Geometry Persistency

I. Hrivnacova, IPN Orsay

Geant4 ED PHENIICS Tutorial,
14 - 18 May 2018, Orsay

Outline

- Text files (ASCII)
- Geometry Description Markup Language (GDML)
- Exchanging geometries with ROOT
- Importing geometries from CAD

ASCII Text Models

ASCII File Format

- Well defined syntax for identifying the different geometrical entities:
    - materials, solids, volumes and volume attributes
- Dedicated manual:
    - https://geant4.cern.ch/collaboration/working_groups/geometry/docs/textgeom/textgeom.pdf (link)
- Example of use of ASCII text model:
    - extended/persistency/P03

ASCII File Format - Example

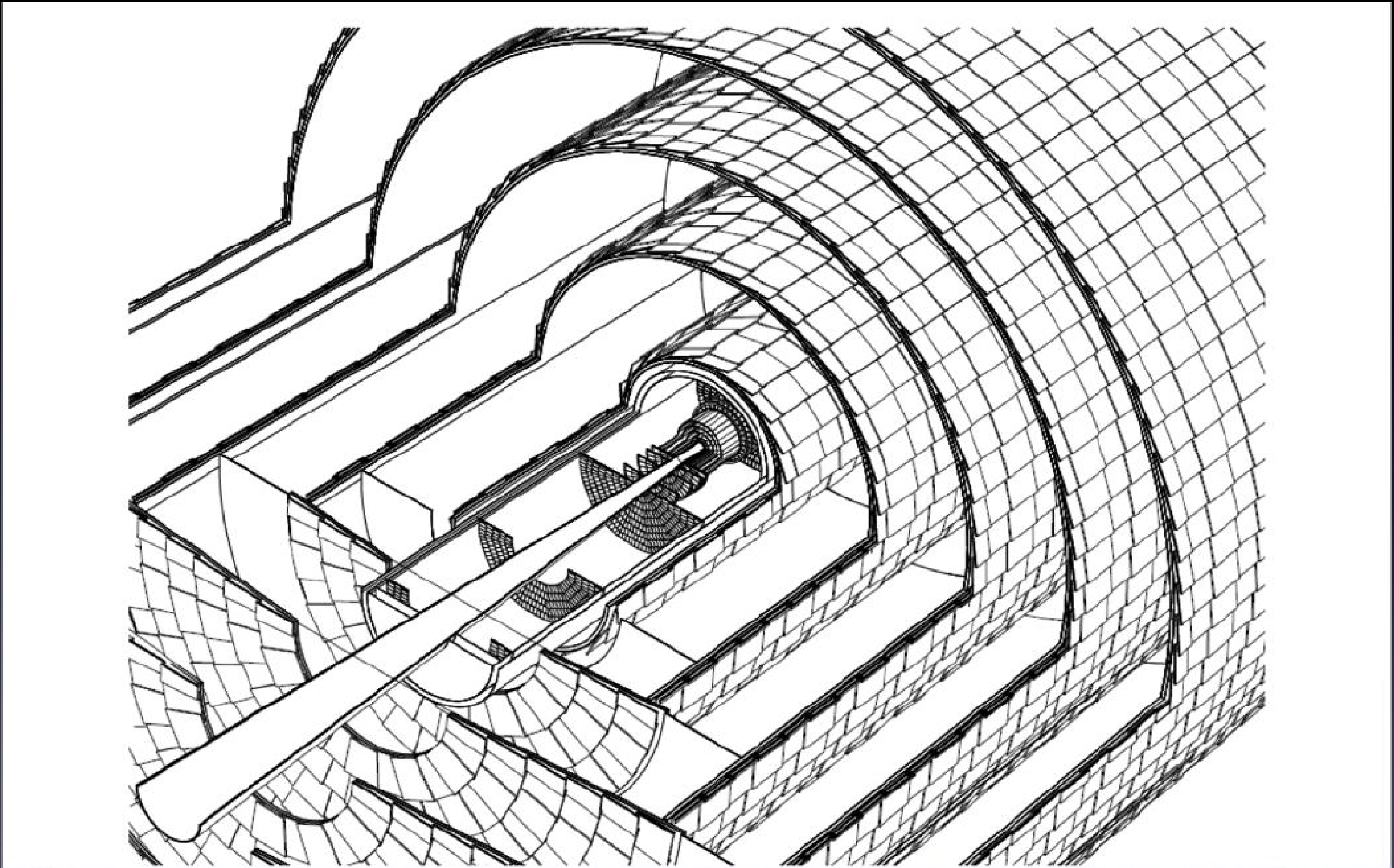
g4geom_simple.txt

```
// Define a parameter for later
// use
:P POSZ 5.

// Define materials
:ELEM Hydrogen H 1. 1.
:ELEM Oxygen O 8 16.
:ELEM Nitrogen N 7 14.
:MIXT Air 1.214E-03 2
      Nitrogen   0.75
      Oxygen     0.25

// Define rotation matrix
:ROTM R00 90. 0. 90. 90. 0. 0.
// unit matrix
```

```
// Define volumes and place them
:VOLU world BOX 30. 30. 30. Air


:VOLU "my tube" TUBS 0. 10. 20.
0. 360. G4_WATER
:PLACE "my tube" 1 world R00 0.
0. $POSZ


:VOLU sphere ORB 5.  G4_AIR
:PLACE sphere 1 "my tube" R00 0.
1. 10.
```

GDML

Defining Geometry in GDML



Silicon Pixel & Microstrip Tracker for Collider Detector
Norman Graf, LCDD Collaboration, SLAC

GDML Format - Example

```
<box name=”worldBox” x=”1”  y=”1”  z=”1”  unit=”m” />
<box name=”boxA”     x=”10” y=”10” z=”10” unit=”cm” />

<position name=”pos1” x=”25.0” y=”50.0” z=”75.0” unit=”cm” />
<rotation name=”rotZ” z=”30.0” unit=”deg” />

<volume name=”World”/>
   <material ref=”Air” />
   <solid ref=”WorldBox” />
   <physvol >
       <volumeref ref=”boxA”  />
       <positionref ref=”pos1”  />
       <rotationref ref=”rot1”  />
    </physvol >
</volume >
```

- Examples of use of GDML with Geant4:
  - extended/persistency/gdml

GDML Geometry

- An XML-based language designed as an application-independent persistent format for describing geometries of detectors
  - Allows to define hierarchy of volumes, their materials and solids
- As pure XML, GDML can be used universally
  - Not just for Geant4
  - Can be used for interchanging geometries among different applications, used also to translate CAD geometries to Geant4
- XML (Extensible Markup Language) = a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
- XML is simple
  - Rigid set of rules, self-describing data validated against schema
- XML is extensible
  - Easy to add custom features, data type

Geant4 <-> GDML

- G4GDMLParser class provides import/export of GDML files into/from Geant4
  - Import:

```
#include “G4GDMLParser.hh”

G4GDMLParser* parser;
parser.Read(”geometryFile.gdml”);
G4VPhysicalVolume* world = parser.GetWorldVolume();
```

  - Export

```
G4VPhysicalVolume* worldPV = ...;   // Get world
physical volume
G4GDMLParser* parser;
parser.Write(”geometryFile.gdml”, worldPV);
```

ROOT <-> GDML

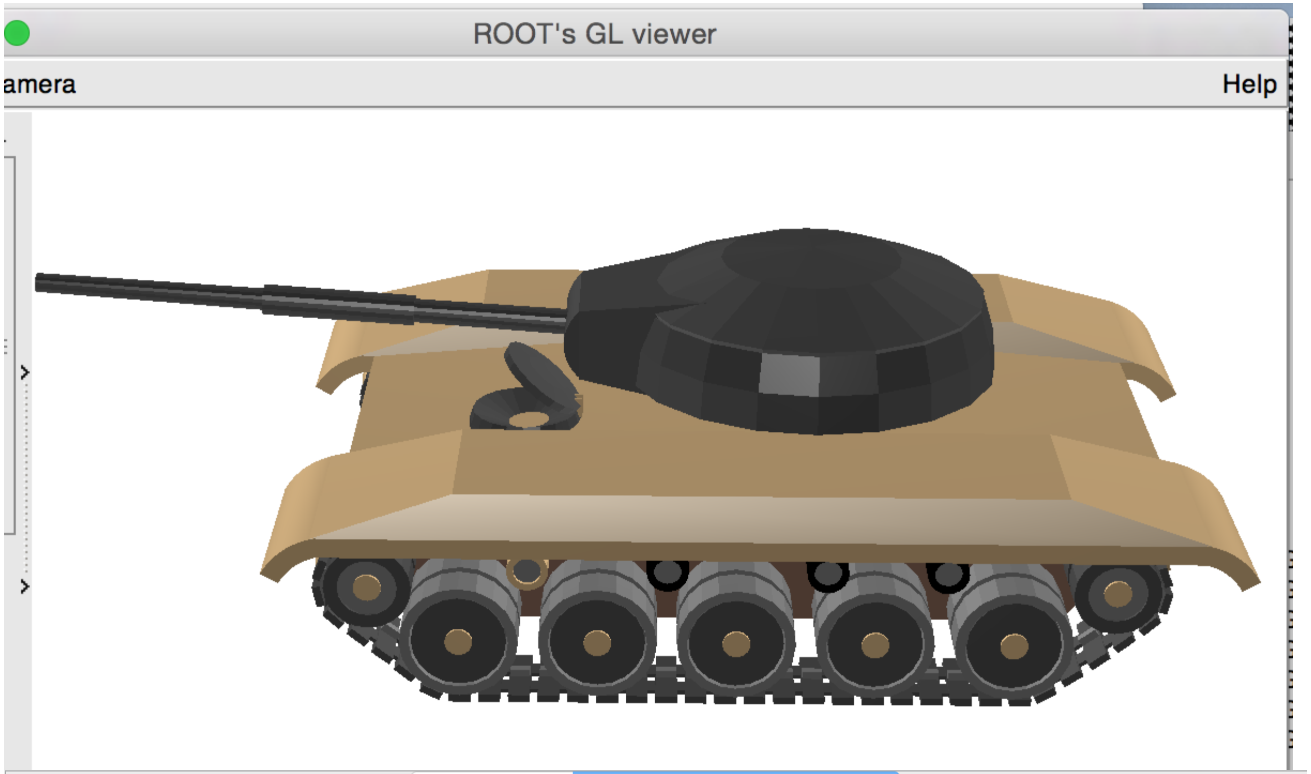
- ROOT geomety model can also import/export GDML
  - Import:

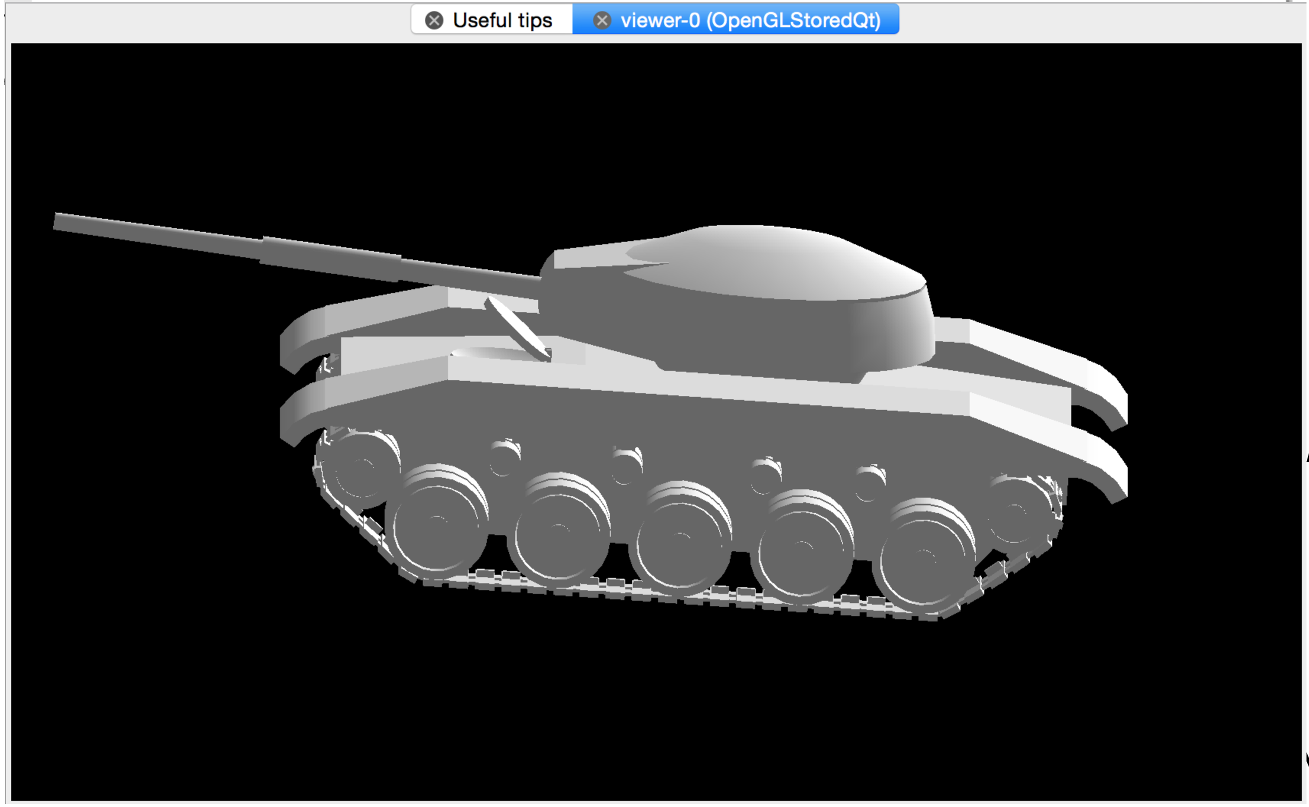
```
root[0] TGeoManager::Import(”geometryFile.gdml”);
```

  - Export

```
root[0] gGeoManager->Export(”geometryFile.gdml”);
```

- GDML can be used to exchange geometry between ROOT and Geant4
  - The converted geometry may be incomplete if the source geometry is using solids not supported in the destination geometry model
  - ROOT geometry allows overlapping volumes, **not alllowed** in Geant4





root/tutorials/geom/tank.C

↓

tank.gdml

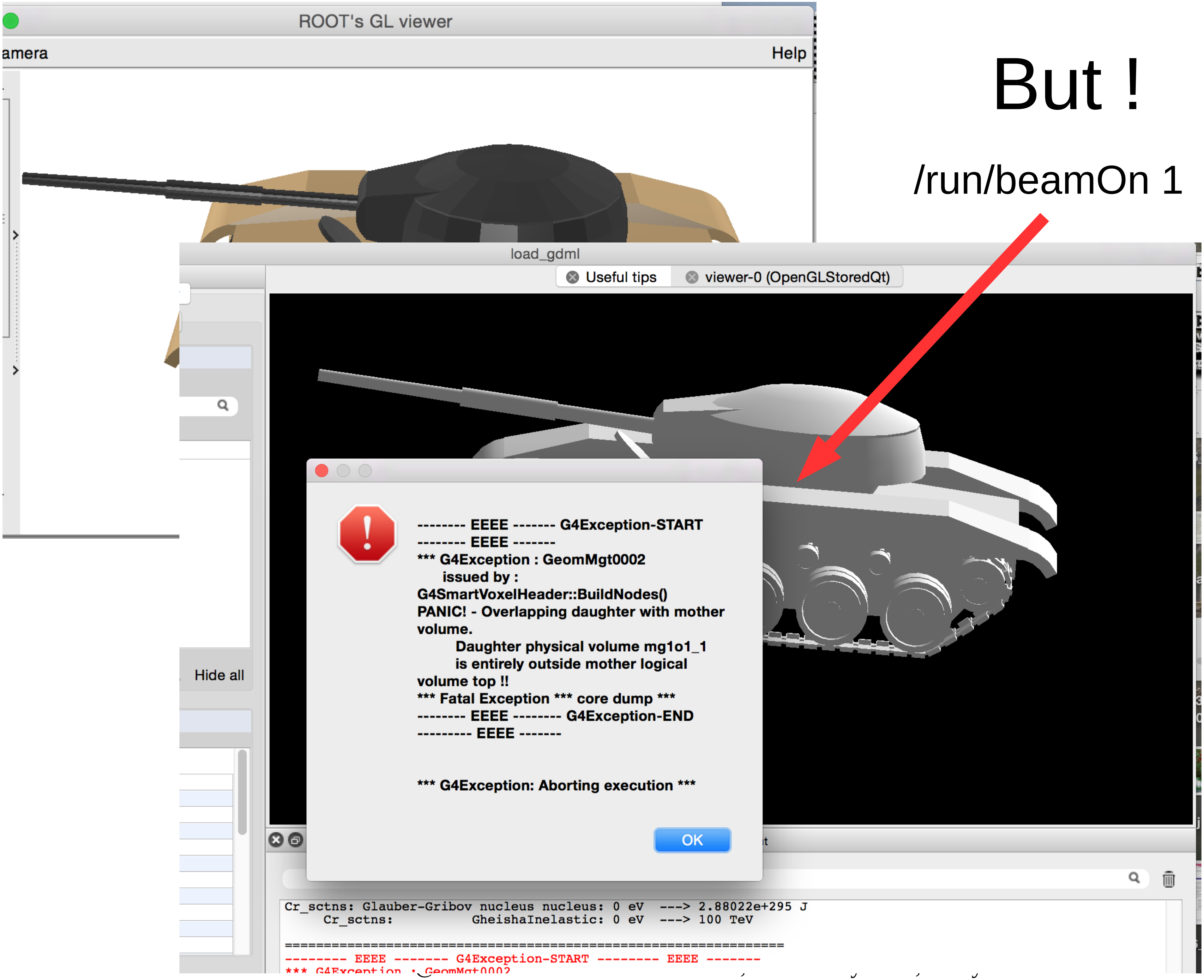
↓

geant4/examples/extended/persistency/gdml/G01

./load_gdml

But !

/run/beamOn 1

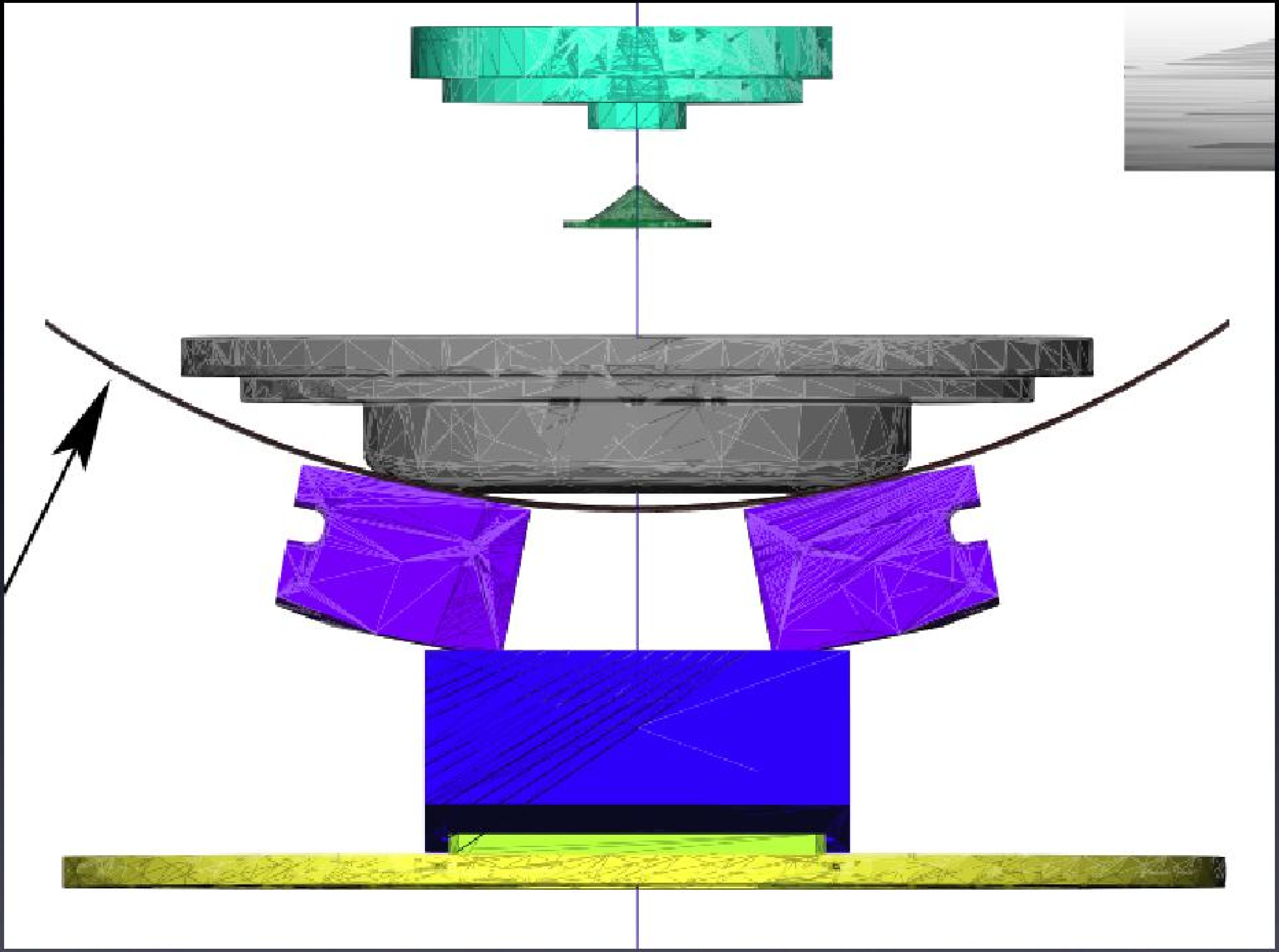


```
-------- EEEE ------- G4Exception-START
-------- EEEE -------
*** G4Exception : GeomMgt0002
      issued by :
G4SmartVoxelHeader::BuildNodes()
PANIC! - Overlapping daughter with mother
volume.
         Daughter physical volume mg1o1_1
         is entirely outside mother logical
volume top !!
*** Fatal Exception *** core dump ***
-------- EEEE -------- G4Exception-END
--------- EEEE -------


*** G4Exception: Aborting execution ***
```

```
Cr_sctns: Glauber-Gribov nucleus nucleus: 0 eV  ---> 2.88022e+295 J
     Cr_sctns:           GheishaInelastic: 0 eV  ---> 100 TeV

================================================================
-------- EEEE ------- G4Exception-START -------- EEEE -------
*** G4Exception : GeomMgt0002
```

CAD

Importing Geometry from CAD



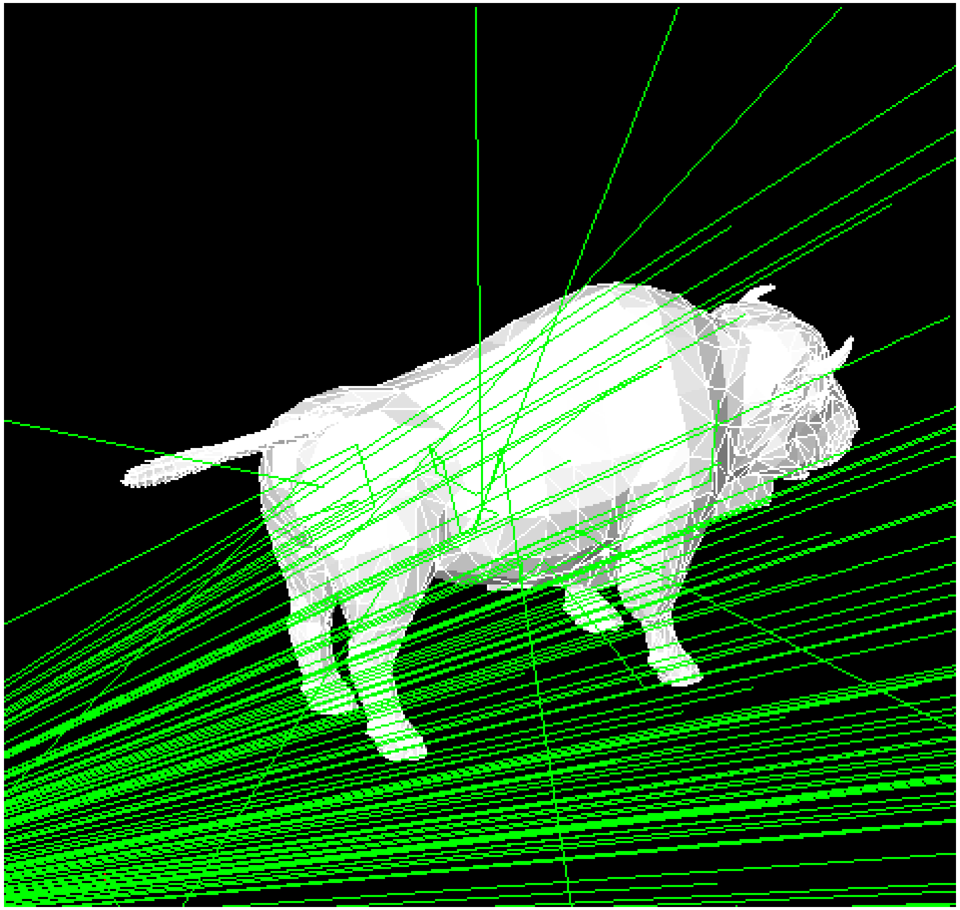
Varian TrueBeam Linac, M. Constantin et. al, Stanford University

Importing CAD Geometries

- CAD geometries: 3D engineering drawings
- No unique file format, but most CAD programs support STEP output format
- Current solutions of converting STEP to Geant4
  - **ST-Viewer**: http://www.steptools.com/
  - **FastRad**: http://www.fastrad.net/
  - Not free, not perfect, but still helpful

Importing CAD Geometries (2)

- **Cadmesh:** https://github.com/christopherpoole/cadmesh



- Alternative solution to go from CAD to STL (Stereo Litography) or PLY (Polygon File Format) to G4TessellatedSolid
- Use of cadmesh classes in Geant4 application for reading STL or PLY files (from CAD) and generating G4TessellatedSolid objects in memory

Backup

GDML Overview

- Definitions
- Materials
- Solids
- Structures
- Setup

GDML Definitions

- Allow to define numerical values of constants, positions, rotations and scales that can be used later on in the detector description
  - Use of CLHEP expressions
- Constants

```
<constant name=”length” value=”6.25”/>
```

- Variables

```
<variable name=”x” value=”6”/>
<variable name=”y” value=”x/2”/>
```

  - Once defined, can be used anywhere later, eg.

```
<box name=”my_box” x=”x” y=”y” z=”z”/>
```

GDML Definitions (2)

- Positions

```
<position name=”pos1” x=”25.0” y=”50.0” z=”75.0” unit=”cm” />
```

- Rotations

```
<rotation name=”rotZ” z=”30.0” unit=”deg” />
```

- Scales

```
<scale name=”reflectionZ” x=”1.0” y=”1.0” z=”-1.0” />
```

- Matrices

```
<matrix name=”m1” coldim=”3” values=” 0.4   9  126
                                       8.5   7   21
                                      34.6   7    9  />
```

GDML Materials

- Isotopes

```
 <isotope name=”U235” Z=”92” N=”235” >
    <atom type=”A” value=”235.04”  />
</isotope >
```

- Simple Elements

```
 <element name=”Oxygen” formula=”O” Z=”8” >
    <atom value=”16”  />
</element >
```

  - Compare to:

```
G4Isotope* U235
   = new G4Isotope(”U235”, 92, 235, 235.04*g/mole);
G4Element* O
   = new G4Element(”Oxygen”, “O”, 8.0, 16.0*g/mole);
```

GDML Materials (2)

- Elements with user-defined isotopic abundances

```
  <element name=”Enriched_uranium” >
     <fraction ref=”U235” n=”0.9”  />
     <fraction ref=”U238” n=”0.1”  />
  </element >
```

- Compare to:

```
 G4Element* enU
    = new G4Element(”enU”, “U”, 2)
 enU->AddIsotope(isoU235, 90.*perCent);;
 enU->AddIsotope(isoU238, 10.*perCent);;
```

GDML Materials (3)

- Material created directly from an element

```
  <material name=”Al” Z=”13.0”  >
     <D value=”2.70”  />
     <atom value=”26.98”  />
  </material >
```

- Material created from previously defined elements or materials by number of atoms (“molecule”):

```
  <material name=”Water” formula=”H2O”  >
     <D value=”1.0”  />
     <composite n=”2” ref=”Hydrogen”  />
     <composite n=”1” ref=”Oxygen”  />
  </material >
```

GDML Materials (4)

- Material created as a fractional mixture of previously defined elements or materials by number of atoms (“compound”):

```
<material name=”Air” formula=”air”  >
   <D value=”0.00129”  />
   <fraction n=”0.7” ref=”Nitrrogen”  />
   <fraction n=”0.3” ref=”Oxygen”  />
</material >
```

- Material via NIST ?

GDML Solids

- Collection of all Geant4 solid definitions
- Box, cone segment, ellipsoid, elliptical tube, elliptical cone, orb, paraboloid, parallelepiped, polycone. polyhedron, sphere, torus segment, trapezoid, general trapezoid, tube with hyperbolic profile, cut tube, tube segment, twisted box, twisted trapezoid, twisted general trapezoid, twisted tube segment, extruded solid, tessellated solid, tetrahedron
- Example of box:

```
<box name=”my_box” x=”x” y=”y” z=”z”/>
```

GDML Boolean Solids

- Supported boolean operations: *union, subtraction and intersection*

```
<box name=”box_A”  x=”1” y=”5” z=”20”/>
<box name=”box_B”  x=”4” y=”4.5” z=”18”/>
...
<union name=”union”  >
    <first ref=”box_A”  />
    <second ref=”box_B”  />
    <position ref=”union_position”  />
    <rotation ref=”union_rotation”  />
</union >
```

GDML Volumes

- Volumes are created from solids and materials that were previously defined in this or a linked GDML files
- Both logical and physical volumes are defined in one structure:

```
<volume name=”WorldBox ”/>
   <material ref=”Air” />
   <solid ref=”WorldBox” />
   <physvol >
      <volumeref ref=”box_B”  />
      <positionref ref=”union_position”  />
      <rotationref ref=”union_rotation”  />
   </physvol >
</volume >
```

- Replicated and parameterised volumes are also supported
  - See the documentation and examples

GDML Setup

- The top volume is defined in “setup” element:

```
<setup name=”TestI”  version=”1.0” />
   <world ref=”World” />
</setup >
```

- Multiple geometry setups can be defined choosing different volumes as world volumes
- Geometry description can be split over multiple files, allowing more granular and/or distributed development