Geant4 timing tests with GDML Workbench exports: Tessellations vs Booleans

In developing models for exporting non-GDML solids to gdml, we are faced with the choice of whether to model the non-GDML solids in terms of booleans of the intrinsic GDML solids or to simply export the solid as a tessellation. Presently, for the former, we have to develop the modeling ourselves – a time consuming process; for tessellations we can rely on the existing tessellation algorithms of FreeCAD. For tessellations we expect the tracking times in geant4 to be worse than for booleans of intrinsic shapes.

We present here some results that compare the performance for solids constructed from booleans vs tessellations. The bottom line is that tracking times for tessellations are up to a factor of three or more longer than for booleans of intrinsics.

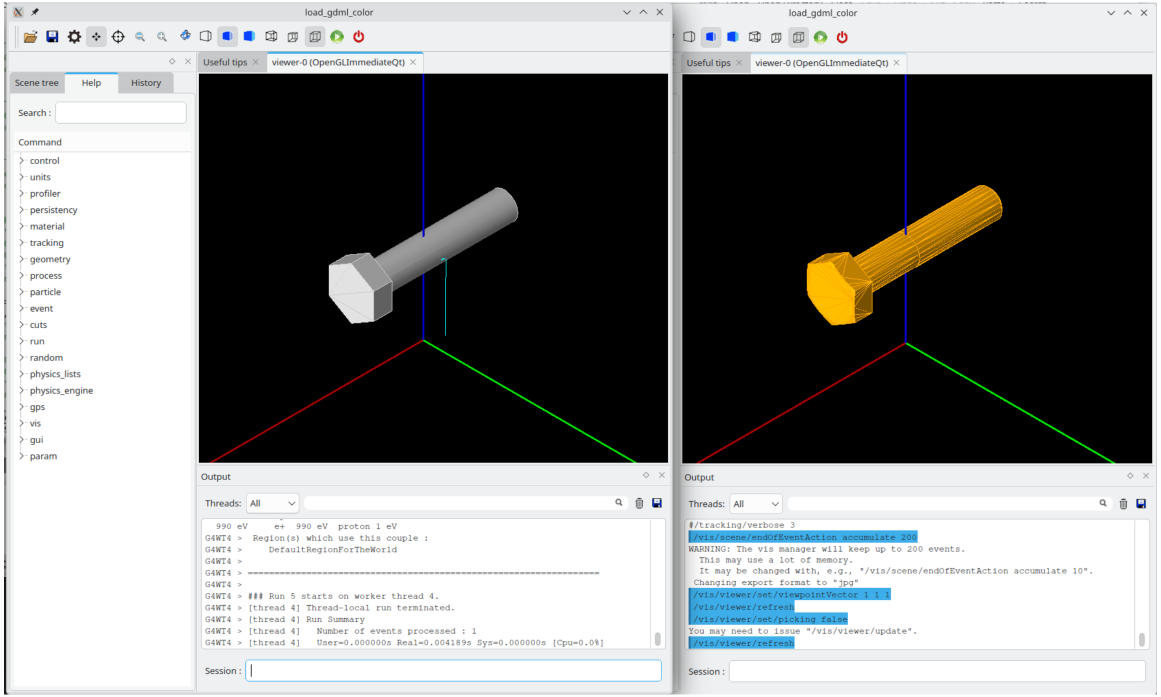
The tests were conducted on a Dell XPS 8930, with a
12 x Intel Core i7-8700 CPU @ 3.20 GHz

The operating system was (uname -a):

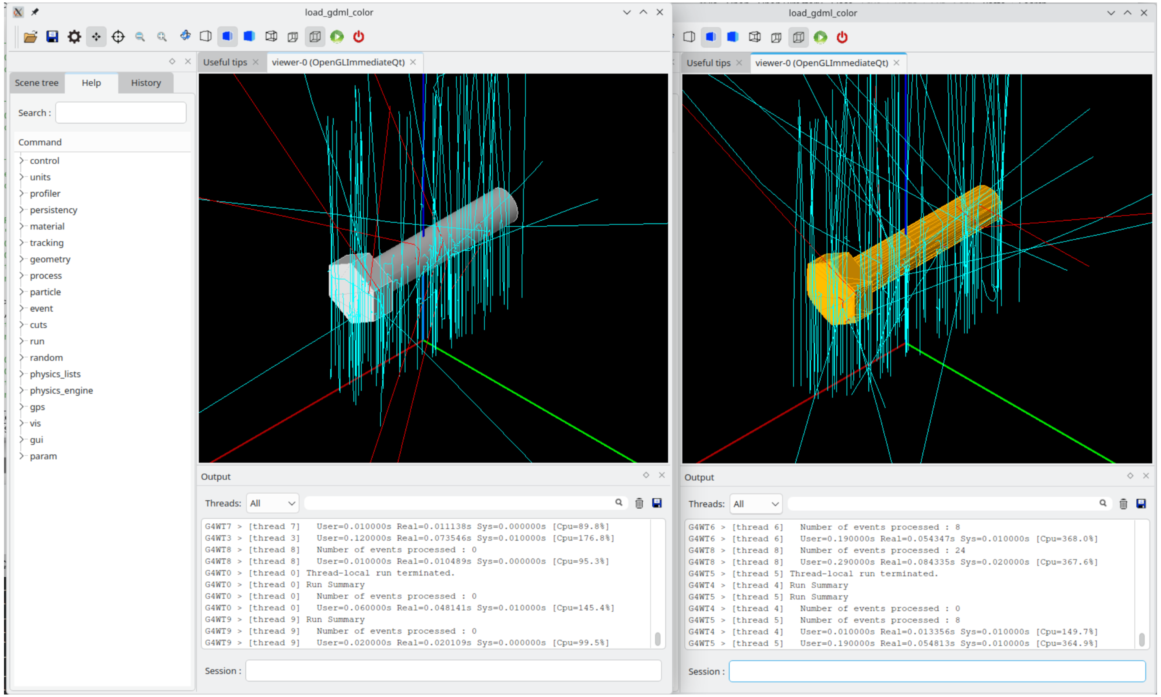
Linux dell 6.10.6-200.fc40.x86_64 #1 SMP PREEMPT_DYNAMIC Mon Aug 19 14:09:30 UTC 2024 x86_64 GNU/Linux

The version of geant4 was geant4-v11.2.1.

Bolt modeled as a boolean of an extrude and a tube, vs tessellated



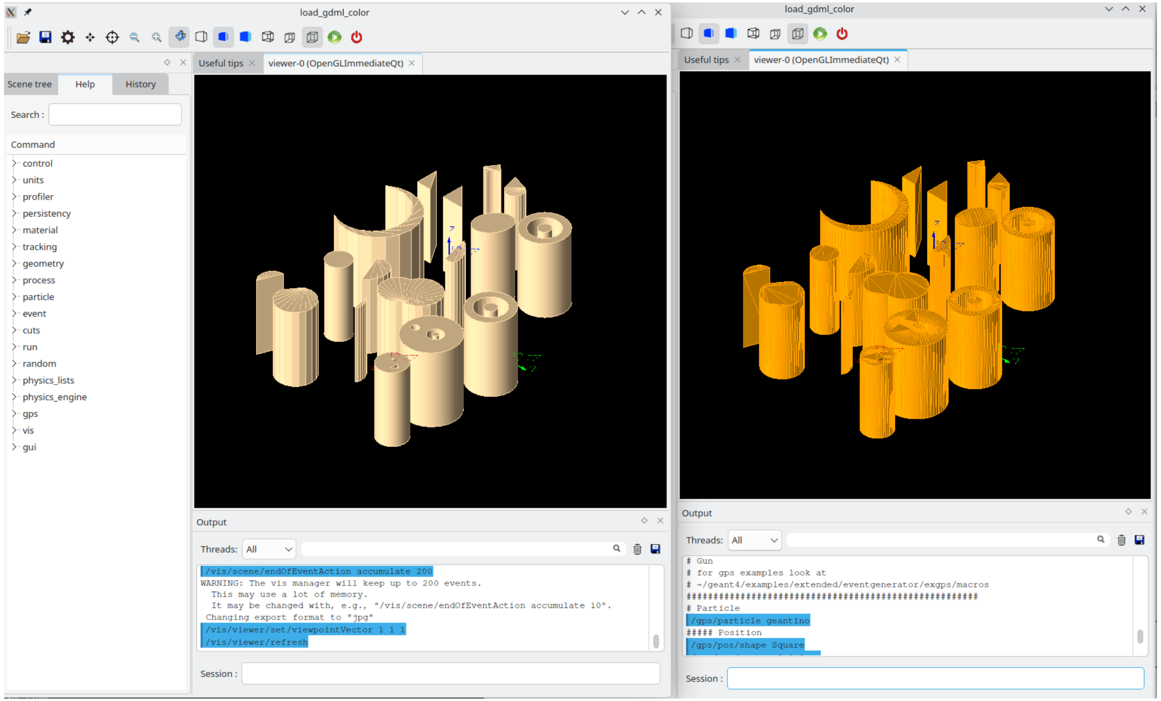
The solid on the right is a tessellation of an M6x30 unthreaded bolt;
the solid on the left a model of it using a union of a tube and an extruded hexagon.



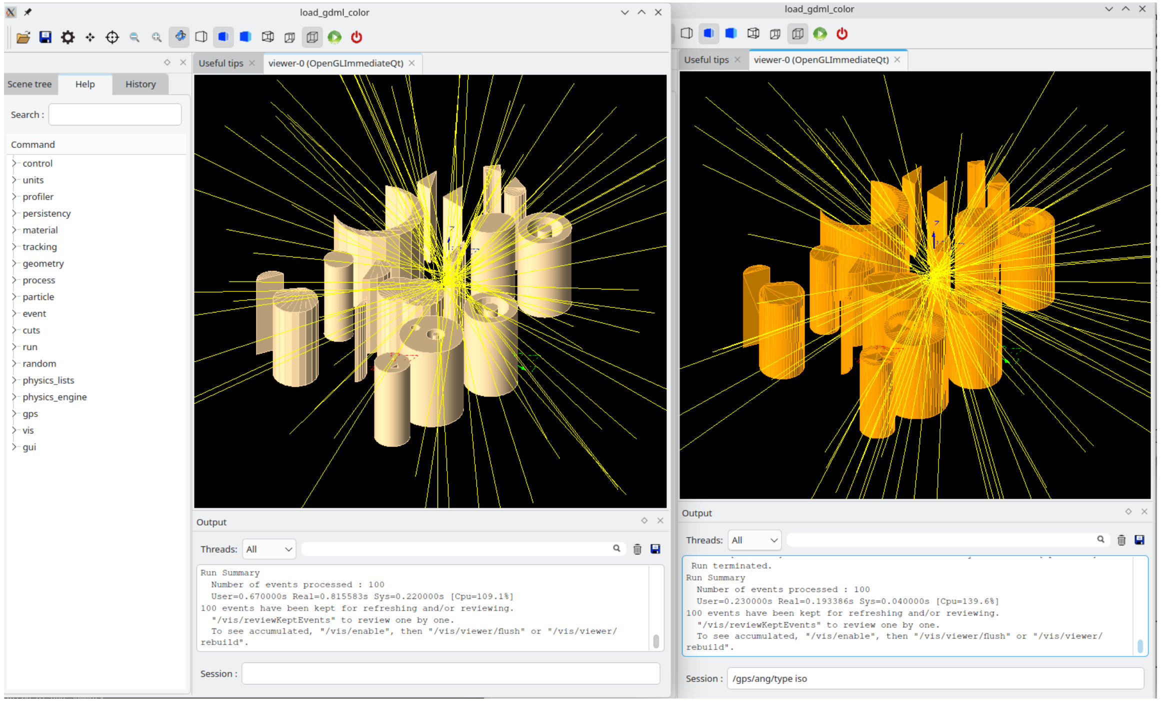
Material: G4_STAINLESS-STEEL		Beam: 2 MeV e-; 10^6 events	
	User time (s)	Real time (s)	CPU usage
Booleans	1346	115	1169%
Tessellation	4916	420	1170%

The tessellation is a factor 3.7 slower

Extrusions as booleans of intrinsic shapes vs a tessellation

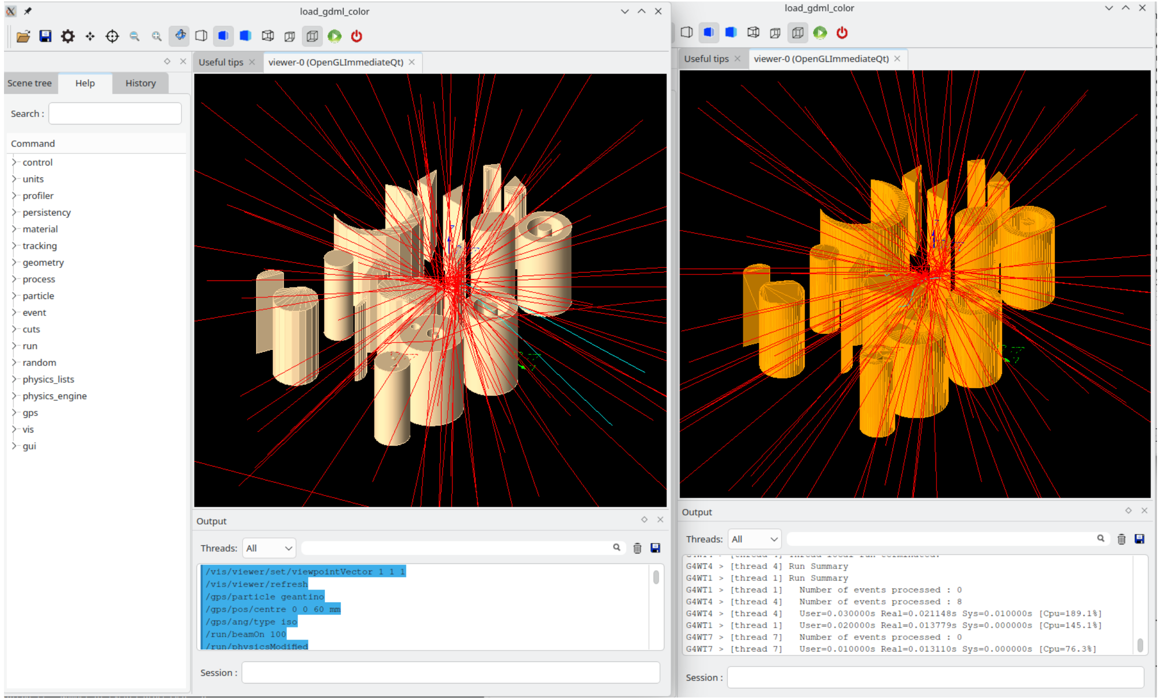


Extrusion on the left exported as booleans of intrinsic shapes; extrusion on the right exported as a tessellation.



Material: G4_WATER		Beam: Geantino; 10^6 events	
	User time (s)	Real time (s)	CPU usage
Booleans	81.9	7.2	1136%
Tessellation	95.2	8.4	1142%

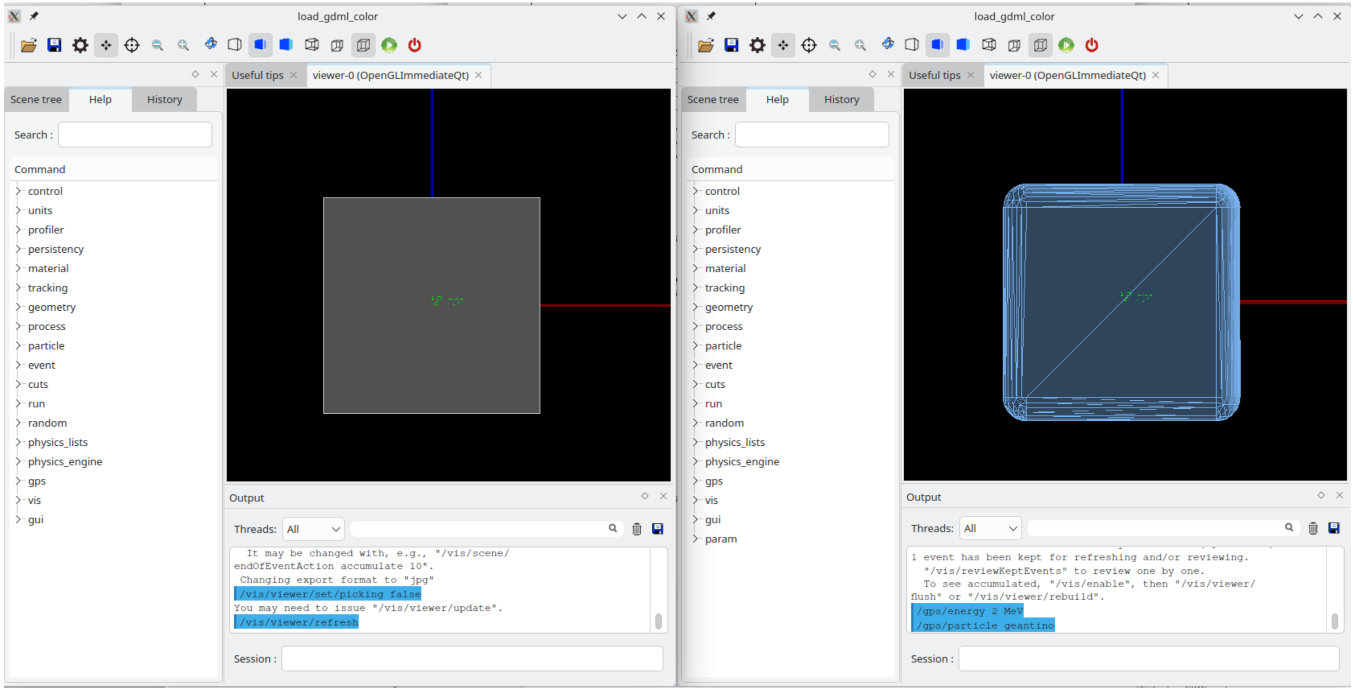
The tessellation is a 16% slower



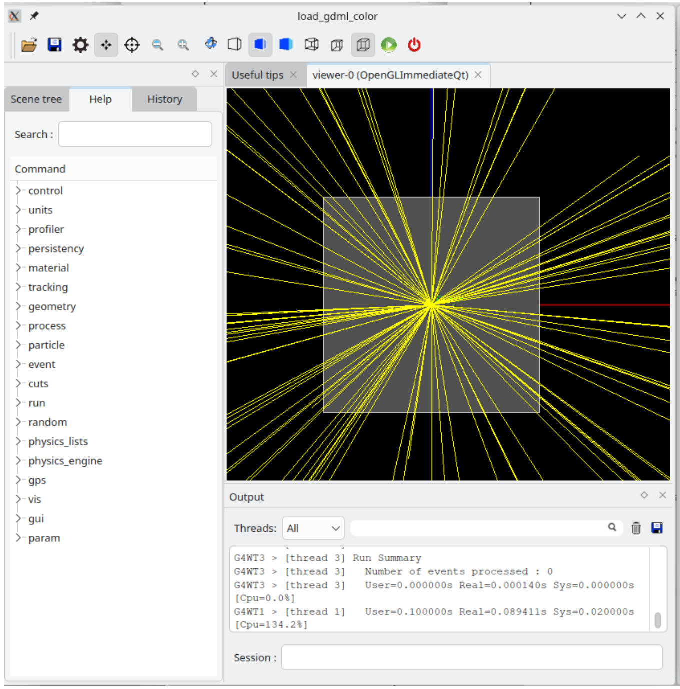
Material: G4_WATER		Beam: 2 MeV gamma; 10^6 events	
	User time (s)	Real time (s)	CPU usage
Booleans	164.4	14.2	1154%
Tessellation	567.7	48.7	1165%

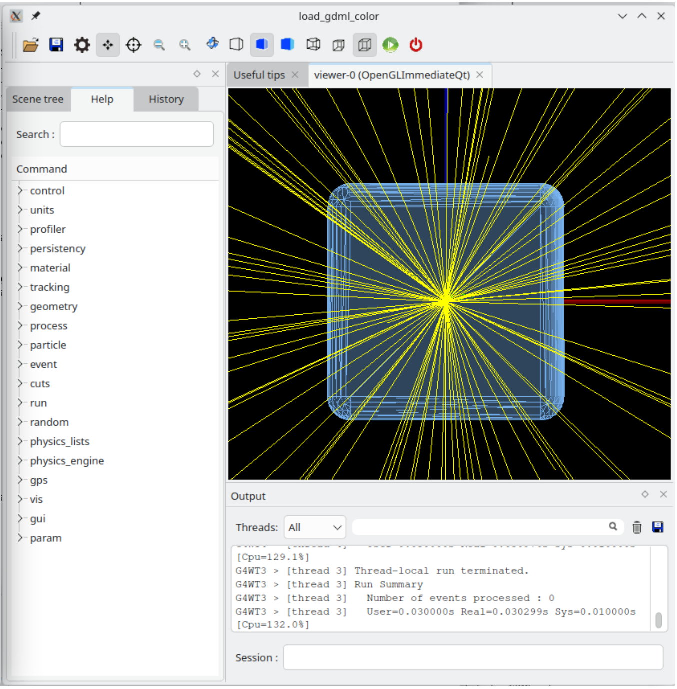
The tessellation is a factor 3.5 slower

Simple box vs filleted, tessellated box



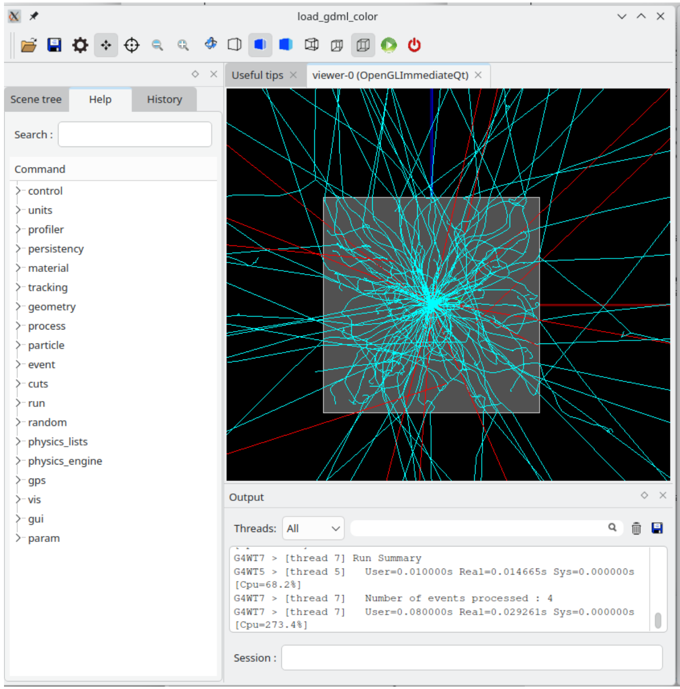
Simple box is on the left, a filleted version is on the right. The simple box is exported as a GDML Box, while the filleted version is exported as a tessellation.

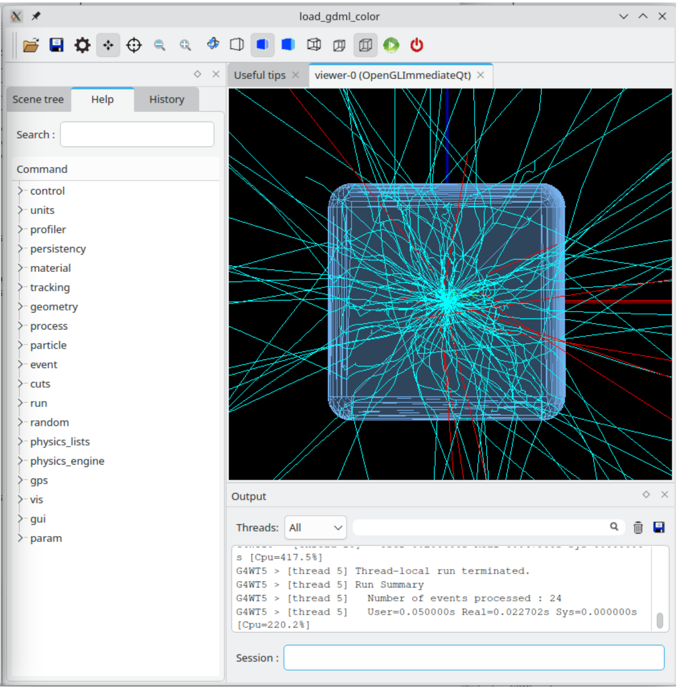




Material: G4_WATER		Beam: Geantino; 10^6 events	
	User time (s)	Real time (s)	CPU usage
Simple box	74.1	6.5	1130%
Filleted box	81.8	7.3	1128%

The fillet is 10% slower





Material: G4_WATER		Beam: 2 MeV e-; 10^6 events	
	User time (s)	Real time (s)	CPU usage
Simple box	1272	109	1168%
Filleted box	3562	305	1170%

The fillet is a factor of 2.8 slower

Conclusions

These few tests show that the timing performance is always better (shorter) for the simple intrinsic GDML solids, or solids constructed from their booleans, than for a tessellated version of the same shape. The degree of improvement, however, depends on several factors: the type of particle, the type of material, (both of these imply just the range of the particle in the material), how likely a particle will traverse the solid. The simulations with tessellated solids ranged from only 10% longer, to almost 300% or more longer. However, the length of time does not seem to be directly proportional to the number of faces. For example, for the filleted cube, there are 620 triangular facets (as compared to 6 face for the simple cube), but runs with the fillet were 3 times as long, not 100 times as long.

To state the obvious, each user should consider carefully the design of their parts and what fraction of time they expect particles in the simulation will spend interacting with that part before deciding whether a tessellation is acceptable or not.