Simulation-aided optimization of detector design using portable representation of 3D objects
Outline

• Describe traditional approach

• Contrast STL-based approach
  • STL pros
  • STL cons

• ‘plumbing’: CADMesh, VMs

• Example: DarkLight detector design

• Conclusions
Typical 2-prong detector design process

Physicists
Geant model

Engineers
CAD model
Typical detector design process ... takes time

Physicists
bring idea

- purpose of experiment
- key components
- preferred technology
- space constraints

• simulated response
→ changed requirements

• high cost
→ changed requirements

Engineers
bring technical knowledge

- detailed construction plan
- realistic material budget
- realistic price tag

• revised design & price tag

• revised design & price tag
**Geant4 representations of solids**

**Constructive Solid Geometry (CSG)**

**Boundary Represented Solids (BREPs)**

---

**CAD ⇒ Standard Tessellation Language (STL)**

**Geant4**: to import sphere in **STL format** use:

```c++
CADMesh * mesh = new CADMesh("mySpehre.stl", "STL", ...)
G4VSolid * mySolid = mesh->TessellatedMesh();
```

---

To build a **full solid sphere** use:

```c++
G4VSolid *
G4Orb(const G4String& pName, G4double pRmax)
```
Streamlined sharing of detector design

Physicists exchange geometry using Standard Tessellation Language (STL)

Other tools which can be used to generate meshes to be then imported in Geant4 as tessellated solids are:

- **STL2GDML** - A free STL to GDML conversion tool.
- **SALOME** - Open-source software allowing to import STEP/BREP/IGES/STEP/ACIS formats, mesh them and export to STL.
- **ESABASE2** - Space environment analysis CAD, basic modules free for academic non-commercial use. Can import STEP files and export to GDML shapes or complete geometries.
- **CADMesh** - Tool based on the VCG Library to read STL files and import in Geant4.
- **Cogenda** - Commercial TCAD software for generation of 3D meshes through the module gds2mesh and final export to GDML.
Advantages of STL over G4-native (CSG)

- no ‘logical disassembly’ into TBox, TTube, etc. of complex, real life shapes
- portability between Geant4 projects (similar to GDML)
- STL fidelity (accuracy) can be changed while exporting from CAD

(subjective)
**Advantages of CAD over Geant4-native detector design**

- intuitive assembly in 3D of non-trivial volumes and their alignments
- high quality graphics and animations - important for public relations
**Requirement:**
design GEM detector (1) made out of frames, (2) placed it at an angle to the beam, (3) fitting it in to a tube

a) Slice a cylinder with the plane inclined to the beam  
b) define one GEM frame (green)
Assemble a non-trivial volume

c) assemble 6 ‘planes’ of GEMs

d) export GEM frame to Geant4 and re-assemble

e) run Geant4 simulations

f) conclude: acceptance for coincidence is bad

g) drop this design

Total time spent: 1 week
**Requirement:**
this blower sits in front of your detector and obstructs measured particles. Simulate its impact.

How would you implement it in Geant4 using :TBox, TTube, ..., etc?

STL representation of the blower exported directly from CAD

Geant4 implementation time : ~ hour
Volumes conflict detection, inspection

Checking overlaps for volume TpcSens15 ...
*** G4Exception : GeomVol1002
   issued by : G4PVPlacement::CheckOverlaps()
Overlap with volume already placed!
   Overlap is detected for volume TpcSens15
   with TpcSens14 volume's
   local point (-116.344,198.404,441.614), overlapping by at least: 2 mm
*** This is just a warning message. ***

Problem easier to diagnose

Sometimes you just want to know:

AutoCAD

Geant 4
Limitations of STL geometry model

EMCAL: S-shape scintillator in Pb-block
3 parts positioned in space

- no sub-volumes (parts)
- positioning ~lost
- material type ~lost
- fidelity has its price (simulation time)
- add gaps between parts to reduce risk of volume conflicts at lower fidelity
EMCAL: S-shape scintillator in Pb-block
3 parts positioned in space

Many approaches ...

in CAD
• export as STL single copies of U, B parts in the parts local ref frames

in Geant4
• re-assign materials to parts
• re-position parts in MARS (6 dof)
• clone part U2
Fidelity of tessellation

Low fidelity STL
- surface deviation $60 \, \mu m$
- 1k facets, file size 51 kB

Medium fidelity STL
- surface deviation $9 \, \mu m$
- 6k facets, file size 320 kB

High fidelity STL
- surface deviation $1 \, \mu m$
- 50k facets, file size 2.5 MB

CAD representation 'infinite' fidelity
Fidelity of tessellation vs. CPU cost

G4 simulation time for 400 positrons with selected energies (E)

CPU vs. # of facets

- E=100 GeV
- E=25 GeV
- E=5 GeV
- E=1 GeV

CPU time (seconds)

Higher fidelity ⇒ 4x more CPU

+60% CPU 2 x CPU

Fidelity of geometry representation

thrown 5 GeV positron in to Pb-Scint ECAL

5 x double: 40 B
disk size: 50 kB
disk size: 2.5 MB
Separation between parts

Low fidelity STL
*Fixed*: surface deviation 90 μm
*Varied*: separation gap between 2 parts

For rounded surfaces exported to STL with low fidelity and for narrow gaps between parts
- clearance between parts may shrink, be non-uniform
- parts may eventually collide
Geant4+STL: software configuration

Start: empty SL5.8 VM

Install base (compile locally):
  • CERN root, ver 5.34.04
  • Geant 4.9.6.p01

Add components needed for STL
  • ASSIMP for reading various CAD file formats
    • http://sourceforge.net/projects/assimp/files/assimp-3.0
    • minor code fixes
  • CADMesh parsing STL to Geant4
    • http://cadmesh.googlecode.com/files/cadmesh-v0.9.tar.bz2
  • Test STL (example provided)
    • ./build/cadmesh_example
CAD+Geant4 on one laptop

Virtual Machine #1: Linux/Geant4

Virtual Machine #2: Windows/AutoCAD

STL's in shared directory
Practical example: design of DarkLight detector

Hybrid approach

- re-invented in Geant4 as TTube, etc
- STL ported CAD ⇒ Geant4

- Magnet yoke = STL
- 3 Pb cylinders = TTube
- GEM frame = STL
- Aft beam pipe = STL
- Support disk = TTube
Several detector models were designed in CAD and efficiency was simulated in Geant4.
• **Tessellation Standard (STL)** simplifies automatic translation of solids from CAD to Geant4 shortening time of design-to-simulation cycle

• **CADMesh** interface preserves shape of solids but position and material assignment needs to be redone in Geant4

• Hybrid approach combining STL definition for complex volumes with native Geant4 definitions for simple shapes works well

• **Some practice is needed to apply this scheme for larger detector systems**