Geometry Tools for Simulation

HOW2019, Jefferson Lab

B.Couturier CERN

Feedback for this talk

From ALICE, ATLAS, Belle2, CMS, LArSoft, LHCb…
Non exhaustive list, limited by time to prepare the talk

Special thanks to J. Chapman, G. Corti, G. Cosmo, A.Gheata, I.Hrivnacova, M.Kirby, E.Moyse, D.Muller, I.Osborne, W.Pokorski, M.Ritter, A.Salzburger, E.Sexton-Kennedy, S.Wenzel

Tried to regroup per point instead of presenting each experiment in turn

Misunderstandings are mine...

Plan

- Geometry description
- Misalignments and side effects
- Future tools and performance improvements
- CAD interfaces

Geometry descriptions

Obviously, geometry description is the basis for detector simulation

- Experiments have different ways to describe their geometries, in the persistent form on disk or in memory (mostly Constructive Solid Geometries)
    - ROOT TGeo (C++) representation: ALICE/FAIR
        - Loaded from GDML for LArSoft
    - Geant4 solids: Belle2
    - DD4hep (XML + C++ constructors, TGeo in memory)
        - FCC, STCF, ILD, Sid and CLICdp
    - Experiment designed frameworks:
        - CMS: DDCMS (XML, C++ constructors)
        - ATLAS: GeoModel (XML, C++ constructors)
        - LHCb: DetDesc (XML representation on disk, custom memory representation)

Simulation frameworks

- Limited number of simulation toolkits: Geant4, FLUKA, Geant3
  - + Fast simulation tools (c.f. later)
- 2 ways to perform the simulations
  - Get the simulation to use the navigation tools from the experiment:
    - E.g. Virtual Monte Carlo, as used by ALICE, FAIRRoot
    - Implemented as part of ROOT https://root.cern.ch/vmc And Geant4 (Geant4_VMC)
  - Convert own geometry to the simulation framework model
    - ATLAS: Geo2G4
    - CMS: DD2G4
    - LHCb: G4 converter in the Gauss application
    - DD4hep: DDG4...
    - Belle2: VGM: https://github.com/vmc-project/vgm

There should not be overlaps in the converted geometry...

How good should the model be?

- Ideally we would like a perfect representation of the reality
  - What if we could directly convert the engineering drawings ?
- This is however not feasible
  - CAD model conversion is an issue (c.f. later slide)
  - Even if we could, the model would be far too complex
    - Memory usage explosion
    - CPU time for simulation would be far too high
- We need a model “good enough” for the task at hand
  - That of course depends on the task at hand (physics simulation ? study of the detector response ? radiations studies ?)
  - Using complex solids (e.g. using a tessellated solid for the LHCb RF Foil) not always the solution
    - A hand crafted detector model is a big investment but can save a lot of CPU time in simulations (more than 2/3rd of grid use for LHCb)

Different representations

- Several descriptions are therefore needed depending on the cases
    - E.g. CMS have tracking *(reconstruction)* and simulation geometries *(built from the same description)*
    - Fast Simulation may require simplified detector
- Example solutions
    - Manual tuning of geometry for various purposes
        - LHCb: Simplified geometry, Delphes model etc
    - ATLAS Fatras https://iopscience.iop.org/article/10.1088/1742-6596/898/4/042016/meta
    - CMS Fastsim https://arxiv.org/abs/1701.03850

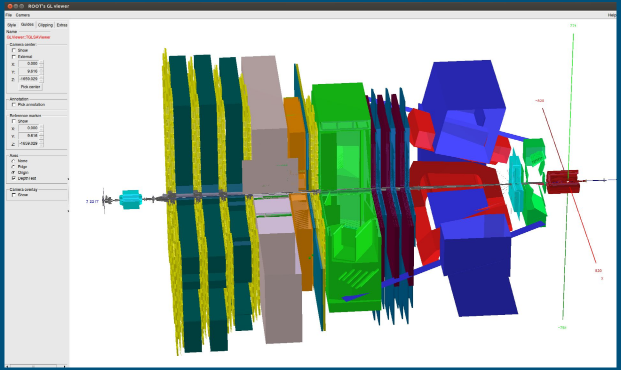
Representation model vs Implementation model

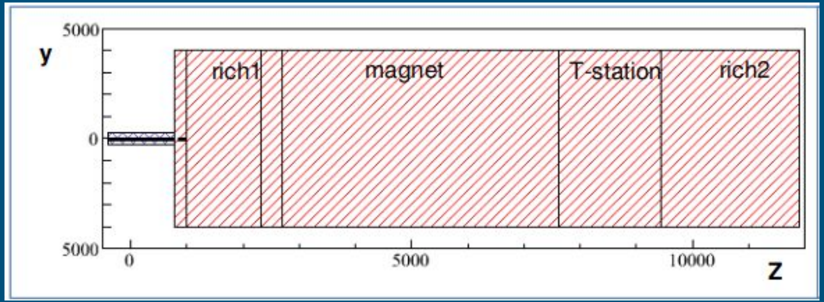
How to ensure consistency between all representations ?

Can we automatically simplify the geometry ?

The LHCb example

The simplified geometry doesn't necessarily match the envelopes of the shapes in the detector...





- C.f. simplified model in CMSSIM
- Can we have ways to keep track of alternative representations made by hand to make sure they are updated in sync ?

Issues with geometry conversion

- Geometry conversion implies converting simulation results back to original geometry
  - Need for mapping between the geometries
  - Native in e.g. TGeo
  - But adds complication, memory and CPU time
- Coupling sensitive volumes with geometry is not ideal
  - Need for some flexibility for various simulations
- Conversion to FLUKA geometry is not automated
  - And even cannot be as the descriptions are incompatible, except for simple volumes

Geometry evolution

- During the life of an experiment modifications have to be done to the detector description
  - Improved modelling
  - New survey
  - [...]
- Integrating geometry evolution tracking in the toolkits could be useful
  - Maybe goes beyond what the geometry modelly could/should offer ?
  - n.b. In some cases one have to keep C++ libraries and XML description in sync...

Unification of descriptions

- Do we need so many modelling toolkits ?
  - ATLAS, CMS, LHCb had to write its own converter to G4 model for example...
  - Makes it hard to work on common tools (verifications, visualization…)
    - One of the reasons for LHCb to move to DD4hep
- Problem already identified
- VecGeom/USolids proposes to unify the implementation of the geometrical primitives and navigation algorithms (c.f. http://aidasoft.web.cern.ch/USolids)
  - And to optimize the use of the geometry
  - International R&D effort

Non ideal geometries

- Needed as soon as the detector is built…
    - But also a little before to prepare the alignment procedure
    - And to check the impact on tracking etc
- Need to modify the model accordingly
- In the LHCb case:
    - Only coarse corrections are applied to the DetDesc description (Survey results)
        - Also need to adapt to the position of the VELO
    - G4 converter can take most alignments into account
        - Lengthy Detailed check for overlaps
        - does not work for all detectors (depending on how the geometry was implemented)

Misalignments and overlaps

- Overlap issue cited as a problem by most
  - The way you describe your geometry can make the problem worse (e.g. CMS thinking about redoing part of their geometry with assemblies)
  - ALICE mentioned this as one of the reasons for not using the G4 navigator (G4 navigator can deal with overlap within the defined geometrical tolerance)
  - ATLAS have the issue as well
  - Belle2 mentioned the overlap checking being a manual and painful operation as well
- The overlap tools work, but:
  - Results can be cryptic (conversions can mean rounding errors XML-> Memory->Geant4)
  - solving the problem is not trivial, depending on how the geometry is organized...

What can be done?

- This is a big issue for all experiments

  With no obvious/easy solution at the level of geometry tools

- Would it be useful to add extra info e.g. per volume “tolerance” to the geometry ?
  - Ensuring clearance between the various parts as is done in mechanical engineering
- Still a large investment in the tools and design
  - And not necessarily worth for a whole system
- Finding issues is reasonably easy. **Debugging and fixing is difficult**
  - An expert is needed to understand and fix the problems
  - Could extra metadata (e.g. to link the subparts to the mechanical design drawings) be useful ?

Geometry and multithreaded frameworks

- Computing landscape is changing, experiments are moving to multithreaded frameworks
    - CMSSW is multithreaded, ATLAS, LHCb following suit
      Working correctly in a multithreaded environment is a must
- Not all tools support natively multithreading:
    - TGeo creation cannot be done by multiple threads in parallel
      (Navigation doesn't suffer that problem)

Performance aspects

- Loading time is important (more for reconstruction jobs than for simulation)
  - Custom frameworks can be tuned finely (e.g. DDCMS faster than DD4hep for the moment)
  - LHCb looking at optimization in that domain (e.g. cache the DD4hep generated TGeo model)
- Improvements to the navigation
  - VecGeom made it its goal to have better primitives in which the navigation can be vectorized…
  - Provides a vectorized "voxelized" access to the geometry
  - One order of magnitude faster than Geant4 for complex faceted solids (https://indico.cern.ch/event/736011/contributions/3057840/attachments/1677553/2693817/draft_chep2018.pdf)
    - But not necessarily much gain on simple volumes (as reported by LHCb) (c.f. https://indico.cern.ch/event/658060/contributions/2918177/attachments/1624343/2586060/HSF20180328-LHCbSimForRun3-GCorti.pdf)
- Description design impacts navigation performance
  - Multiple ways to model the same detector, with very different performance
  - Toolkits do not help modellers do the "right thing"
    - Heuristic checks could help ?
    - Training ? tips and tricks ?

TGeo/Geant4/VecGeom Integration

- VecGeom brings promises of faster performance on complex volumes
  - Vectorized navigation
  - Improve performance on Polyhedra, Polycones, Extruded solids
- ALICE (S.Wenzel) looking into integration of VecGeom in their workflow
  - Transparently convert the geometry from TGeo to VecGeom
  - Let G4 navigator dispatch to VecGeom navigation interfaces
  - And makes simulation result useable in the TGeo
- CMS has integrated VecGeom in its Geant4-based simulation application
  - 9-13% CPU savings given the complexity of CMS' G4 shapes
  - **Validated for physics and in production since 2018**

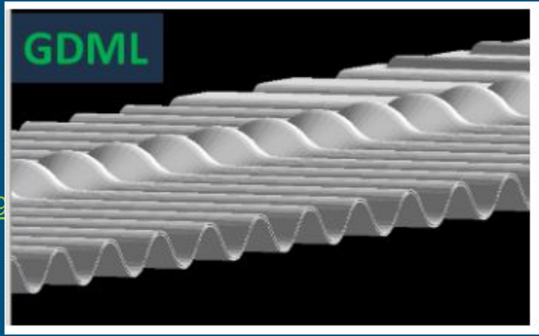
Visualization and debugging

- Many tools available
  - G4 tools, ROOT Visualization
  - That can be tricky. We need to focus on what G4 / FLUKA actually see
- Metadata ? Overlap check tools crucial to simulation
  - Are the tools good enough ? Do they give us enough debugging information ? Could heuristic based checkers help ?
  - Difficult to fix the geometry, depending on how it is specified (containing volumes or assemblies)
  - What are the good practices ? What are the bad one ? Tips and tricks ?

Interfacing with CAD Software

Importing from CAD

- Problem: B-Rep models have a semantic richer than CSG.
  Mesh can be problematic
  - STEP (STandard for the Exchange of Product model data)
    ISO 10303 STEP-2-ROOT (https://iopscience.iop.org/article/10.1088/1742-659
  - Step to GDML https://arxiv.org/abs/1105.0963 (CADMesh)
  - FastRAD - STEP to GDML (https://www.fastrad.net/publications/)
  - SolidWorks to GDML: https://arxiv.org/pdf/1702.04427.pdf
  - LHCb Blender based tools
  - [...]
- Generated Geometry may be too complex anyway and require manual modification
  - Complex mesh are too slow to simulate



LHCb RF Foil model

Exporting to other formats

- Exporting to other format
  - Tools are available e.g. TGeoCad (https://iopscience.iop.org/article/10.1088/1742-6596/523/1/012017/pdf) Exports to OpenCascade (In B-Rep)
    - Support discontinued on ROOT 6.16 due to lack of users
- Geometry may need to be simplified for:
  - Reconstruction
  - Visualization (e.g. outreach tools)
- Q: What are the needs regarding those tools?

Conclusions

- Many commonalities between experiments
  - But also many differences so the toolkit approach is appropriate
- Community efforts for new tools
  that are being adopted by experiments
  - LHCb moving to DD4hep
  - CMS investigating using it as a DD Mediator
    (i.e loading the description, XML and C++ algorithms remaining the same)
    Expected to complete by the end of 2019
- Designing, simplifying, misaligning geometries is not easy
  - (how) can the tools be improved to help ?
- HSF maybe can also help with good practices, advice etc...