PENGEOM – A general-purpose geometry package for Monte Carlo simulation of radiation transport in complex material structures (New Version Announcement)

Julio Almansa[a], Francesc Salvat-Pujol[b], Gloria Díaz-Londoño[c], Artur Carnicer[d], Antonio M. Lallena[e], Francesc Salvat[d,*]

[a] *Servicio de Radiofísica y P.R., Hosp. Univ. Virgen de las Nieves, Avda. de las Fuerzas Armadas 2, 18014 Granada, Spain*

[b] *CERN, Esplanade des Particules 1, 1211 Geneva 23, Switzerland*

[c] *Escuela de Física, Universidad Nacional de Colombia-Sede Medellín, Carrera 65 Nro. 59A - 110, Colombia*

[d] *Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain*

[e] *Departamento de Física Atómica, Molecular y Nuclear, Universidad de Granada, 18071 Granada, Spain*

Abstract

A new version of the code system PENGEOM, which provides a complete set of tools to handle different geometries in Monte Carlo simulations of radiation transport, is presented. The distribution package consists of a set of Fortran subroutines and a Java graphical user interface that allows building and debugging the geometry-definition file, and producing images of the geometry in two- and three dimensions. A detailed description of these tools is given in the original paper [*Comput. Phys. Commun.* **199** (2016) 102–113] and in the code manual included in the distribution package. The present new version corrects a bug in the Fortran subroutines, and it includes various improvements of the Java graphical user interface.

Keywords: Constructive quadric geometry, Monte Carlo particle transport, Ray tracing, Geometry visualization

*Corresponding author.
E-mail address: francesc.salvat@ub.edu

NEW VERSION PROGRAM SUMMARY

Program Title: PENGEOM
CPC Library link to program files: (to be added by Technical Editor)
Code Ocean capsule: (to be added by Technical Editor)
Licensing provisions(please choose one): CC BY NC 3.0
Programming language: Fortran 90, Java
Journal reference of previous version: Comput. Phys. Commun. **199** (2016) 102–113
Does the new version supersede the previous version?: Yes
Reasons for the new version: The original Fortran code contained a subtle bug, which had an effect only when 1) a particle moving within a void region entered a material body, and 2) the particle was left in the void volume because of numerical roundoff errors (undershot). In such case, the steering main program erroneously concluded that the particle was in the *external* vacuum and, consequently, it discontinued the simulation of the particle trajectory. This anomalous behavior has been corrected by shifting the position of the particle a small distance, determined by the parameters of the master ray equation, which places the particle a small distance inside the material body.
In the original Fortran subroutines, inconsistencies in the geometry-definition file resulted in a stop of the program; this reaction was hard to control from a graphical user interface. In the new version, any apparent syntax error is resolved by returning the control to the calling program with an error flag activated and with an error message.
Various aspects of the Java user interface have been improved: the text editor now includes line numbers, which are used to locate possible inconsistencies and errors in the geometry definition file; the size of the application window can now be modified, and the possible sizes of the graphics windows are adapted to the actual resolution of the monitor; the action of keystroke commands under Windows, Linux, and macOS is now the same. The present new version has been verified to work correctly under Java 8, Java 11, and Java 13.
Nature of problem: The Fortran subroutines perform all geometry operations in Monte Carlo simulations of radiation transport with arbitrary interaction models. They track particles through complex quadric geometries, *i.e.*, through material systems consisting of homogeneous bodies limited by quadric surfaces. Particles are moved in steps (free flights) of a given length, which is dictated by the simulation program, and are halted when they cross an interface between media of different compositions or when they enter selected bodies. At the end of each step, PENGEOM returns the indices of the material and body where the particle is.
Solution method: PENGEOM is tailored to optimize simulation speed and accuracy. Fast tracking is accomplished by the use of quadric surfaces, which facilitate the

calculation of ray intersections, and of modules (connected volumes limited by quadric surfaces) organized in a hierarchical structure. Optimal accuracy is obtained by considering fuzzy surfaces, with the aid of a simple algorithm that keeps control of multiple intersections of a ray and a surface. The 64-bit Java GUI PenGeomJar provides a complete geometry toolbox; it allows building and debugging of the geometry definition file, as well as visualizing the resulting geometry in two and three dimensions.
Additional comments including restrictions and unusual features: All geometrical operations are performed internally. The connection between the steering main program and the tracking routines is through a Fortran module, which contains the state variables of the transported particle, and the input-output arguments of subroutine `STEP`. Rendering of two- and three-dimensional images is performed by using the PENGEOEM subroutines, so that what we see in the images is what is passed to the simulation program.
By default PENGEOM can handle systems with up to 5,000 bodies and 10,000 surfaces. These numbers can be increased by editing the source file.

[1] J. Almansa, F. Salvat-Pujol, G. Díaz-Londoño, A. Carnicer, A.M. Lallena, and Francesc Salvat, *Comput. Phys. Commun.* **199** (2016) 102–113.

Acknowledgements

We are indebted to Dr Josep Sempau for pointing out the flaw of the Fortran code that motivated the present new version. Financial support from the Spanish Ministerio de Ciencia, Innovación y Universidades / Agencia Estatal de Investigación / European Regional Development Fund, European Union, (project no. RTI2018-098117-B-C22), Ministerio de Economía y Competitividad (project no. PID2019-104888GB-I00) and Junta de Andalucía (projects nos P18-RT-3237 and FQM387) is gratefully aknowledged.