Computer Physics Communications 239 (2019) 150–156

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

GUIMesh: A tool to import STEP geometries into Geant4 via GDML ☆

M. Pinto *, P. Gonçalves

Laboratório de Instrumentação e Física Experimental de Partículas, Avenida Gama Pinto, n.2 piso 3, Complexo Interdisciplinar (3is), 1649-003 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 26 July 2018
Received in revised form 15 January 2019
Accepted 23 January 2019
Available online 12 February 2019

Keywords:
Geant4
STEP
GDML
Radiation
Simulation
Mesh

ABSTRACT

Detailed radiation analysis of instruments flown in space is critical to ensure mission safety, often requiring the use of state-of-the-art particle transport simulation tools. Geant4 is one of the most powerful toolkits to simulate the interaction and the passage of particles through matter, but it is not prepared to receive Standard for The Exchange of Product data (STEP) files, the most versatile Computer-Aided Design (CAD) format, as input, requiring previous conversion to other CAD formats. This may lead to loss of detail and under or overestimation of the quantities under study, especially when the instruments have complex shapes, and/or a large number of volumes. Though several solutions have been proposed to import complex geometries from STEP files into Geant4, so far, only commercial options are available. In this paper we present a new tool, GUIMesh, that embeds FreeCAD libraries, an open-source CAD editor, to tessellate volumes, and convert them to Geometry Description Markup Language (GDML), a Geant4 readable format, in a straightforward way. Several degrees of freedom are given to the user regarding mesh precision and choice of material. Different geometries were tested for material definition, geometry and navigation errors, and the method used was successfully validated.

Program Summary

Program Title: GUIMesh
Program Files doi: http://dx.doi.org/10.17632/c3c9xbspzp.1
Licensing provisions: GNU General Public License 3 (GPLv3)
Programming language: Python
Nature of problem: No open-source software allows to import STEP geometries intro Geant4, one of the most powerful toolkits to simulate radiation interaction with matter. Since CAD software is extensively used in the design of particle and radiation detection instruments, it is highly desirable for physicists that STEP geometries can be imported to Geant4 with little effort instead of having to code structures made of a very large number of solid volumes which are difficult to accurately reproduce with Geant4 C++ classes.
Solution method: STEP geometries are converted to tessellated volumes (with some margin of error) using FreeCAD libraries. A Python script then writes GDML files based on the results allowing users to import these geometries with Geant4. A graphical user interface provides several options to the user, including material assignment and mesh precision setting for each volume.

1. Introduction

Particle transport simulation is fundamental for space and medical applications as well as for nuclear and particle physics. Geant4 is a C++ object-oriented toolkit developed to simulate particle interactions and track their path in materials [1–3]. Geometries in Geant4 can be described through several classes, which reproduce simple geometric shapes such as parallelepipeds, spheres and cylinders amongst others, and by combining them via Boolean operations. For more irregular solids, geometries can be defined by boundary representation such as splines, b-splines or tessellated surfaces. The later, which are used in this work, can be defined by Geant4 native class G4Tesselation or by a Geometry Description Markup Language (GDML) [4,5] file, a geometry description format based on XML. Geant4 does not import Computer-Aided Design (CAD) geometries directly but a few tools have been developed with some success though with different limitations. While some of these tools, such as CADMesh [6,7] and STL2GDML [8], only interface Stereolithography (STL) [9], a CAD mesh format, files with Geant4, requiring previous effort from CAD users to convert solids into this format, others such as SW2GDML [10] which allows SolidWorks [11] geometries to be translated into the toolkit,

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: mpinto@lip.pt (M. Pinto).

https://doi.org/10.1016/j.cpc.2019.01.024
0010-4655/

work with specific CAD formats only. These tools require that the instrument is designed with a specific CAD editor which is a major drawback since it might not be an option for the users. Finally, there is a third category of tools, that converts STEP [12] files to GDML, the most widely used data exchange CAD format. These tools, namely FASTRAD [13], ESABASE2 [14] and STEP-Tools [15] are more versatile than those in other categories but they are neither open-source nor free.

In this work, FreeCAD [16], an open-source CAD editor with Python [17] scripting capabilities, was used to tessellate volumes from STEP files. Since FreeCAD libraries can be easily imported in Python, a Graphical User Interface (GUI) was programmed with Python. This application, GUIMesh, allows users to import STEP files using FreeCAD libraries, manage materials (STEP files do not include material information), mesh volumes, and export them to GDML files. Materials are chosen from the Geant4 material database based on the NIST [18] library or created by mixing materials listed in that database. Assignment of materials to volumes can be performed on a one-to-one basis or loaded from a CSV file. Meshing is performed via FreeCADs "tessellate" function which converts all surfaces into a mesh of triangles. A GDML file is written for each volume, to provide some flexibility. An additional GDML file is also written, the "mother" GDML, which is read by the GDML parser function in Geant4, providing the geometry tree so that multiple volumes can be integrated in the toolkit.

2. Method

GUIMesh can be divided into three functionalities: CAD processing using FreeCAD to read step files and tessellate volumes, material management and GDML file writing.

2.1. Interface

The Graphical User Interface (GUI) built with Tkinter [19], a Python [17] extension, used to facilitate the procedure of converting STEP [12] geometries to GDML [4,5], is shown in Fig. 1. It is composed of three panels. A menu with eight buttons, corresponding to different user options, a volume list of the imported STEP file, and a panel with the volume properties where the volume material as well as the maximum meshing deviation (MMD) can be changed by the user. The user is also given the option not to write specific volumes to GDML. All these options can be applied to a single volume or to all of them simultaneously.

2.2. CAD processing

GUIMesh imports FreeCAD libraries with Python 2.7. Both 0.15 and 0.16 versions of FreeCAD [16] are compatible with GUIMesh though volume names may differ from the original when volumes are imported. FreeCAD imports STEP files (.STEP and .STP extensions) via its "Import" module.

Tessellation of surfaces is performed by the FreeCAD standard meshing algorithm, which has one degree of freedom, the Maximum Mesh Deviation (MMD). The default value, 0.1 mm, can be changed for each volume individually or for the whole geometry. Lower MMD values provide better precision in the geometry description but require larger memory allocation and increased CPU time at simulation level, due to the increase in the number of triangles in the tessellation. The precision for planar surfaces does not change with MMD.

2.3. Material definition

All (270) materials from the Geant4 NIST library are predefined in GUIMesh and can be accessed directly using their Name in the Geant4 Material Data Base (G4_H for Hydrogen, G4_He for Helium, etc.). These materials cannot be changed but they can be mixed to create new materials of a given name, density and number of elements, each representing a fraction of the material composition. These new materials can be created, stored and loaded in the Material Manager window that is displayed in Fig. 2. When loaded into GUIMesh all volumes are given a default material, Silicon (G4_Si), that can be altered directly in GUIMesh or latter in Geant4. In the former, materials can be assigned to volumes on a one-by-one basis, or uploaded through a CSV file containing all materials, where each line contains the volume name and the material name assigned to the volume, in the same order that is displayed in the GUI.

2.4. GDML

GDML is an XML-based language interchange format, designed to describe geometries for physics simulations. This format accepts both elementary shapes and tessellated solids. Tessellation was the method chosen for GUIMesh, since it allows to describe complex irregular solid [4,5].

GDML format is composed of five fields: define; materials; solids; structure and setup.

The field "define" is used to define different values that may be used in the rest of the file. Here is where the vertex positions of a given volume are registered. The field "materials" is used to define the materials assigned to the objects. NIST materials do not need to be defined in detail, since Geant4 will correctly interpret material included in the Geant4 Material Database reference. Any compound or mixture of materials however, may be defined in this field as a mix of NIST materials, each defined by the corresponding mass fraction in the compound or mixture.

The "solids" field corresponds to the geometric definition of volumes. The world volume is defined as a simple box with dimensions defined in GUIMesh (default size is $1\ \text{m} * 1\ \text{m} * 1\ \text{m}$). Tessellated solids from the STEP file are registered as a series of triangles enclosing a surface with the vertex that are written in the "define" field.

The "structure" part defines the geometry hierarchy. Unlike in STEP files, where the hierarchy is flat, Geant4 has a geometric hierarchy where volumes are placed inside others with the world volume as the top geometry. Since in GUIMesh each volume is written in a separate file, each volume is the top most geometry in its file. For this reason, GUIMesh also writes an extra "mother" file, which defines the world volume and allocates the other GDML files (volumes) inside it. Material references are assigned to the corresponding volumes in their respective files. There is no limit to the number of physical volumes that can be assigned to the global geometry with this method. Finally, the field "setup" identifies the top volume. For each GDML file, the top volume corresponds to the volume defined in the field "solids". For the mother GDML file, the top volume is the world volume.

The geometry thus defined can then be imported into Geant4 by the GDMLParser function [5], with the mother file as argument.

2.5. Validation

To confirm that the volumes are correctly imported into Geant4, geometry, navigation and material tests were performed. Three test geometries were defined with 0.1 mm MMD: a cylinder with a 5 mm radius and 10 mm height; a torus with 10 mm outer radius and 5 mm radius section; and a 5 mm sphere, since due to their curved surfaces they are more prone to precision problems than plane geometries. Since we expect to import several volumes at once into Geant4, a system of two spheres, where a larger 5 mm radius sphere, with a 2 mm radius spherical hole in its

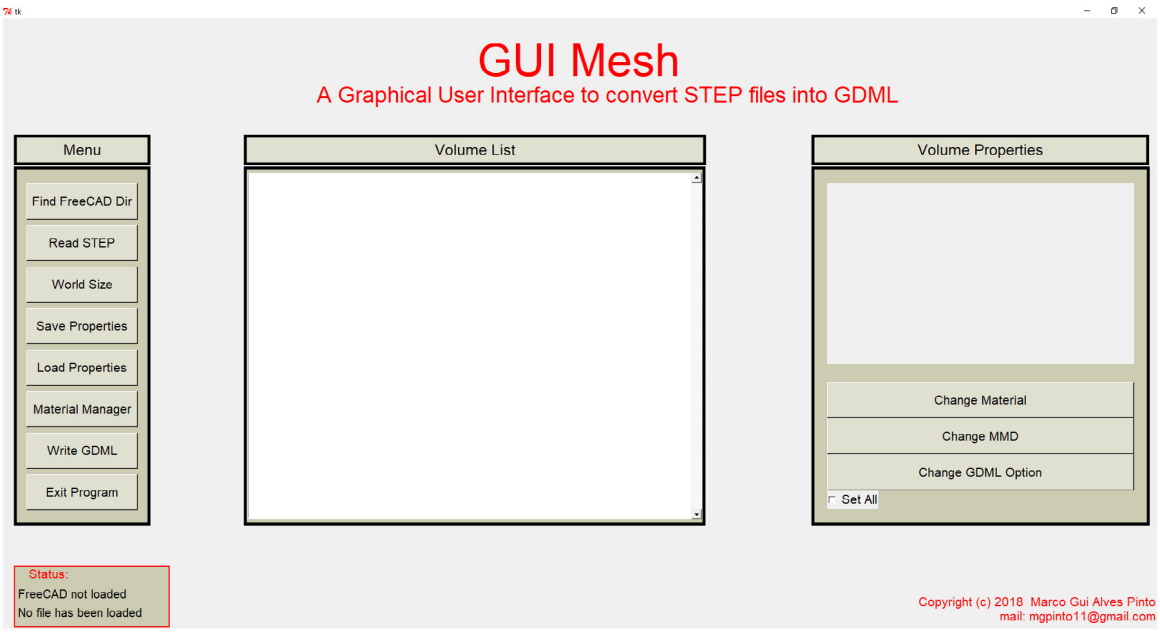

Fig. 1. GUIMesh GUI. On the left side a menu with different functions is presented to the user as buttons. The middle panel displays the list of the loaded volumes. The right side is dedicated to volume editing.

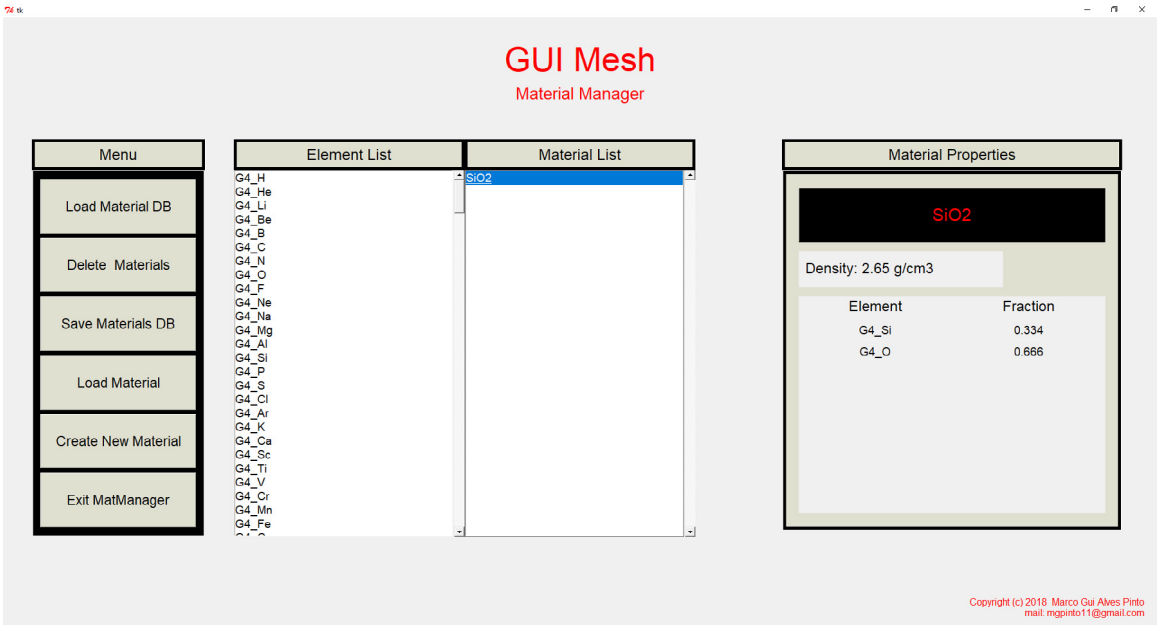

Fig. 2. Material manager interface. Left side presents the user options regarding loading, saving, deleting and material. The middle section presents the session available elements (NIST materials) and materials created by the used. The right side displays selected material properties.

center, encases a second sphere with the same size as the hole, is also tested. The sphere system was tested with several different MMDs. Fig. 3 shows all tested solids in STEP format visualized with FreeCAD.

All solids imported into Geant4 were visually analyzed for correctness. The precision of the system with two spheres was also analyzed with the Geant4 test particle, the geantino, which is a "particle" that crosses the geometry and does not interact with any materials. 10E+05 geantinos were generated omnidirectionally from a point-like source located at the center of the sphere system. The positions where the geantinos changed volumes, from the inner sphere to the outer sphere, and from the outer sphere to the outside world, were recorded. Since the spheres are meshed, the radius of the sphere is not constant in all directions. The average and maximum values of the radii of both spheres for different MMDs were thus computed. Their masses were also calculated with Geant4 "GetMass" function and compared to the value obtained from FreeCAD libraries.

Particle navigation inside the geometries was also tested. Errors may occur in Geant4 due to particles getting stuck near triangular faces. A particle is considered stuck in the geometry when it does not change position for 10 consecutive steps. When this happens Geant4 slightly moves the particle by 10E−7 mm. To account for navigation errors 10E+5 geantinos were generated omni-directionally from an 8 mm radius spherical source surrounding the test volumes. All solids presented in Fig. 3 were submitted to this navigation error test.

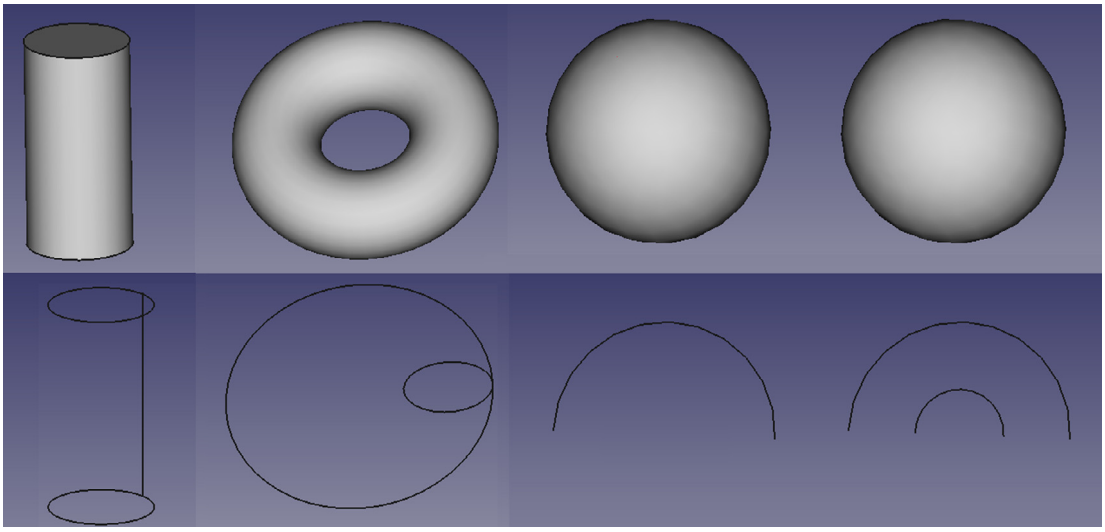

Fig. 3. STEP format of all solids used in the tests, a cylinder, a torus, a sphere and a second sphere with a spherical hole in the middle to encapsulate another a sphere that fills it. Below each solid, a wire image of its volume can be found for clarity.

Table 1
Summary of the solids/tests performed.

Solids			Tests			
Shape	Mesh max deviation (mm)	Area/MMD (#/mm^2)	Geometry		Navigation	Energy deposition
			Visual	Precision		
Cylinder	0.1	7.85E+03	×	–	×	×
Torus	0.1	1.97E+04	×	–	×	–
Sphere	CSG	–	–	–	–	×
	0.1	3.14E+03	×	–	×	×
	0.01	3.14E+04	–	–	–	×
Sphere System Outer + Inner	10	3.64E+01 + 5.03E+00	×	×	×	–
	1	3.64E+02 + 5.03E+01				
	0.1	3.64E+03 + 5.03E+02				
	0.01	3.64E+04 + 5.03E+03				
	0.001	3.64E+05 + 5.03E+04				

An additional test was performed to compare the distributions of energy deposited in meshed and Constructive Solid Geometry (CSG) spheres. This was done for each type of solid (meshed and CSG) with Silicon assigned to them, generating a parallel beam of electrons with a flat energy spectrum ranging from 100 keV to 500 MeV in a circular plane, with the same radius as the sphere (5 mm), 10 mm away from it, and registering the deposited energy in the volume. The same source was used to compare deposited energy in a cylinder filled with an elementary material, iron (Fe), and a composed material, silica (SiO2), assigned to the volume with Geant4 and with GUIMesh.

A summary of the solids, and of their characteristics, used for each test is shown in Table 1. The total surface area to MMD ratio is also given since it can be used as reference for the mesh precision for a given shape. Notice that the precision is smaller in the 2 mm radius sphere than in the 5 mm radius sphere for the same MMD, since mesh deviation will be larger when compared to the total surface area. All tests showed in this paper were executed with Geant4.10.01.p02.

2.6. Benchmarking

To understand how tessellation affects simulation time and memory usage, 10E+05 geantinos runs with a point source located in the center of the geometry, but without any readout, were performed for the Sphere System mesh with the five MMD values mentioned in Table 1, as well as for a similar configuration, implemented with CSG solid class in Geant4. All simulations ran on a system with an Intel(R) Xeon(R) CPU E5540 @2.53 GHz with 4 GB available memory.

3. Results

Several tests to the imported geometry and to particle tracking inside the volumes as well as deposited energy were performed to validate the method. CPU time and memory usage were also compared using the sphere system mesh with different MMD values and for one defined by CSG with Geant4 user classes.

3.1. Geometry

All solids were successfully imported into Geant4. Fig. 4 shows the imported solids, where all single volume geometries were meshed with 0.1 mm MMD. The two-sphere system was meshed with 10 mm, 1.0 mm, 0.1 mm, 0.01 mm and 0.001 mm to understand the effect of this parameter. The shape of the spheres is highly dependent on the MMD. In fact, in Fig. 4 we can see that we can have a shape that no longer resembles the original sphere from Fig. 3 (10 mm MMD) and good spherical approximations (0.1 mm, 0.01 mm and 0.001 mm MMD). The Geant4 "CheckOverlaps" function was used with 10E+05 test points. No overlaps were found in any case.

The average and maximum deviation of the radius in the meshed spheres are given in Table 2. As expected, both the average and maximum deviation become smaller for smaller MMD. It is also noticeable that the inner surface radius has a larger deviation

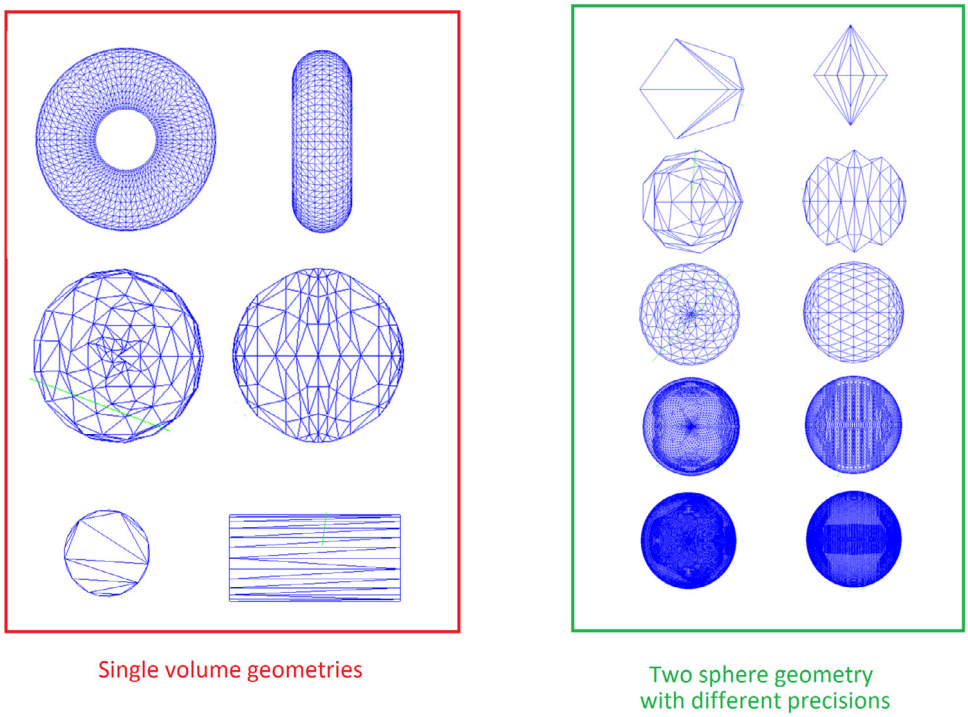

Fig. 4. Solids imported with Geant4. In the red box the single geometry solids mesh with 0.1 mm MMD can be seen. In the green box the two-sphere system can be seen (only the outside sphere is visible) meshed with 10 mm, 1 mm, 0.1 mm, 0.01 mm and 0.001 mm from top to bottom . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

than the outer one since both were meshed with the same MMD though their sizes are different. The mass calculated with Geant4 for the meshed geometries (with silicon as material) are compared with CSG/CAD in Table 2. Again, it can be seen that, the lower the MMD, the closer the mass is to the expected value. For the lower MMD spheres this difference is negligible. The mass value is always lower in the case of meshed geometries due to the nature of the mesh generation algorithms.

3.2. Navigation

Navigation tests showed no stuck events for all geometries, except the largest MMD sphere system where the percentage of stuck events was 2.84% ± 0.02%. This is due to the existence of small angles between the triangle edges, which result in large sections of the solid where the dimensions are small enough to affect position determination by Geant4.

3.3. Deposited energy

The distributions of the energy deposited by electrons, with energies ranging from 100 keV to 500 MeV, in three 5 mm radius spheres, two of them meshed with different precisions and a third defined using the Geant4 CSG sphere class, are shown in Fig. 5. The lower precision meshed sphere (green line) shows different deposited energy when compared to the other two tested solids, although the total deposited energy differs from the two other cases by approximately 1%. It is noticeable that the deposited energy in this sphere has a step like behavior due to the transition and shape of its triangles. The other two energy profiles, for the more precise meshing and for the CSG sphere implementation, display an excellent agreement.

Element and compound material definition by GUIMesh were also tested. Deposited energy in the two tested materials, SiO2 and Fe, has very good agreement between GDML and Geant4 implementation, as shown in Fig. 6. Deposited energy in SiO2 (green and black) and in Fe (blue and teal) show the same behavior for the whole range of energies. Material properties reported by Geant4 after the geometry was loaded is also the same in both cases.

Table 2
Average and maximum deviation from the mathematical surfaces of the sphere systems.

MMD (mm)	Surface	Average deviation (%)	Maximum deviation (%)	Mass deviation (%)
10	Outer (5 mm)	41.4032	63.1302	74.36
	Inner (2 mm)	41.403	63.1302	74.26
1	Outer (5 mm)	5.0308	17.4846	12.88
	Inner (2 mm)	11.5995	34.002	29.44
0.1	Outer (5 mm)	0.8344	1.6554	2.29
	Inner (2 mm)	1.7125	5.0685	5
0.01	Outer (5 mm)	0.07	0.3052	0.11
	Inner (2 mm)	0.166	0.7675	0.50
0.001	Outer (5 mm)	0.0092	0.0326	0.03
	Inner (2 mm)	0.02	0.0845	0.04

3.4. Benchmarking

Benchmarking was done by performing ten runs, each with 10E+5 geantinos, generated from the center of the sphere system with different implementations of varying precision meshed geometries and for a CSG implementation, all with the same material, Silicon. Fig. 7 shows the CPU time and memory used to load the geometry as well as the CPU time for the whole simulation. Both the CPU time and memory increase exponentially for lower MMD. It is important to take this into account when the geometry is complex and/or there are a large number of volumes. Notice that CSG is always faster and requires less memory, even when compared to the sphere system meshed with the largest MMD value.

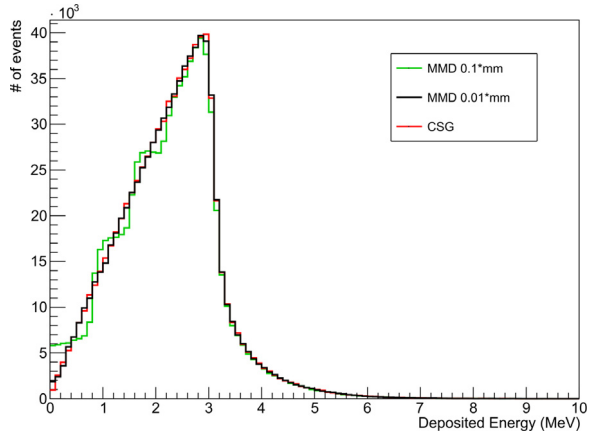

Fig. 5. Comparison between the deposited energy of electrons in three spheres, two meshed with different precisions and one implemented with a Geant4 CSG class. Energy deposited in the 0.01 mm MMD sphere (black line) is similar to the deposited energy in the CSG sphere (red line). The energy spectrum in the 0.1 mm MMD sphere (green line) however is different due to its structural defects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

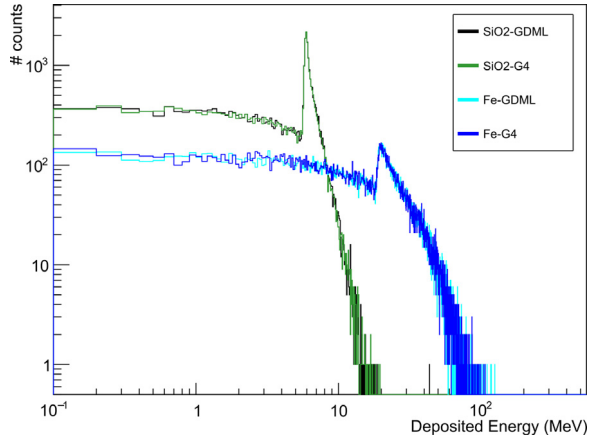

Fig. 6. Comparison between the deposited energy of electrons in two materials, SiO2 (green and black) and Fe (blue and teal), defined by Geant4 and by GDML. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

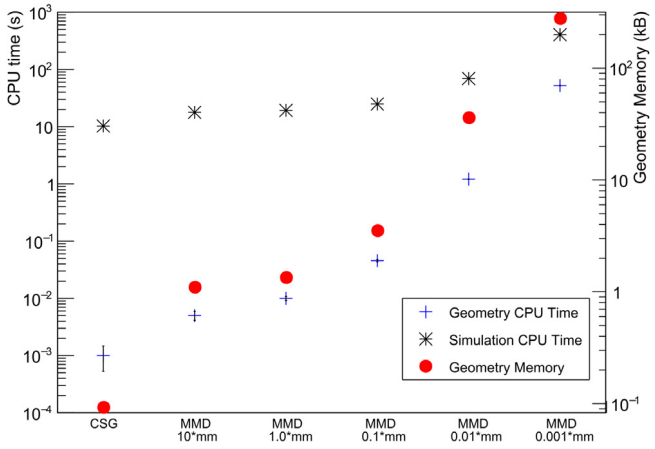

Fig. 7. Comparison of CPU time for the whole simulation and to load the geometry between different mesh precisions of the sphere system and CSG solid. Memory required to load the geometry is also shown.

4. Discussion

Several geometries in STEP format were successfully imported into Geant4 via GUIMesh. No holes were found in the tessellated geometries. Also, no overlaps were detected since all triangles of the meshed solids are inside the original shape.

For curved surfaces, the shape form was shown to be highly dependent on the MMD. This is due to the MMD limiting the maximum distance from an edge of the mesh to the original surface that it describes, e.g., larger MMD values allow larger deviations from this surface. This results in finer tuning of the surfaces for lower MMD values, as the number of triangles increases and their size decreases. Deviation, defined as the distance from the surfaces in Geant4 to the solids mathematical definition, was also computed and shown to be smaller or of the same order of magnitude as the MMD value. This means that for the same MMD, spheres with different radii will have different normalized deviations; hence when choosing the MMD one should also consider the size of the solids to be meshed. This is especially important since, as explained in Section 2, there is also a penalty in CPU time from the lack of geometry hierarchy inherent to CAD geometries.

Navigation errors resulting from "stuck" particles in some cases were studied. These happen when Geant4 is not able to compute in which volume the particle is located. No errors occurred in all geometries except the 10 mm MMD sphere system where the percentage of events displaying this behavior was under 3%. The distributions of deposited energy were found to depend on the meshing precision. In fact, while the distribution of energy deposited on a 0.1 mm MMD sphere displayed discontinuities, for the 0.01 mm MMD sphere results matched perfectly those obtained for the CSG sphere. This has to do with the path of particles going through the sphere. If one crosses the sphere at the center it will travel a longer path than one that goes through it at the edge. If the sphere is ideal as in a CSG solid, the linear distance is a smooth function dropping to zero at the limit. In the case of a meshed surface this function is no longer smooth. In fact, the function changes from triangle to triangle as can be seen in the meshed solid images. For more accurate tessellations this feature is less evident due to the decrease of the step amplitude (smaller triangles). Scattering of electrons and the bin size of the deposited energy histogram also contribute to this effect. Mixed materials defined in GUIMesh were also validated.

Memory and time consumption were found to increase for solids meshed with lower MMD values, which means that for complex and/or large number of solids careful consideration should be made regarding solid precision.

5. Conclusion

GUIMesh enables users to import CAD geometries in STEP format into Geant4. A STEP file describing a global geometry is read by GUIMesh, and the geometry is meshed with a given Maximum Mesh Deviation and registered with the corresponding material in a GDML file structure, readable by Geant4. A set of test volumes were used to test GUIMesh performance in terms of geometrical accuracy, particle navigation errors and material implementation. It was concluded that geometry precision is highly dependent on the chosen value for MMD. Lower values of MMD result in more accurate geometries and fewer errors in particle tracking, at the cost of memory and processing time.

No overlaps were found in the tested geometries and it was shown that material definition in GUIMesh successfully implements materials into Geant4.

GUIMesh can be a useful tool for all users of Geant4, allowing them to import STEP geometries of arbitrary size and complexity into the simulation toolkit.

References

[1] S. Agostinelli, et al., Nucl. Instum. Methods A 506 (3)) (2003) 205.
[2] Allison J., et al., IEEE Trans. Nucl. Sci. 53 (1) (2006) 270–278.
[3] J. Allison, et al., NIMA 835 (2017) 186–225.
[4] R. Chytracek, J. McCormick, W. Pokorski, G. Santin, IEEE Trans. Nucl. Sci. 53 (5) (2006) 2892–2896.
[5] http://gdml.web.cern.ch/GDML/doc/GDMLmanual.pdf.
[6] C. Poole, et al., Austral. Phys. Eng. Sci. Med. 35 (3) (2012) 329–334.
[7] C. Poole, I. Cornelius, J. Trapp, C.M. Langton, IEEE Trans. Nucl. Sci. 99 (2012) 1–7.
[8] https://www.solveering.com/InStep/instep.aspx (accessed on 14-11-18).
[9] 3D Systems, Inc. Stereolithography Interface Specification, July 1988.
[10] C. Vuosalo, et. al, J. Phys.: Conf. Ser. 898 (2017) 042024.
[11] Dassault Systèmes SOLIDWORKS Corporation, Waltham, Massachusetts.
[12] STEP-file, ISO 10303-21 – Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure.
[13] T. Beutier, E. Delage, M. Wouts, O. Serres, P.F. Peyrard, Fastrad new tool for radiation prediction; 2003 – RADECS – Proceedings of the 7th European Conference on Radiation and Its Effects on Components and Systems, Volume, Issue, Page(s): 181-183.
[14] https://esabase2.net/wp-content/uploads/2017/09/ESABASE2-Fact-Sheet.pdf (accessed on 14-11-18).
[15] https://www.steptools.com/ (accessed on 14-11-18).
[16] https://www.freecadweb.org/ (accessed on 14-11-18).
[17] http://www.python.org/ (accessed on 14-11-18).
[18] https://physics.nist.gov/cgi-bin/Star/compos.pl (accessed on 14-11-18).
[19] https://wiki.python.org/moin/TkInter (accessed on 14-11-18).