementation
     // VECGEOM_REPLACE_USOLIDS
#endif
#endif // USOLIDS UBox
                                                                               Fermilab
                                                                    2015/09/30
                 G.Lima | Geant4 Collaboration Meeting
```

6

CallGrind graphs: using Geant4 shapes

lat <u>P</u> ro	file		@ X	G4Polyhedra::Inside(CLHEP::Hep3Vector const&) const
earch:	Inside		Source File 🛟	
Self	Sour	ce File		
89.32 📕 (unknown) (12)				G4PVPlacement::CheckOverlaps(int,
				double, bool, int)
				3 002 x
	c. 10			G4Polyhedra::Inside(CLHEP::Hep3
0.02	SelF	Called Function 3 002 G4Polyhedra::Inside(CLHEP::Hep3Ved	tor const&) const	Vector const&) const
0.0	1 0.01	6 050 G4Tubs::Inside(CLHEP::Hep3Vector co	onst&) const	0.02 /0
0.0	0.00	79 G4VCSGFaceted::Inside(CLHEP::Hep3	Vector const&) cons	│ □ 3 002 x □ 79 x
0.0	1 0.00	316 G4PolyhedraSide::Inside(CLHEP::Hep	3Vector const&, dou	*
0.00	0.00	5 006 G4Cons::Inside(CLHEP::Hep3Vector co	onst&) const	:losingCylinder::MustBeOutside(G4VCSGfaceted::Inside(CLHEP::Hep3
0.00	0.00	1 035 G4Sphere::Inside(CLHEP::Hep3Vector	const&) const	EP::Hep3Vector const&) const Vector const&) const
0.00	0.00	2 006 G4Box::Inside(CLHEP::Hep3Vector co	nst&) const	0.01%
0.00	0.00	158 G4PolyPhiFace::Inside(CLHEP::Hep3V	ector const&, double	
0.00	0.00	1 004 G4Orb::Inside(CLHEP::Hep3Vector co	nst&) const	
0.00	0.00	160 G4PolyPhiFace::InsideEdges(double, o	double, double double	G4PolyPhiFace::Inside(CLHEP::Hep3 G4PolyhedraSide::Inside(CLHEP::Hep3
0.00	0.00		ubie, doubie, doubie	Vector const&, double, double*) Vector const&, double, double*)
				0.00 %
				1316X
				C4DelubedraCideuCetDbi/CLUEDuLlea2
Note: These may be unreadable on			e on 🔰	Vector const&) CLHEP::Hep3Vector const&. G4
the conference room projection, but			but	0.00 %
it is a solid to the solid to so from the				
It is possible to read them from the			the	
PDF file on your own screen!				
				atan2
. (Parts Callees Call Graph All Callees Caller Map Machine Code
			10	

‡Fermilab

2015/09/30

geant4.callgrind.out [1] - Total Instruction Fetch Cost: 2 150 555 330

CallGrind graphs: using VecGeom shapes

Flat <u>P</u> ro	file			ð X	vecgeom::cxx::ScalarShapeImuble> const&, double) const
Search:	Distance	Toln	Sour	rce File 💲	
Self	Source	e File			G4VoxelNavigation::ComputeStep(CLHEP::Hep3Vector const&, CLHEP::He
83.	19 🔳 (uni	known) (12)		G4USolid::DistanceToIn(CLHEP::Hep3 Vector const&, CLHEP::Hep3Vector co
Incl.	Self	Called	Function		
0.04	0.01	10 016	G4USolid::DistanceToIn(CLHEP::Hep3Vector const&) const		32 x
0.01	0.00	898	G4USolid::DistanceToIn(CLHEP::Hep3Vector const&, CLHEP::Hep3Vector const&) const		★
0.00	0.00	245	vecgeom::cxx::ScalarShapeImplementationHelper <vecgeom::cxx::coneimplementation<-1, -1,="" vecgeom::cxx::scalarshapeimplementationhelper<vecgeom::cxx::torusimplementation2<-1,="">:Di</vecgeom::cxx::coneimplementation<-1,>	istanceToIn(ve	vecgeom::cxx::ScalarShape
0.00	0.00	257	vecgeom::cxx::ShapeImplementationHelper <vecgeom::cxx::paraboloidimplementation<-1 -1=""> >::Dist</vecgeom::cxx::paraboloidimplementation<-1>		ImplementationHelper <vecgeom::cx< td=""></vecgeom::cx<>
0.00	0.00	121	vecgeom::cxx::ShapeImplementationHelper <vecgeom::cxx::trdimplementation<-1, -1,="" p="" vecgeom::cxx::<=""></vecgeom::cxx::trdimplementation<-1,>	:TrdTypes::Univ	0.00 %
0.00	0.00	32	vecgeom::cxx::ScalarShapeImplementationHelper <vecgeom::cxx::polyhedronimplementation<(einne< td=""><td>erRadii)0, (EPh</td><td></td></vecgeom::cxx::polyhedronimplementation<(einne<>	erRadii)0, (EPh	
0.00	0.00	35	vecgeom::cxx::ShapeImplementationHelper <vecgeom::cxx::sphereimplementation<-1, -1="">>::Distance</vecgeom::cxx::sphereimplementation<-1,>	eToIn(vecgeoi	
0.00	0.00	89	vecgeom::cxx::ShapeImplementationHelper <vecgeom::cxx::tubeimplementation<-1, -1,="" p="" vecgeom::cxx<=""></vecgeom::cxx::tubeimplementation<-1,>	x::TubeTypes::	
0.00	0.00	46	vecgeom::cxx::ShapeImplementationHelper <vecgeom::cxx::orbimplementation<-1, -1="">>::DistanceTo</vecgeom::cxx::orbimplementation<-1,>	oln(vecgeom::c	vecgeom::cxx::kScalar::bool_v
0.00	0.00	2	vecgeom::cxx::ShapeImplementationHelper <vecgeom::cxx::boximplementation<-1, -1="">>::DistanceTo</vecgeom::cxx::boximplementation<-1,>	oin(vecgeom::c	
0.00	0.00	1	void vecgeom::cxx::Wedge::DistanceToIn <vecgeom::cxx::kscalar>(vecgeom::cxx::Vector3D<vecgeom::< p=""></vecgeom::<></vecgeom::cxx::kscalar>	::cxx::kScalar::p	
					60 x
					+
					void vecgeom::cxx::Wedge::Generic KernelForContainsAndInside <vecgeo< td=""></vecgeo<>
		ы	ata: Thasa may ba upraadabla an		0.00 %
		IN	ole. These may be unreadable on		
		th	e conference room projection, but		120 x
		it	is possible to read them from the		vecgeom::cxx::kScalar::bool_v
		P	DF file on your own screen!		vecgeom::cxx::Wedge::IsOnSurface
					((())))
GC					Parts Callees Call Graph All Callees Caller Map M 41
Vacana	a calloring	d out [1]	- Total Instruction Eatch Cost: 2 E10 789 200	10	



How can I use VecGeom shapes within Geant4?

• Overview

- Step 1: build VecGeom and Usolids libraries
- Step 2: build Geant4 libraries linked to both USolids and VecGeom libraries
- Step 3: build your Geant4 application

Tip: CMake takes care of carrying the configuration over from one step to the next



- Step 1: build USolids and VecGeom libraries
 - VecGeom provides the USolids library configured to use VecGeom shapes and algorithms

```
cmake -DBACKEND=Scalar -DGEANT4=OFF -DUSOLIDS=ON -DUSOLIDS_VECGEOM=ON \
  [...other optional VecGeom switches as needed...] \
  -DCMAKE_INSTALL_PREFIX=${VGINSTALLDIR} ${VGSOURCE}
```

```
make install
```

- Notes:
 - Using "make install" is important to properly install libraries, header files and configuration scripts, to help next steps to find them
 - Only scalar backend can be used by Geant4 (one track at a time)
 - -DGEANT4=OFF is required to avoid circular dependencies, as -DGEANT4=ON would introduce Geant4 dependencies in VecGeom
 - If <u>-DUSOLIDS_VECGEOM=ON</u> is omitted, USolids shapes and algorithms will be used instead of VecGeom ones.



- Step 2: build Geant4 libraries
 - Geant4 must be compiled with switches to enable the *external USolids library* provided by VecGeom

```
export Usolids_DIR=${VGINSTALLDIR}/lib/Cmake/Usolids/

cmake -DCMAKE_INSTALL_PREFIX=${G4INSTALL} \
	-DGEANT4_BUILD_CXXSTD=c++11 \
	-DGEANT4_USE_USOLIDS=ON -DGEANT4_USE_SYSTEM_USOLIDS=ON \
	-DGEANT4_INSTALL_DATADIR=${G4INSTALL}/share/Geant4-10.2.0/data \
	-DGEANT4_USE_GDML=ON -DGEANT4_BUILD_MULTITHREADED=OFF \
	${G4SOURCE}
```

```
make install
```

- Notes:
 - C++11 is required by VecGeom
 - Switch -DGEANT4_USE_SYSTEM_USOLIDS=ON will force the use of an external USolids library, rather than the one which comes with Geant4
 - Few modifications to Geant4 code were needed, already included in the devel version ref-08, to be released with version 10.2 (December)

🏞 Fermilab

2015/09/30



- Step 3: build your Geant4 application
 - Tell CMake where to find the Geant4 libraries from step 2 and build it

```
cmake -DGeant4_DIR=${G4INSTALL}/lib/Geant4-10.2.0 ${SRCDIR}
```

make

 More complete instructions available in subdirectory doc of VecGeom repository

https://gitlab.cern.ch/VecGeom/VecGeom



Testing with a complex detector

- FullCMS test in geant4 tests repository
- Built agains geant4.10.2.beta
- CMS geometry loaded from a GDML file
 - some error messages due to rounding in GDML data
 - during simulation, hints of overlapping volumes
- Job runs for some time, then crashes
 - debugging is under way, some pathological bugs identified and fixed
- Plan to use it for performance comparison
 - hopefully in a time scale of a couple weeks



Full CMS tests

- Identified some missing features
 - e.g. points on surface from polycone's Inside()
- Some pathological cases tracked and fixed
 - negative Rmin in polycone bounding tube, negative safeties, missing tolerances, ...
- Geant4 tools used extensively, specially navigation checks



polycone's bounding tube



tolerance added to Rmax and subtracted from Rmin – *if non-zero*



VecGeom integration – Status

Current status

- first batch of shapes available
- choice of shapes based on a full CMS model
- ready: box, trapezoid, tube, cone, polycone, polyhedra, with native phi-wedges and Rmin where appropriate
 - tested under FullCMS conditions
- almost ready: orb, sphere, trd, torus, paraboloid
 - not fully tested yet, since not used for FullCMS
 - trd and torus are part of a newer cms2015 model
- visualization is not available yet
 - CreatePolygon() required for each shape
- FullCMS tests under way, expected soon



Summary

• It is possible to use VecGeom shapes from Geant4

- integration code is very new
 - it will go through review before being released, but the concept has been proved
 - once validated, VecGeom shapes will replace corresponding USolids shapes
- nice leverage of Geant4 testing and validation tools
- no changes needed in user application code
- final stages of testing on a full CMS model
- shapes ready: box, trapezoid, tube, cone, polycone, polyhedra, with native *phi*-wedges and *Rmin* where appropriate
 - partially tested under FullCMS conditions
- shapes almost ready: orb, sphere, trd, torus, paraboloid
 - not fully tested yet, since not used for FullCMS