RECENT DEVELOPMENTS AND UPGRADES TO THE GEANT4 GEOMETRY MODELLER

J. Apostolakis, R. Chytracek, G. Cosmo, M. Dressel, V. Grichine, O. Link, W. Pokorski,
CERN Geneva, Switzerland
K. Hoshina, Chiba University, Japan
P. Arce, CIEMAT Madrid, Spain
D. Anninos, Cornell University, USA
G. Guerrieri, INFN Genova, Italy
I. Hrivnacova, IPN Orsay, France
M. Asai, SLAC Stanford, USA
M.H.Mendenhall, Vanderbilt University, USA

Abstract

The Geometry modeller is a key component of the Geant4 toolkit [1,2]. It has been designed to exploit at the best the features provided by the Geant4 simulation toolkit, allowing the description of the geometrical structure of complex detectors in a natural way, ranging from a few up to hundreds of thousands volumes of the LHC experiments, as well as human phantoms for medical applications or devices and spacecrafts for simulations in the space environment.

The established advanced techniques for optimizing tracking in the geometrical model have been recently enhanced and are currently under evolution to address additional use-cases. New geometrical shapes increased the rich set of primitives available, and new tools help users in the process of debugging the geometrical setup.

The major concepts of the Geant4 geometry modeller are reviewed, focussing on recent features introduced in the last releases of the Geant4 toolkit.

INTRODUCTION

Several design iterations and improvements were introduced to the Geant4 geometry modeller since its first implementation released in 1996 as part of the R&D *alpha* version of the source code. In the past three years there have been many extensions added, most of them originated from the requirements advanced in the user community and the needs in different application domains.

The Geant4 geometry modeller allows for the description of complex experimental setups of the size of the biggest LHC experiments, as well as complex regular structures like the snapshot or a whole description of a human phantom for use in medical physics. Geometries can spawn from the description of a planet's surface to a DNA cell model. Special optimization techniques are adopted and implemented in order to achieve efficient navigation during tracking and tune at the best the navigation structure according to the geometrical topology of the model under consideration.

Tools have been developed to help debugging ill or malformed geometrical setups, to help in handling misaligned geometries or setups that vary in time, to compute associated physical quantities (like the geometrical volume of a structure or its mass), and to apply special biasing techniques associating weights to geometrical elements.

The modeller has been recently extended by introducing the concept of a *region* to 'tag' areas of a detector model with attributes for tuning physics simulation. The possibility to group volumes as assemblies, or to automatically apply reflection to complete structures is available and integrated together with the rich variety of placement techniques already existing in Geant4.

The ability of importing/exporting persistent descriptions of the geometry is now possible thanks to the enhancements introduced to the GDML (Geometry Description Mark-up Language) schema, where now most of the shapes and positioning techniques of Geant4 are supported.

IMPROVED OPTIMISATION TECHNIQUE

In order to handle tracking with high efficiency in a wide variety of geometry setups and topologies, Geant4 has been developing since its early years techniques [3] to allow reducing the CPU time spent in computing intersections with the different geometrical elements (volumes) composing a setup. Calculating the intersection of a track with every positioned volume at each tree level would be extremely inefficient. The currently adopted technique (*smart voxels*) has been studied in order to optimise this process by lowering the number of candidate volumes to be tested for intersection. The

technique is derived from the voxels-based method also adopted in ray tracing, where space is subdivided into cubical volume elements (voxels) and a tree-based map is created by recursively dividing the detector into octants. A traditional voxel based technique retains the disadvantage of grid-based methods in that every voxel intersected along the particle's trajectory must be tested for intersection.

For each mother volume (the top-volume of a hierarchy of a tree or sub-tree of volumes), a one-dimensional virtual division is performed. The best axis for the virtual division is chosen by using a heuristic where equal subdivisions are gathered into single bins. Each division containing too many volumes is then refined by applying a virtual division again, using a second Cartesian axis. If the resulting subdivisions contain too many volumes, a further refinement can be performed by dividing again along the third Cartesian axis. Voxels not containing any volume are merged with the neighbouring voxel in the 3D space. Such technique allows to greatly improving run-time performance for 'pure' tracking and does not require the need to tune detector description parameters [4].

The voxels are computed at initialisation time, and require small memory and computing resources. At tracking time searching is done in the hierarchy of virtual divisions. The technique has been recently enhanced to allow for efficient tracking also on geometries making use of parameterisations for volume placement and shape; the generation of 3D voxelisation can be enabled for the most complex parameterisations or in setups where more than one parameterisations are mixed together. Additional extensions have been introduced to reduce the time spent at initialisation time, when the voxels are computed. This is particularly helpful in the case of 'dynamic' geometries (i.e. setups varying a portion of the geometry or the whole structure during time or every simulation run) or geometries to be tuned for alignments. Partial regeneration of the voxelisation is now possible and can be easily triggered as an option.

GEOMETRICAL SHAPES

Geant4 provides the definitions of a wide variety of geometrical shapes (solids). Solids with simple shapes, like rectilinear boxes, trapezoids, spherical and cylindrical sections or shells, are available directly as solid objects, according to the Constructed Solid Geometry (CSG) specifications. Functionalities provided by each of these solids have been recently reviewed to improve the accuracy in the response especially concerning the computation of normals on surfaces, edges and corners.

Other more specific solids are provided, like polyhedra, polycones, elliptical tubes and hyperbolic tubes. This set of solids was recently extended to include a new family of 'twisted' shapes (tube, box, regular and irregular trapezoids, see Fig. 1), an ellipsoid with cut in Z, a cone with elliptical base and cut in Z and a generic tetrahedron

(i.e. a shape which can be defined by 4 points in the 3D space).
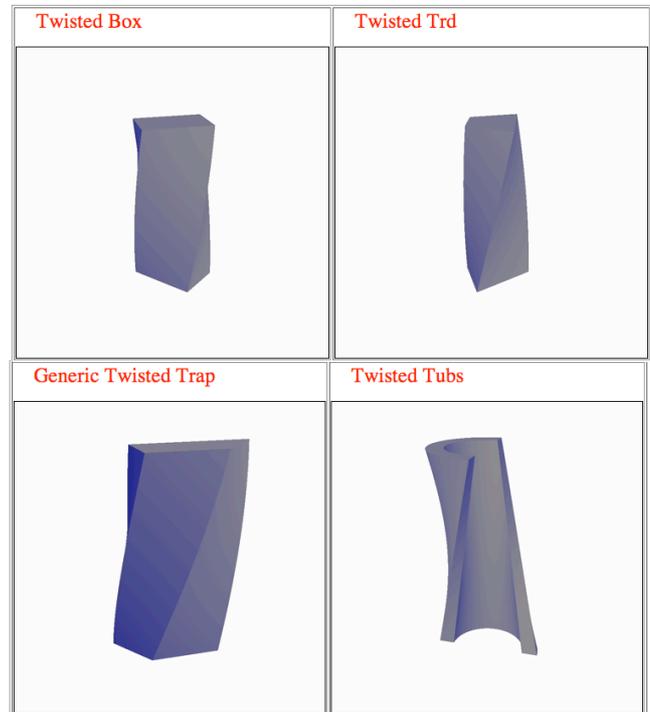
Figure 1: The new set of twisted solids.

More complex solids can be defined by their bounding surfaces, which can be planes or second order surfaces: *Boundary REPresentations* (*BREPs*) [5].

Other solids (*Boolean solids*) can be obtained by combining simple solids with Boolean operations, like unions, intersections and subtractions. The Boolean operation can also be performed by providing an optional transformation for one of the two solids involved. The solids used can be either two CSG solids (e.g., a box, a spherical shell, or a tube) or another Boolean solid (the product of a previous Boolean operation). With Boolean solids one can describe particular shapes in a simple and natural way with very efficient geometrical navigation inside them.

It was recently added the ability to compute for all shapes, either if a simple solid or a Boolean composition, the geometrical volume; for complex compositions or complex shapes, the value is estimated adopting a specific Monte Carlo technique by which an accuracy of the volume calculation on the level of 0.1% error is achieved. According to the accuracy of the detector description (level of detail, material composition, etc.), it is also possible to compute the *mass* of a given setup.

PLACEMENT OF GEOMETRICAL ELEMENTS

A detector's geometry is described by listing the different elements it contains and specifying their positions and orientations. A physical volume represents the spatial positioning of the volumes describing the detector elements, as positioned with respect to an enclosing (mother) volume. Structures of the detector that are repeated can be usually described as one volume placed several time in different places allowing for great memory saving in case of complex structures. Volumes can be replicated or divided according to a regular structure or can be parameterised according to a specific mathematical formula applied to their shape, positioning or attributes, such that only one instance of the physical volume will be finally created in memory.

It is also possible to define *nested* parameterised volumes, a feature which is particularly useful in the definition of voxelised regular structures for usage in medical physics applications.

Since release 3.1, it was introduced in Geant4 the ability to 'assemble' geometrical elements together in order to help the creation of irregular combined geometry structures to be easily placed multiple times. *Assemblies* are particularly useful to represent regular patterns of positioned volumes, for structures, which would be otherwise hard to describe with simple replicas or parameterised volumes. This is achieved by considering an assembly as a virtual *envelope* for the daughter volumes, which are part of the assembly; this envelope can then be 'imprinted' several times by providing the appropriate transformations.

A complete hierarchy of volumes can also be *reflected* according to a specified axis plane, therefore replicating the whole set of volumes with 'reflected' transformations. This feature is possible thanks to a specialised 'factory class' which takes care of determining the kind of placement technique adopted (replication, division, or simple placement), applying the appropriate transformations, positioning in the reflected counterpart and bookkeeping all new volumes being created. The factory also takes into account the replication of the attributes (material, electromagnetic field, visualization attributes, etc.) to be assigned in the reflected components. Query facilities for retrieving and/or identifying reflected volumes are also provided through the factory.

DEBUGGING TOOLS

Several built-in tools are provided in Geant4 for helping a user in modelling a detector's geometry. A frequent mistake a user can do while designing his/her own geometry, is the creation of overlaps while placing the different geometrical elements. The built-in tools in Geant4 help in identifying most overlaps in particular for those volume overlaps by which Geant4 tracking algorithms are quite sensitive and in general not tolerant. Volumes are defined as 'overlapping' when they actually protrude from their mother volume, or when volumes inside a common mother volume actually intersect each other.

Several techniques are implemented in Geant4 for detecting these erroneous setups, either by making use of the built-in tracking capabilities, or by querying directly the solids and the functionalities they provide, or by adopting visualization techniques (DAVID tool) [6].

A powerful technique recently introduced allows for determining overlaps at the time of creation of a geometrical setup, by optionally activating a check when placing a single volume in the setup. The placement of the volume is checked against the already existing volumes placed in the structure, by means of the generation of random points on the surface of the geometrical element being placed and verifying that none of the points are inside any other placed volume.

GEOMETRICAL REGIONS

In complex geometry setups, such as those found in large detectors in particle physics experiments, it is useful to think of specific structures of volumes as representing parts of the detector setup (sub-detectors), which perform specific functions. In such setups, the processing speed of a real simulation can be tuned and increased by assigning specific *production cuts* to each of these detector parts. This allows for a more detailed simulation to occur only in those regions where it is required. *Regions* can be assigned to volume hierarchies (logical volumes) in Geant4 and can be associated to specific production cuts or *cuts in range* bound to particles. The propagation of the region's attributes is triggered at initialisation time and happens recursively through the whole volume hierarchy until completion or if a new 'root volume' (i.e. a volume acting as top of a new region hierarchy) is found.

Different volume hierarchies can share regions. In a detector setup, for example, barrel and end-caps can be associated to two different regions respectively.

Memory management for regions happens as for volumes and solids through a registration / de-registration mechanism, which acts also as garbage collector at the end of the application.

Regions were recently extended to support also attributes related to user information classes, user limits and specifications for fast simulation.

BIASING TECHNIQUES

The Geant4 geometry modeller also provides the way to apply event-biasing techniques, which can be associated to the geometrical description of a detector [7].
These are particularly useful for saving computing time in applications like for example the simulation of radiation shielding. The biasing techniques implemented so far are: geometrical splitting and Russian roulette (also called geometrical *importance sampling*), and *weight roulette* [8].

'Weights' can be assigned to volumes in either the tracking geometry, or to volumes defining a *parallel* geometry over imposed to the real one, and where also scoring techniques can be applied.

Future developments foresee the definition of a coherent scheme for allowing parallel navigation on geometrical structures defined for usage in importance biasing, fast simulation and scoring.

PERSISTENCY WITH GDML

A geometry setup can be defined in several ways in Geant4: either by directly coding the various geometry components in pure C++ classes, making use of the API defined in the Geant4 toolkit; either by utilising the visual Java tool GAG [9] and generate through it the related C++ code to be integrated in the final application; either by importing/exporting the geometry model with GDML (Geometry Description Mark-up Language) [10].

GDML is a mark-up language based on XML and is aimed to define a common approach for sharing and exchange of geometry description data.
GDML makes possible geometry interchange between different applications, to allow for example comparisons of different Monte Carlo engines, using the features provided by their transportation algorithms and physics at the best. It also allows to import/export geometry descriptions for usage in different visualization systems, since thanks to XML, GDML defines an application independent format.

With GDML it is possible in Geant4 to store persistently on an ASCII file all the information related to a geometry model: material definitions, solids and volumes compositions, including now also replications, divisions and size/positioning parameterisations.

CONCLUSIONS

The geometry modeller of Geant4 is capable of describing complex geometries made of a combination of a large variety of shapes, recently extended to support also a set of twisted solids.

The adopted optimisation technique allows to greatly reducing the CPU time spent in computing volume intersections; the algorithm was recently revised to reduce substantially the initialisation time spent in optimising dynamic geometries. Memory consumption can be greatly reduced thanks to the replication and parameterisation techniques available. Regular or irregular patterns can be easily replicated, assembled or reflected.

Powerful tools are provided for the detection of overlaps in the geometry setup; detection of overlaps is now possible also at construction time.

Biasing techniques and properties for tuning physics production at the best can be associated to the geometry, thanks to the concept of a region.

Geometry models can be easily exchanged by importing/exporting their GDML descriptions.

REFERENCES

[1] S.Agostinelli et al., Geant4 Collaboration, Nuclear Instruments and Methods in Physics Research, **A 506** (2003) 250. See also the Geant4 web page: http://cern.ch/geant4.
[2] J.Allison et al., Geant4 Collaboration, IEEE Trans. Nucl. Sci. 53, February 2006, in press.
[3] Pure Tracking and Geometry in Geant4, P.Kent, April 1995, CERN note.
[4] Minimising Precision Problems in Geant4 Geometry, P.Kent, April 1995, CERN note.
[5] GEREP, a Boundary Representation Modeller proposal for Geant4, J.Sulkimo and J.Vuoskoski, IT Division Internal Report, CERN.
[6] S.Tanaka and K.Hashimoto, Proceedings of the CHEP '98 Conference, Chicago, September 1998.
[7] M.Dressel, Geometrical importance sampling in Geant4: from design to verification, CERN-OPEN-2003-048, 2003.
[8] T.Booth, A Sample Problem for Variance Reduction in MCNP, Los Alamos National Laboratory Report LA-10363-MS, October 1985.
[9] M.Nagamatsu, T.Kodama, H.Uno, H.Yoshida, K.Ohtsubo, S.Tanaka, M.Asai, Proceedings of the CHEP '98 Conference, Chicago, September 1998.
[10] R.Chytracek, Proceedings of the CHEP '01 Conference, Beijing, September 2001.
See also the GDML web page: http://cern.ch/gdml.