
ARTICLE

SimpleGeo – New Developments in the Interactive Creation and Debugging of Geometries for Monte Carlo Simulations

Christian THEIS^{1,*}, Karl Heinz BUCHEGGER², Eduard FELDBAUMER¹, Doris FORKEL-WIRTH¹,
Lukas JAEGERHOFER¹, Stefan ROESLER¹ and Helmut VINCKE¹

¹ CERN DGS-RP, Geneva 23, CH-1211 Geneva, Switzerland

² GASCAD, Teichstrasse 2, A-4595 Waldneukirchen, Austria

Nowadays radiation transport Monte Carlo simulations have become an indispensable tool in various fields of physics. The applications are diversified and range from physics simulations, like detector studies or shielding design, to medical applications. Usually a significant amount of time is spent on the quite cumbersome and often error prone task of implementing geometries, before the actual physics studies can be performed. SimpleGeo is an interactive solid modeler which allows for the interactive creation and visualization of geometries for various Monte Carlo particle transport codes in 3D. Even though visual validation of the geometry is important, it might not reveal subtle errors like overlapping or undefined regions. These might eventually corrupt the execution of the simulation or even lead to incorrect results, the latter being sometimes hard to identify. In many cases a debugger is provided by the Monte Carlo package, but most often they lack interactive visual feedback, thus making it hard for the user to localize and correct the error. In this paper we describe the latest developments in SimpleGeo, which include debugging facilities that support immediate visual feedback, and apply various algorithms based on deterministic, Monte Carlo or Quasi Monte Carlo methods. These approaches allow for a fast and robust identification of subtle geometry errors that are also marked visually.

KEYWORDS: Monte Carlo, geometry, visualization, FLUKA, PHITS

I. Introduction

Particle transport codes commonly use textual description and supply their own scripting language, which makes the creation of geometries a non-trivial undertaking due to the lack of immediate visual feedback. SimpleGeo is an interactive three dimensional solid modeler, which allows for intuitive creation of geometries for simulations. In general, SimpleGeo can be classified as a hybrid solution as it retains the analytical description of the combinatorial solid geometry (CSG) tree for export, while a boundary representation 3D model of the former is created online and used for visualization. CSG is a modeling approach which has been popular already since the 60s and was subsequently used as the geometry description and visualization method in numerous tools like POV-Ray,¹⁾ BRL-CAD²⁾ or MCNP VisEd.³⁾ The CSG method itself is a so called image space approach and depending on the available hardware, the complexity of the scene as well as the required image quality is often limited to static visualization.⁴⁾ Interactive visualization of geometries based directly on CSG models has only become possible in the recent years with special implementations utilizing specialized graphics processing units (GPU) and the level of interactivity is still limited significantly by the complexity of the geometry. Boundary representation models however lend themselves easily to being visualized with high performance by standard, even low-cost, graphics hardware. These me-

thods are so called object-space approaches which are based on rasterizing triangulated meshes. This approach allows for immediate visual feedback and thus, the required time to build geometries is significantly reduced for complex problems and the whole process is facilitated especially for novice users. Therefore, SimpleGeo is based on a hybrid solution which on one hand allows for retaining the original description of the geometry models using simple, yet powerful CSG expressions, while providing fast interactive inspection of the results.

II. New Developments

1. Improvements in the Solid Kernel

Due to its design SimpleGeo is independent of any specific Monte Carlo code and its first release included the import, export and visualization of FLUKA geometries.^{5,6)} Further work has been done and a number of import and export modules have been added, which now include an exporter for MCNPX and a preliminary export module for PHITS.^{7,8)} In addition, a number of CAD file formats (Wavefront OBJ, AutoCAD DXF, 3D Studio, VRML, PBRT, STL and PLY) are supported for export, which can be read by ray-tracing packages that can be used for the creation of photo-realistic images.

As explained in Ref. 4) SimpleGeo is a hybrid solid modeler,⁴⁾ which simultaneously retains the analytic description of the CSG tree as well as an approximated boundary representation (B-rep), which is used for the rendering. This

*Corresponding author, E-mail: christian.theis@cern.ch

approach converts solids into a collection of polygons that constitute the shell of the respective region. One of the main advantages of using a hybrid representation of CSG models with boundary representations is that integral properties like volumes can be computed in a very elegant and fast way. The approach used is a direct extension of the idea of calculating the area of an arbitrary polygon in 2D. In three dimensions this method utilizes the integration over the signed volumes of tetrahedrons, into which all solids can be partitioned. Due to space restrictions the complete mathematical background cannot be given here but a more detailed description of the algorithm can be found in Ref. 10).

2. Debugging

Naturally, the creation of error-free and valid geometry models is of great importance. However, the definition of validity differs greatly with respect to the application. For engineering purposes the geometry must be visually correct, meaning that all positions, dimensions and also materials of each used entity must show resemblance to the real construction at an appropriate detail-level. In contrast to this, the constraints imposed upon models used for Monte-Carlo simulations are more stringent. Due to the fact that particle interactions are strongly dependent on material properties it is mandatory that each point in 3D space is uniquely associated with one region, which a material must be assigned to. This association must not be ambiguous as it would lead to incorrect results or it could crash the simulation if there was no link to a region description at all. Consequently, it is very important to identify areas which do not belong to any region or to multiple regions. Therefore, sophisticated and preferably automated approaches for checking the validity must be devised.

One solution to this problem is to superimpose a specified grid of points onto the geometry and to test for each point, whether it is unambiguously defined within a region. Monte-Carlo simulations usually use their built-in tracking algorithms for this task. In the case of SimpleGeo an independent tracker was implemented which can be queried for containment. Due to the object oriented design and the hybrid combination of a CSG and B-rep data structure, any newly implemented primitive type is automatically incorporated into the tracker without requiring any additional programming. The reason for this is that by default the containment tests are performed using solid angle based point-in-polyhedron tests which operate on the boundary representations. Therefore, these tests are available even for primitives for which no analytic test can be performed in a simple way. However, the result is on one hand limited by the accuracy of the surface mesh representation and on the other hand requires quite a lot of computational overhead for numerous surface patches. Therefore, analytic tests are implemented for most primitives (like spheres, cylinders, boxes, etc.), which override the default method. Naturally, in these cases the issue of limited accuracy due to a triangulated boundary representation is irrelevant.

A major drawback of the approach using a regular grid of points is that the user has to define a volume and a specific

grid-spacing that should be considered for debugging. If the step-size would be infinitesimally small one would detect an error with a certainty of 100%. However, this is usually not possible in practice. Using finite step-sizes that are larger than the smallest region, errors in such volumes cannot be found by definition, unless the chosen origin and the step-size of the regular grid allows for placing a sample directly in the erroneous region. However, this would require some a-priori information which is contradictory to the commonly encountered situation of initially debugging geometries without any prior knowledge. It is obvious that the probability to find errors in very small volumes is strongly dependent on the grid parameters that have to be chosen beforehand.

In order to improve this situation some research has been conducted on the development of alternative debugging strategies. To some extent the process of sampling and checking points for their containment within a geometric region fits the point sampling paradigm for the evaluation of integrals using Monte Carlo methods. Therefore, the first additional algorithm that was implemented is based on purely random samples within a region that has been indicated by the user. Due to the fact that no grid with fixed grid-sizes is used it is more probable to detect small problematic regions. The second stochastic method is based on importance sampling applying a parameterized probability distribution function. Although these random sampling strategies are certainly more favorable than the approach of a regular grid they are not free of shortcomings either. Random sampling based on pseudo-random numbers commonly results in clustering which is particularly obvious with a low number of samples. Consequently, a large sampling population is required to obtain a distribution covering the sampling space uniformly.

Most simulation algorithms are based on independent, identically distributed uniform random variables. However, it is intuitively clear that for the purpose of debugging the efficiency can be improved by a strategy that yields a higher measure of uniformity for an identical number of samples. A solution was eventually found by studying a family of deterministic approaches – so called Quasi-Monte-Carlo techniques – which drop the requirement of statistical independence.¹¹⁾ These methods have the advantage of delivering distributions that cover the sampling space uniformly already with a comparably low number of samples.

The quality of the uniform distribution in the s -dimensional interval $[0,1]^s$ can be evaluated by looking at regions in this domain and the number of points inside these regions. The number of the points inside one region in relation to the total number of points should approximately resemble the region's volume in relation to the total volume of the domain. For example, one third of the points should cover exactly one third of the total volume. This ideal case would allow for estimating the covered volume in relation to the total volume by the number of points in that volume in relation to the total number of points. The difference between the actual volume and the volume estimated by the points is called discrepancy and therefore, can be regarded as a measure for the quality of the uniform distribution of points in a domain. This notion can be formalized mathematically by

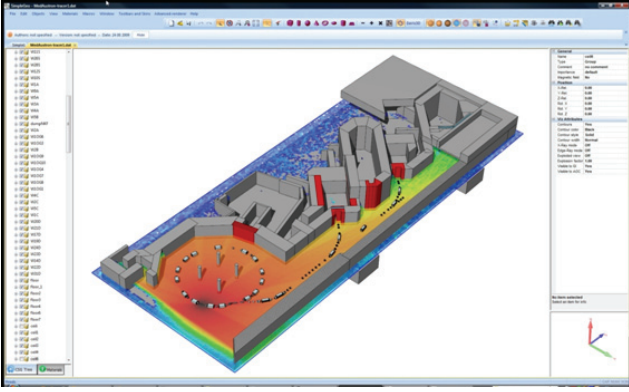


Fig. 1 Screenshot of SimpleGeo showing the 3D model of a hadron therapy facility¹⁴⁾ and the associated dose-rate values

writing the discrepancy D^* as a measure of the deviation of a point set $U_N = \{x_0, \dots, x_{N-1}\}$ from a uniform distribution:

$$D^*(U_N) = \sup_{A=\prod_{j=1}^s [0, a_j]} \left| \lambda_s(A) - \frac{1}{N} \sum_{i=0}^{N-1} \chi_A(x_i) \right| \quad (1)$$

with denoting families of subsets of the sampling space I^s in the interval $[0, a_j]$, where a_j represents the upper boundary of the point set U_N in the respective dimension j . λ_s is the Lebesgue-measure in the sampling space, and χ_A represents the characteristic function of the set A . Equation (1) can be interpreted as the largest error of the quadrature defined by U_N when integrating characteristic functions of subcubes A of the unit cube $I^s = [0, 1]^s$. It should be noted that D^* is an anisotropic measure and as such it is not invariant to rotation. Therefore, the sampling domain that is applied by the Quasi-Monte-Carlo approaches in SimpleGeo is represented by an axis aligned box, similarly to the grid-sampling approach. From Eq. (1) follows intuitively:

$$\lim_{N \rightarrow \infty} D^*(U_N) = 0 \Leftrightarrow x_i \in U_N \quad (2)$$

are uniformly distributed in I^s . Naturally, in practice it is not possible to obtain discrepancy of zero but the application of certain deterministic patterns allows for minimizing the deviation. As shown by Niederreiter the discrepancy of random point sets U_N^{random} and deterministic low-discrepancy point sets U_N^{ld} compares as follows:¹²⁾

$$D^*(U_N^{random}) \in \mathcal{O}\left(\sqrt{\frac{\log \log N}{N}}\right), D^*(U_N^{ld}) \in \mathcal{O}\left(\frac{\log^s N}{N}\right) \quad (3)$$

As can be seen these specially designed deterministic low-discrepancy point sets U_N^{ld} , like van der Corput, Halton or Hammersley sequences, are clearly preferable over pseudo-random sampling in order to obtain a thorough coverage of the debugging volume. The mathematical details of the sampling methods used in SimpleGeo naturally exceed the scope of this paper, but a more detailed discussion of them can be found in the following reference.¹³⁾ The current version of SimpleGeo provides a number of different Monte-Carlo and Quasi-Monte-Carlo based debugging algorithms. These are also available in parallelized versions that are targeted specifically towards utilizing the full parallel power of modern multi-core CPUs.

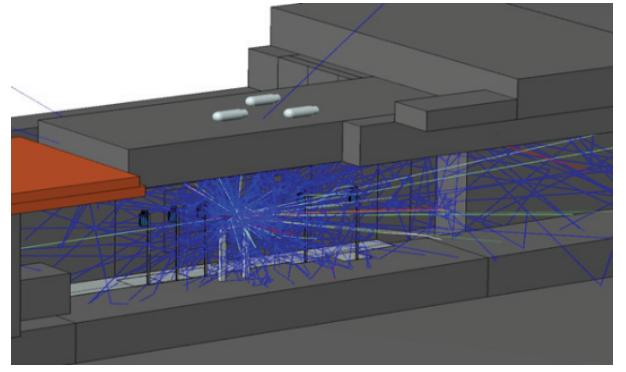


Fig. 2 Visualization of hadronic tracks at the CERF facility at CERN, originating from one proton impinging on a copper target

3. Plugin Interface

In order to keep the modeler open and extensible a plugin interface has been developed. This allows for extending the software with additional functionality at runtime. Basically, these plugins can access the modeling kernel and can thus be used to automatize certain modeling procedures or to implement the import or export of 3rd-party file formats. In addition, an interface layer has been implemented that allows these plugins to integrate seamlessly into the rendering pipeline of SimpleGeo at runtime. Consequently, these extensions can also be used for data visualization in combination with geometry models.

Nowadays color contour plots provide an illustrative means to convey sometimes quite complex results. The creation of these plots from the data obtained from Monte Carlo programs is in general quite tedious and very often implies the application of numerous scripts and other utilities. DaVis3D is a plugin which was developed to facilitate this process in an interactive way. The user can directly load 3D data, selecting from a number of proprietary formats like FLUKA's USRBIN output, MCNPX mesh tallies or an open-format ASCII file. Subsequently it is possible to select one to three projection planes that can be moved interactively through the dataset, while the current data can be displayed superimposed onto the geometry (see **Fig. 1**). In addition various means of manipulating (averaging, summing, etc.) querying or extracting data interactively are provided as well.

For various applications it is helpful to actually display the tracks of particles obtained from the Monte Carlo simulation. This can be used to investigate specific behavior as well as to verify and understand results. PipsiCAD3D is a plugin for SimpleGeo which is based on a previous tool and can be used to interactively display particle tracks as well as their evolution as a function of time (see **Fig. 2**).¹⁵⁾

III. Conclusion

The latest developments in SimpleGeo include a debugger which provides immediate visual feedback of the errors as well as various methods to find even problems in very small regions of the geometry efficiently. In addition, the import and export modules for FLUKA have been extended and the

performance of the modeling kernel was greatly improved. A plugin mechanism was developed to allow for extending the functionality beyond the pure modeling capabilities in view of data display and analysis. A number of plugins have already been developed for these tasks. Future work will include the native import of geometries from other Monte Carlo codes like MCNPX and further improvements of the proprietary renderer as well as the data analysis plugins.

Acknowledgment

The authors would like to thank Christina Urscheler, Markus Brugger, Alessandro Porta and Matthias Karacson for their patient beta testing and feedback, as well as Thomas Kollig and Alexander Keller for the kind permission to adapt their Quasi-Monte Carlo implementations as a basis for our sampling methods. Furthermore, we would like to thank Tatsuhiko Sato for his great efforts to provide feedback and his help in the implementation of the PHITS export module.

References

- 1) Persistence of Vision Raytracer Pty. Ltd, PoV-Ray, <http://www.povray.org>
- 2) Mike Muuss, BRL-CAD, <http://brlcad.org>
- 3) R. A. Schwarz, L. L. Carter, N. Shrivastava, "Creation of MCNP Input Files With a Visual Editor," *Proc. of the 8th Int. Conf. on Radiation Shielding*, 454-459, Arlington, Texas, *Published by the American Nuclear Society* (1994).
- 4) C. Theis, K. H. Buchegger, M. Brugger, D. Forkel-Wirth, S. Roesler, H. Vincke, "Interactive three-dimensional visualization and creation of geometries for Monte Carlo calculations," *Nucl. Instr. Meth. Phys. Res.*, **A562**, 827-829 (2006).
- 5) A. Fassò, A. Ferrari, J. Ranft, P. R. Sala, *FLUKA: a multi-particle transport code*, CERN Yellow Report, INFN/TC_05/11, SLAC-R-773 (2005).
- 6) G. Battistoni, S. Muraro, P. R. Sala, F. Cerutti, A. Ferrari, S. Roesler, A. Fassò, J. Ranft, "The FLUKA code: Description and benchmarking," *Proc. of the Hadronic Shower Simulation Workshop 2006*, Fermilab 6--8 September 2006, M. Albrow, R. Raja (eds.), *AIP Conf. Proc.*, **896**, 31-49 (2007).
- 7) H. G. Hughes, R. E. Preal, R. C. Little, *MCNPX - The LAHET/MCNP Code Merger*, X-Division Research Note XTM-RN(U)97-012, LA-UR-97-4891, Los Alamos National Laboratory (LANL) (1997).
- 8) K. Niita, T. Sato, H. Iwase, H. Nose, H. Nakashima, L. Sihver, "PHITS-a particle and heavy ion transport code system," *Radiat. Meas.*, **41**, 1080-1090 (2006).
- 9) E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional (1995).
- 10) M. Mantyla, *An Introduction to Solid Modeling*, Computer Science Press, (1988).
- 11) T. Kollig, A. Keller, "Efficient multidimensional sampling," *Comput. Graph. Forum*, **21**[3], 557-563 (2002).
- 12) H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Pennsylvania, (1992).
- 13) C. Theis, *The radiation environment in underground workplaces of the LHC*, CERN-THESIS-2009-027, CERN/TU-Graz, (2007).
- 14) L. Jaegerhofer, *private communication* (2010).
- 15) H. Vincke, "FLUKACAD/PIPSICAD: A 3 dimensional interface between FLUKA and Autocad," *Proc. of the SATIF-5 workshop*, 361 - 370 (2000).