Geant4 Version

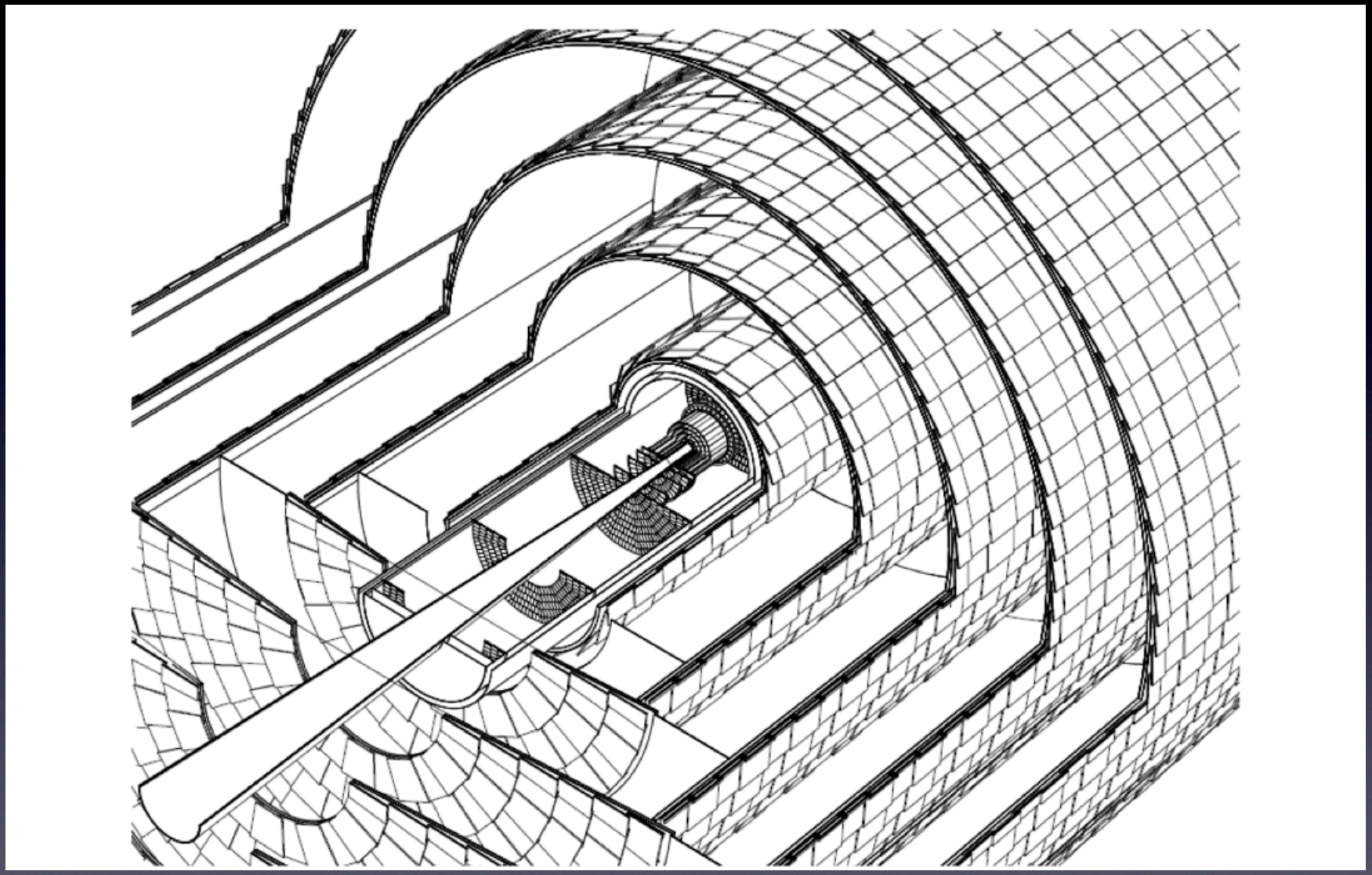
Geometry III: GDML and CAD

Joseph Perl
Geant4 Tutorial Course

Overview

- Up to now, the course has shown how to define materials and volumes from C++.
  - and messenger classes can allow some control over geometry from the Geant4 command line (e.g. example A01)
- This talk shows some alternate ways to define geometry at runtime by providing a file-based detector description

Defining Geometry in GDML



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

GDML: Geometry

- An XML-based language designed as an application-independent persistent format for describing the geometries of detectors.
  - Implements"geometry trees"which correspond to the hierarchy of volumes a detector geometry can be composed of
  - Allows materials to be defined and solids to be positioned
- Because it is pure XML, GDML can be used universally
  - Not just for Geant4
  - Can be format for interchanging geometries among different applications.
  - Can be used to translate CAD geometries to Geant4

Why XML?

- Simplicity
  - Rigid set of rules
  - Self-describing data validated against schema
- Extensibility
  - Easy to add custom features, data types
- Interoperability
  - OS, languages, applications
- Hierarchical structure
  - Appropriate for Object-Oriented programming
  - Detector/subdetector relationships
- Open W3 standard
- Many tools for validating, parsing, translating
- Automatic code-generation for data-binding

GDML Overview

- Definitions
- Materials
- Solids
- Structures
- Setup

GDML Definitions

- Contains numerical values of constants, positions, rotations and scales that will be used later on in the geometry construction.
  - Uses 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, e.g.
    - <box name="my_box" x="x" y="y" z="x+y"/>

GDML Definitions II

- Positions
  - <position name="P1" x="25.0" y="50.0" z="75.0" unit="cm"/>
- Rotations
  - <rotation name="RotateZ" z="30" unit="deg"/>
- • Scales
  - <scale name="my_reflection" x="-1" y="-1" z="1"/>
- • Matrices
  - <matrix name="m" coldim="3" values=" 0.4 9 126
    8.5 7 21
    34.6 7 9"/>

GDML Materials I

- Isotopes
  - <isotope name="U235" Z="92" N="235">
    - <atom type="A" value="235.04">
  - </isotope>
- Compare to:
  - G4Isotope* U235 =
    new G4Isotope("U235",92,235,235.04);

GDML Materials II

- Simple Elements
  - <element Z="8" formula="O" name="Oxygen" >
    - <atom value="16" />
  - </element>
- 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(“eU",“U",2);
  - enU->AddIsotope(isoU235,90.*perCent);
  - enU->AddIsotope(isoU238,10.*perCent);

GDML Materials III

- Materials can be defined in three ways:
- Created directly from an element:
  - <material name="Al" Z="13.0" >
    - <D value="2.70" />
    - <atom value="26.98" />
  - </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 IV

- Created as a fractional mixture of previously defined elements or materials, (“compound”):
  - <material formula="air" name="Air" >
    - <D value="0.00129" />
    - <fraction n="0.7" ref="Nitrogen" />
    - <fraction n="0.3" ref="Oxygen" />
  - </material>

GDML Solids

- Collection of all solid definitions which are used in the geometry description:
  - Box
  - Cone Segment
  - Ellipsoid
  - Elliptical Tube
  - Elliptical Cone
  - Orb
  - Paraboloid
  - Parallelepiped
  - Polycone
  - Polyhedron
  - Sphere
  - Torus Segment
  - Trapezoid (x&y vary along z)
  - 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

GDML Boolean Solids

- The boolean operations union, subtraction and intersection are also supported, e.g.
  - <box name="box_1" x="1" y="5" z="20" />
  - <box name="box_2" x="4" y="4.5" z="18" />

  - <union name="union" >
    - <first ref="box_1" />
    - <second ref="box_2"/>
    - <positionref ref="union_position" />
    - <rotationref ref="union_rotation" />
  - </union>

GDML Structure

- Volumes are created from solids and materials that were previously defined in this or a linked GDML file
- Both logical and physical volumes are defined in one structure:
  - <volume name="World">
    - <materialref ref="Air"/>
    - <solidref ref="WorldBox"/>
    - <physvol>
      - <volumeref ref="vol0"/>
      - <positionref ref="center"/>
      - <rotationref ref="identity"/>
    - </physvol>
  - </volume>

GDML Structure II

- Replicated and Parameterized volumes are also supported.
  - See the documentation for examples.

GDML Setup

- The top volume is defined here.
  - <setup name="Test1" version="1.0">
    - <world ref="World"/>
  - </setup>
- It is possible define multiple geometry setups choosing different volumes as world volumes
- Geometry description can be split over multiple files, allowing more granular and/or distributed development

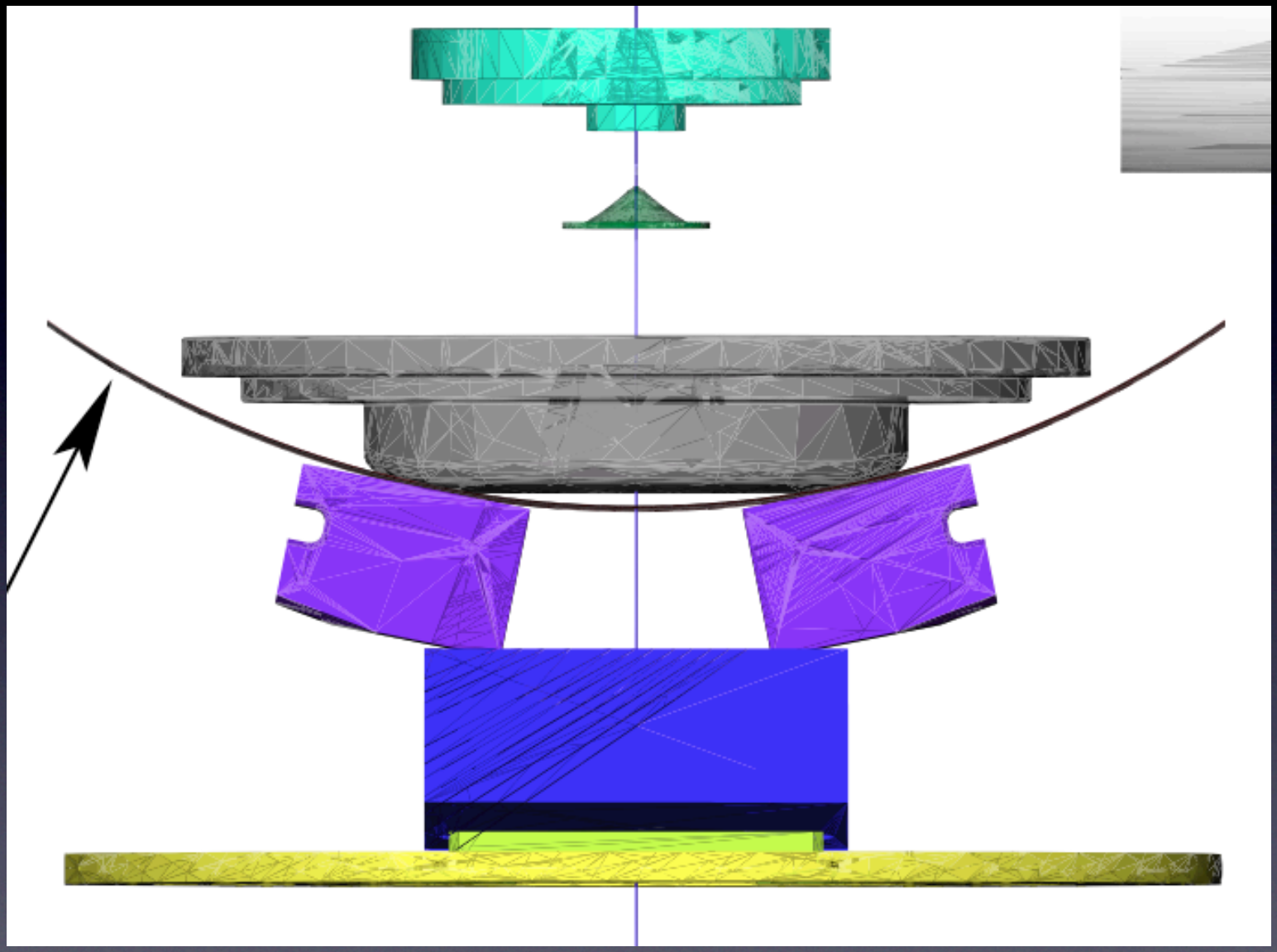
Importing into Geant4

- GDML files can be directly imported into Geant4 geometry, using the GDML plug-in facility:
  - #include “G4GDMLParser.hh”
- Generally you will want to put the following lines into your DetectorConstruction class:
  - In the Class Constructor:
    - G4GDMLParser parser;
  - In the Construct method:
    - parser.Read(“geometryFile.gdml");
  - To access the World volume:
    - G4VphysicalVolume* W=parser.GetWorldVolume();

Getting Started

- To include the Geant4 module for GDML:
  - Install the XercesC parser (version 2.8.0 or 3.0.0)
    - http://xerces.apache.org/xerces-c/download.cgi
  - Set appropriate environment variables set when G4 libraries are built:
    - Answer yes to relevant questions in ./Configure -build
    - Or if you don't use ./Configure -build:
      - set XERCESROOT to path to where the XercesC parser library and headers are installed
      - set G4LIB_BUILD_GDML to "1"
- Examples available in:
  - $G4INSTALL/examples/extended/persistency/gdml

Importing Geometry from CAD



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

Importing CAD Geometries

- Users with 3D engineering drawings may want to incorporate these into their Geant4 simulation as directly as possible
- Difficulties include:
  - Proprietary, undocumented or changing CAD formats
  - Usually no connection between geometry and materials
  - Mismatch in level of detail required to machine a part and that required to transport particles in that part
  - CAD is never as easy as you might think (if the geometry is complex enough to require CAD in the first place)
- One output format most CAD programs do support is STEP
  - Not a complete solution, in particular does not contain material information
  - There are movements under way to get new formats that contain additional information, but none yet widely adopted

Converting STEP to GDML

- Imperfect, but still helpful solutions are tools to convert STEP to GDML and provide the user a way to add materials information
- There are two cases where existing CAD programs have added GDML export features
  - Since these CAD programs can also read in STEP, they can be used as STEP to GDML converters
  - Neither option is free
  - Neither option works perfectly
- Discussion of these solutions takes place in the Geant4 Persistency forum:
  - http://hypernews.slac.stanford.edu/HyperNews/geant4/get/persistency.html?
- Useful technical note:
  - Linking computer-aided design (CAD) to Geant4-based Monte Carlo simulations for precise implementation of complex treatment head geometries., Constantin et. al., Phys Med Biol. 2010 Apr 21;55 (8):N211-20. Epub 2010 Mar 26

Current Solutions STEP to GDML

- ST-Viewer
  - A similar solution that was, until recently, free
  - No longer free, now part of ST Developer
  - http://www.steptools.com
- FastRad
  - GDML export extension was funded by European Space Agency Geant4 team
  - Not free (except for limited, trial mode that can handle only a small number of volumes)
  - Not perfect (we have seen rare but significant errors in conversion of some solids)
  - Still has produced some very cool results
  - http://www.fastrad.net/
- There is an effort under way to raise funding to pay to get this job done right: a more accurate tool, easier to use, with a free license
  - Just at the fundraising point right now

Summary

- Geometry definition using text files:
  - opens up the user-base to non-C++ coders
  - allows geometry setup to happen at run time rather than compile time
  - provides an way to take geometry data from CAD or to other databases (construction, survey,...)
    - but the path from CAD to GDML is not completely automatic
    - third party tools are required (free only for trivial use)
    - user must input materials information by hand
- GDML plugin fully supports the whole set of G4 solids, including boolean solids, parameterizations, and replications.
- GDML also allows a way to export Geant4-based geometry for downstream clients (e.g. event reconstruction, display)