Geometry: New Developments & Capabilities

Norman Graf (SLAC)

Geant4 Space Users' Workshop

August 19, 2010

Solids

- G4Sphere:
  - Implemented speed improvements and corrections from joint code review. Cached computation for half-tolerances and use of Boolean flag for identifying if full-sphere, shell or section.
  - Implemented caching of trigonometric values, now directly computed inside modifiers for Phi and Theta angles as required for parameterized cases.
  - Rationalized usage of relative radial tolerances.
  - Correction in DistanceToOut(p,v) for phi sections for rays passing through zero.
  - Fix for the calculation of the normal in DistanceToOut() to avoid cases of division by zero in specific configurations. Addresses problem report #977.

Solids

- G4Tubs, G4Cons:
  - Rationalized usage of modifiers for Phi angles and simplified constructors.
- G4Cons:
  - fix to DistanceToIn(p,v), added a check on the direction in case of point on surface. Fixes problem of stuck tracks observed in CMS.
- G4Torus:
  - fix in SolveNumericJT() in order to take into account the difference in the value of theta for different intervals, [0:pi] or [-pi:0], and for SPhi in [0:twopi] or [-twopi:0]. Addresses problem report #1086.
- G4Ellipsoid:
  - refined fix in DistanceToIn(p,v) for points located on curved surface, and correct treatment of geometrical tolerance. Addresses problem report #1076.

Divisions

- Implemented generic divisions along Z for polyhedras and polycones
  - Divisions can happen along Z-axis with width and offset
    - fulfils the condition that division does not span over more than one segment.
- Fixed initialization of division in Phi for polyhedras to not take into account user defined width.

Fields

- New stepper G4NystromRK4
  - Offers better computing performance in integrating the trajectories of charged particles in a magnetic field.
  - Uses the standard Nystrom method and a novel analytical estimation of the integration error.
  - Greatly reduces the number of field calls per integration step (from 10 to 3).
  - Achieves comparable accuracy with G4ClassicalRK4 in test cases.
  - It can only be used for pure magnetic fields which are not time-dependent.

Fields

- New class G4CachedMagneticField
  - Caches the value of a magnetic field class, in order to reduce the number of calls to an expensive field calculation or interpolation method.
  - For positions within the chosen radius of the previous position where it evaluated the field, it returns the last value of the field is automatically.
  - It can only be used for pure magnetic fields which are not time-dependent.

Fields

- Added new virtual method CalculateRightHandSide() to G4MagIntegratorStepper for use in caching momentum (and field value) by G4NystromRK4
  - Default implementation in G4MagIntegratorStepper calls RightHandSide().
- New class G4EqEMFieldWithEDM
  - Calculates Right Hand Side of equations of motion in a combined electric and magnetic field, with spin tracking for both magnetic and electric dipole moment terms.
  - Courtesy of Kevin Lynch, Phys. Dept. at Boston Univ.

Fields

- G4Region extension to hold local magnetic fields
  - Field manager can now be assigned to a region, and it will be used for all logical volumes it contains, except for those which override it by having a field manager of their own.

Additional Features in Release 9.3

- Fix in CheckOverlaps() for parameterized volumes
- Fixed G4LogicalVolume::TotalVolumeEntities() to become invariant to call sequence.
- Improved handling of small steps at boundaries due to geometry imprecision, navigation optimization, or the details of the algorithm used for tracking in field
- Improved memory management for navigation touchables
  - The new ad-hoc memory allocation system for touchables in navigation has shown an average CPU speed improvement of ~5% and a reduction of overall memory allocation and fragmentation for reasonably complex detector setups.

GDML Persistency in Release 9.3

- Implemented virtual layer to allow customization of the reader/writer for user-extended schemas.
- New ability to write optical surface properties associated to volumes and material properties
- New ability to handle ‘assembly’ structures and ‘expression’ tag in reader
- Corrected handling of ‘quantity’ tag
- Fix for dumping material property vectors
- General code cleanup

Planned Features (1)

- Implementation of precise ComputeSafety() in navigation for EM use - (1)
- Review of navigation verbosity & control at step number - (1)
- Finalize interoperability of multiple navigators/geometries - (1)/(2)

Planned Features (2)

- Extension of regular navigation to parameterization with cylinders - (2)
- New arbitrary trapezoid shape with vertices on parallel planes perpendicular to the Z axis [ALICE request] - (2)
- Extension to divisions to allow for gaps in replicated daughters - (2)
- Review classes exposed to kernel for thread-safety - (2)

Geant4 9.4 Beta

- Release 9.4 BETA is available from:
  http://geant4.cern.ch/support/download_beta.shtml

- Release Notes at:
  http://geant4.cern.ch/support/Beta4.9.4-1.txt

Defining a Detector

- Geant4 provides all the tools to define ~arbitrarily complex geometries.
  - Requires programmers experienced in both C++ & G4.
- Completely defining the detector geometry at runtime by providing a file-based detector description would be useful to a large number of end users.
- Geant4 provides two persistency examples:
  - ASCII: examples/extended/persistency/P03
  - GDML: examples/extended/persistency/gdml/

GDML

- Geometry Description Markup Language
- An XML-based language designed as an application-independent persistent format for describing the geometries of detectors.
- It implements ”geometry trees” which correspond to the hierarchy of volumes a detector geometry can be composed of, allows individual solids to be positioned, as well as to describe the materials they are made of.
- Being pure XML, GDML can be universally used, and in particular it can be considered as the format for interchanging geometries among different applications.
- http://gdml.web.cern.ch/GDML/

Why xml?

- Simplicity
  - Rigid set of rules
  - Self-describing data validated against schema
- Extensibility
  - easily add custom features, data types
- Interoperability
  - OS, languages, applications
- Hierarchical structure ↔ OOP, detector/subdetector
- Open W3 standard, lingua franca for B2B
- Many tools for validating, parsing, translating
- Automatic code-generation for data-binding
- Plain text: easily edited, cvs versioning

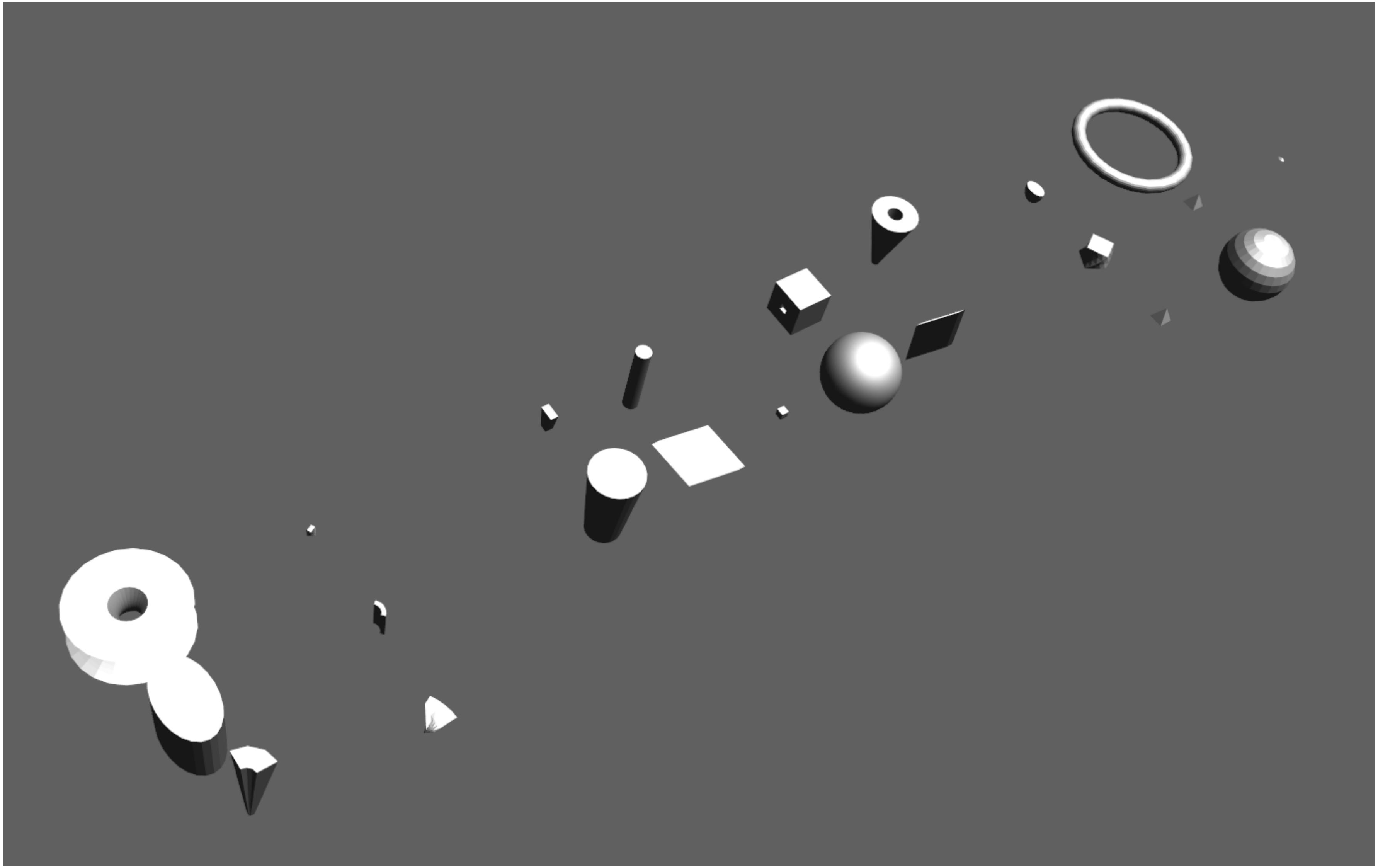
GDML Solids

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
Tesselated Solid
Tetrahedron

- Plus Boolean Solids (union, subtraction and intersection)

Solids.gdml



GDML & 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();
```

GDML Extensions

- GDML only provides a description of the detector geometry (volumes, materials and their hierarchical and geometrical positioning).
- Much more is required for most applications to fully describe the system.
  - Fields, regions, limits, sensitive detectors, etc.
- Example G03 provides a skeleton for extending gdml (uses visualization as an example).
  - examples/extended/persistency/gdml/G03

CAD

- Clients often have 3D engineering drawings of their setup and would like to simply incorporate those into their Geant4 simulation.
- Difficulties include:
  - Proprietary, undocumented or changing formats
  - Often no connection to materials
  - Mismatch in level of detail required to machine a part and to simulate the response of the part to particles.
- Exchange formats
  - Some standard CAD output formats exist, e.g. STEP & IGES, but these are surface-oriented formats and do not contain material information.
  - GDML is one candidate for standard Geant input.

CAD to Geant4 Tools

- FastRad is a free application (but requires a license and has a limit on the part complexity) which can import STEP files, associate materials, and export GDML files. http://www.fastrad.net/
- ST-Viewer from STEP Tools (was free until recently, now part of ST Developer http://www.steptools.com) can import a variety of CAD formats and export a STEP file with an associated material file. Example G02 can import these files and build geometry.
- STL2GDML: (see talk by Francisco García) http://www.solveering.com/products/products_stl2gdml.html
- Neither an endorsement of, nor an advertisement for, these third-party software packages.

CAD to Geant4 Summary

- Runtime geometry definition using text files:
  - opens up the user-base to non-C++ coders / non-G4 experts
  - provides an avenue to connect to CAD geometries or to other databases (construction, survey,…) as geometry input.
    - No cad2g4 command (yet), but some solutions exist.
  - provides an avenue to export the Geant4-based geometry for downstream clients (e.g. event reconstruction, display).
- GDML plugin fully supports the whole set of G4 solids, including boolean solids, parameterizations, and replications.
- Use of xml allows some control over GDML detector description (can be validated against schema), but also allows extensions to be added.

Summary

- Number of new geometry features are available in 9.3, with more to be released in 9.4.
- Glad to see adoption of GDML within this community.
- Looking forward to feedback during the Technical Forum and general discussion on Friday.