



---

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

---

# **Geant[V,X] Geometry**

**Ph. Canal, S.Y. Jun, G. Lima (Fermilab)**

Geant R&D Retreat

September 13, 2019

# VecGeom geometry

# VecGeom maintenance

- Periodic checking of geometry shapes
  - benchmarks and ShapeTester tests for individual shapes
  - quick fixes or file JIRA tickets
- User support and investigation of bug reports
  - g-2 report on inconsistent use of USolids (James Stapleton)
    - problem on the redefinition of building environment
  - Mu2e report (Andrei Gaponenko et.al.)
    - bug on the G4ExtrudedSolid, fixed for next releases
- Extending FPEExceptions to all shape unit tests
- Torus updates (see next)
- Normal benchmark
- Navigation benchmark on GPU (more on this later)
- Other JIRA issues related to geometry

# Torus Updates

- Aldo Miranda was trying to vectorize algorithms, and also to improve the stability of torus functions by adding alternative parametrization to be used when current one does not work well
- Current status: unit tests and benchmarks ok, but detailed tests find some problems
- Several git branches with partial vectorization work, which I am trying to clean up and consolidate
- Easier path is to completely abandon the vectorization efforts, and just fix the current problems.
- I was trying to keep the vectorization progress, but it may be a lot of effort to finish it...

# Navigation benchmark

- Navigation benchmark developed by Sandro, was extended to work on the GPU (few years ago)
- Sandro forked it and added more functionality, to test several new navigators, but with no GPU support
- GPU compatibility is desirable again in the context of the Exascale Proxy project
  - Fix/update the navigation on the GPU

# Navigation benchmark overview

- Exercises navigation and performance assessment
  - Build a 4-level-deep, VecGeom-based geometry
  - Convert it to ROOT and Geant4
    - ROOT can be used for geometry visualization
  - Generate 1000 tracks (points+directions) inside a user-provided volume
  - Navigation validation
    - for each track, LocatePoint() and FindNextBoundaryAndStep()
    - compares results between GeantV, Geant4 and/or Root using validateNavigationStepAgainstRoot() and validateNavigationStepAgainstGeant4()
  - Navigation benchmark
    - CPU time measurements compared
- Current status
  - Validations runs, shows many mismatches against ROOT, crashes during benchmark, trying to access deprecated function SafetyToInMinimize().
  - Sandro's NavigationKernelBenchmark also crashes, but runs tests with several different navigators.

# NavigationBenchmark - current status




```
*** ERROR on validateAgainstROOT: ROOT node=trap_6 outside=0 step=2.64127 <==> VecGeom node=trap_6 step=2.64128 /// Step ratio=1 / step diff=1e-06 / rel.error=3.78605e-07  
0x7f7f7887a3b0 0x7f7f7887a3b0  
  
=====> Summary: ITERATION 996 - pos = (-0.886549, -6.43399, -4.20065) dir = (0.714257, 0.697294, 0.0601486) / Steps (Root/VecGeom): 2.64127 / 2.64128  
Next volumes: trap_6 / trap_6  
  
*** ERROR on validateAgainstROOT: ROOT node=trap_0 outside=0 step=0.223229 <==> VecGeom node=trap_0 step=0.22323 /// Step ratio=1 / step diff=1e-06 / rel.error=4.47969e-06  
0x7f7f78879750 0x7f7f78879750  
  
=====> Summary: ITERATION 998 - pos = (4.80966, 9.13114, 0.947815) dir = (0.668894, -0.587466, 0.455483) / Steps (Root/VecGeom): 0.223229 / 0.22323  
Next volumes: trap_0 / trap_0  
  
*** ERROR on validateAgainstROOT: ROOT node=trap_2 outside=0 step=0.179911 <==> VecGeom node=trap_2 step=0.179912 /// Step ratio=1.0001 / step diff=1e-06 / rel.error=5.55827e-06  
0x7f7f7887a170 0x7f7f7887a170  
  
=====> Summary: ITERATION 999 - pos = (9.13749, -8.10389, 1.70957) dir = (-0.764209, -0.0898739, 0.638676) / Steps (Root/VecGeom): 0.179911 / 0.179912  
Next volumes: trap_2 / trap_2  
VecGeom navigation - serial interface: # ROOT mismatches = 607 / 1000  
libc++abi.dylib: terminating with uncaught exception of type std::runtime_error: SafetyToInMinimize unimplemented  
Abort trap: 6  
Lima@mac-131320: build 🍺
```

Validation function runs, shows many mismatches against ROOT, crashes during benchmark, trying to access deprecated function `SafetyToInMinimize()`.

# NavigKernelBenchmark - current status

Sandro's  
NavigationKernelBenchmark  
also crashes, but runs  
several tests with several  
different navigators.

Could be useful for other  
navigation developments  
and tests, e.g. GPU porting.

```
lima@mac-131320: build  ./NavigationKernelBenchmark navBench.root world
Info in <TGeoManager::Import>: Reading geometry from file: navBench.root
Info in <TGeoManager::CloseGeometry>: Geometry loaded from file...
Info in <TGeoManager::SetTopVolume>: Top volume is world. Master volume is world
Info in <TGeoNavigator::BuildCache>: --- Maximum geometry depth set to 100
Info in <TGeoManager::CloseGeometry>: Voxelization retrieved from file
Info in <TGeoManager::CountLevels>: max level = 3, max placements = 8
Info in <TGeoManager::CloseGeometry>: 25 nodes/ 4 volume UID's in default geometry
Info in <TGeoManager::CloseGeometry>: -----modeler ready-----
*** Conversion of ROOT -> VecGeom finished (0.000965918 s) ***
*** Closing VecGeom geometry finished (0.000110622 s) ***
NavigationKernelBenchmark run on world having 8 daughters
Uncontained capacity for world_1: 3904 units
PlacedVolume created after geometry is closed --> will not be registered
VolumeUtilities: FillBiasedDirs: nhits/size = 288118/500000 and requested bias=0.400000
  really hits 200000, virtually hits 200000
  points filled
  located ...
##
##
2.01175
accum NewSimpleNavigator 1.44286e+06 target checksum 1252172
##
1.94259
accum NewSimpleNavigator 1.44286e+06 target checksum 1252172
##
1.17219
accum SimpleABBoxNavigator 1.44286e+06 target checksum 1252172
##
1.18649
accum SimpleABBoxNavigator 1.44286e+06 target checksum 1252172
##
1.56486
accum HybridNavigator 1.44286e+06 target checksum 1252172
##
1.48054
accum HybridNavigator 1.44286e+06 target checksum 1252172
##
0.21291
Assertion failed: (i != b_to_a.end()), function operator[], file /Users/Lima/work/vecgeom/base/TypeMap.h, line 56.
Abort trap: 6
lima@mac-131320: build  
```



# VecGeom: navigation on the GPU?

- started extending navigation into the GPU
  - complex geometry uploaded
    - mother-daughter hierarchy verified
  - navigation implementation
    - ~2015: mostly working, but being debugged
    - suspended due to navigation interface changes from navigation state arrays to IndexedNavStates
    - currently updating, fixing problems on the CPU
    - next:
      - review the navigation infrastructure on the GPU
      - update VecGeom tests to run on the GPU, if reasonable

# GeantX geometry

# GeantX in Docker containers

- It's easy to build and run GeantX binaries on a docker container:

- install docker, and then run it using a command like this:

```
cd ~/work/ecp/pilot
docker run -it \
  -v ${HOME}:/home/lima \
  -v ${HOME}/.Xauthority:/root/.Xauthority:rw \
  -v /tmp/.X11-unix:/tmp/.X11-unix \
  -e DISPLAY="192.168.177.120:19.0" \
  --net=host \
  geantx/geantx-dev
```

- Nice common development environment:
  - Ubuntu 18.04.2, with gcc 7.3.0, cmake 3.10.2, git 2.17.1 pre-installed
  - All GeantX dependencies like veccore (Vc), vecmath and vecgeom
- Unfortunately, no docker support on the Wilson cluster (GPU access)
- tev-admin suggested to use SyLab's Singularity (developed at LBL(?), similar to docker, with more strict security to comply with DoE policies)
- could not run it after some weeks, decided to build geantx locally...

# GeantX for the Exascale machine

- Using VecGeom geometry in the GPU
  - Started from TestEm3
  - removed everything not related to geometry
  - brought in code to upload a non-trivial geometry to the GPU
  - added VecGeom libs and headers to CMake files

```
lima@mac-131320: pilot 🍷 git show 607ae0e --name-only
commit 607ae0e6de198138143ebb895c4e810d663a6751
Author: Guilherme Lima <lima@fnal.gov>
Date: Tue May 14 15:18:51 2019 -0500

    Adding a geometry test into examples/geometry/GeometryTest.cpp

examples/CMakeLists.txt
examples/geometry/CMakeLists.txt
examples/geometry/GeometryTest.cpp
examples/geometry/source/TestEm3DetectorConstruction.cpp
examples/geometry/source/TestEm3DetectorConstruction.hpp
source/Geant/CMakeLists.txt
source/Geant/geometry/RunManager.hpp
source/Geant/geometry/UserDetectorConstruction.cpp
source/Geant/geometry/UserDetectorConstruction.hpp
lima@mac-131320: pilot 🍷
```

# Uploading geometry to the GPU

```
// Create detector TestEm3 construction
userapplication::TestEm3DetectorConstruction *det =
    new userapplication::TestEm3DetectorConstruction(runMgr);
// Temporary workaround to allow migration to detector construction
geantx::vector_t<geantx::Volume_t const *> volumes;
int numVolumes = 0;
if (det) {
    det->CreateMaterials();
    det->CreateGeometry();
    numVolumes = det->SetupGeometry(volumes);
}
int maxDepth = vecgeom::GeoManager::Instance().getMaxDepth();
geantx::Log(geantx::kInfo).From("main") << "Geometry created with maxdepth" << maxDepth;
geantx::Log(geantx::kInfo).From("main")
    << "Geometry created with " << numVolumes << " volumes and maxdepth " << maxDepth;

// print detector information
det->DetectorInfo();

// CUDA stuff
#ifdef VECGEOM_ENABLE_CUDA
// vecgeom::cxx::CudaManager::Instance().set_verbose(3);
vecgeom::cxx::CudaManager::Instance().LoadGeometry();
vecgeom::cxx::CudaManager::Instance().Synchronize();

vecgeom::cxx::CudaManager::Instance().PrintGeometry();

std::cout << std::flush;
#endif
```

← Used to verify the geometry  
uploaded to the GPU

# Verifying the geometry on the GPU

- Verifying the geometry:
  - From the GPU, call a recursive printing function which shows a volume name, and then calls the same function for each daughters. The whole geometry hierarchy is verified.
- Next step:
  - create navigation tests for the GPU, adapting VecGeom navigation tests discussed earlier for the GeantX environment.
  - First, need to fix those navigation tests on the VecGeom side.

```
HYBRID SETUP TOOK 7.1143e-05s
>>> main :Geometry created with maxdepth4
>>> main :Geometry created with 5 volumes and maxdepth 4

===== Detector Info =====
Calorimeter is 50 layers of [ 2.3 mm NIST_MAT_Pb + 5.7 mm NIST_MAT_lAr ]
=====

ScanGeometry found pvolumes54
Allocating placed volume NUMBER OF PLACED VOLUMES 54
NUMBER OF UNPLACED VOLUMES 5
Geometry loaded on GPU:
[53]:
Transformation3D {{0.00, 0.00, 0.00}, {1.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00}}
LogicalVolume [-1]:
UnplacedBox {24.00, 24.00, 24.00}
Contains %zu daughter:
[52]:
Transformation3D {{0.00, 0.00, 0.00}, {1.00, -0.00, 0.00, 0.00, 1.00, -0.00, 0.00, 0.00, 1.00, -0.00, 0.00, 0.00}}
LogicalVolume [-1]:
UnplacedBox {20.00, 20.00, 20.00}
Contains %zu daughters:
[2]:
Transformation3D {{-19.60, 0.00, 0.00}, {1.00, -0.00, 0.00, 0.00, 1.00, -0.00, 0.00, 0.00, 1.00, -0.00, 0.00, 0.00}}
LogicalVolume [-1]:
UnplacedBox {0.40, 20.00, 20.00}
Contains %zu daughters:
[0]:
Transformation3D {{-0.29, 0.00, 0.00}, {1.00, -0.00, 0.00, 0.00, 1.00, -0.00, 0.00, 0.00, 1.00, -0.00, 0.00, 0.00}}
LogicalVolume [-1]:
UnplacedBox {0.11, 20.00, 20.00}
```

# Perspectives

- Expect faster progress after GeantV performance studies are completed
- Not sure how much development still needed for VecGeom navigation on the GPU
- Plans to create similar tests as for GeantV
  - TestEm3, TestEm5, FullCMS, FullLHCb