

# Construction of geometric models for Monte Carlo simulations of gamma-ray scanning technique

C.; Raitz<sup>1</sup>, C. M.; Salgado<sup>2</sup>  
e-mail: crj@puc-rio.br, otero@ien.gov.br

<sup>1</sup>PUC-RIO, <sup>2</sup>DIRA, IEN

**Keywords:** gamma-ray scanning, distillation column, numerical simulations.

The *Aplicação de Técnicas Nucleares* group of the *Instituto de Engenharia Nuclear* opened recently a line of research involving prevention and troubleshooting of industrial equipment problems using the Gamma-Ray Scanning technique (GRS). The GRS technique consists of measuring a density profile using a sealed radioactive source and a detector placed on opposite sides of an object of interest. A density profile is then measured by advancing source and detector positions along the axis of the object [1].

The MCNP-X code is a valuable tool for simulations involving radioactive sources, and has been thoroughly validated [2]. Its input consists of three blocks describing solids, surfaces, and materials, as well as the simulation options. However, with the growth of the models' complexity and numbers of scenarios, it may get hard to keep tabs on every surface and solid. To challenge the complexity of input files, we are developing a software library in Python language to facilitate the construction of geometric models for MCNP-X code. Currently, our library has the following features: (i) Object encapsulation of materials, geometric surfaces and solids; (ii) Easy geometry manipulation like copying, translating, and rotating; and (iii) built-in frequently used objects, like NaI(Tl) detectors with enclosure.

As a small example of these features, suppose we want to create a cylinder of radius 100. The following command does that (the cylinder axis defaults to Z):

```
c1 = Cylinder(R=100)
```

Then we would like to copy the cylinder and rotate it 90° about the X axis:

```
c2 = c1.copy(rx=90)
```

We can define a solid object inside these two cylinders:

```
solid = CSG(inside=[c1,c2])
```

Finally, we can create a material and assign it to the solid:

```
water = Material([["H",2],["O",1]])  
solid.setMaterial(water)
```

Running this code with a Python interpreter will generate an input file adequate for Monte Carlo simulation with MCNP-X code.

A more complex model is shown in Figure 1. It is a longitudinal view of normally operating distillation column model, created using our library, as shown by the VISED® software. Labels were added for easier visualization. Each solid is bounded by black lines and different materials have different colors. Note that each column stage is the same object group rotated and translated. Also note the detector, with multiple solids, was created with a standard detector object.

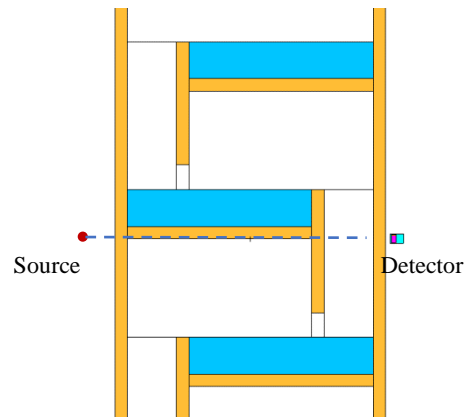


Figure 1. Model of a distillation column

In future steps, we will add the following features to our Python library: solid creation using ray-tracing to detect volume boundaries, and model viewing using the Python graphic tools. We will be generating several faulty case models for a distillation column, to analyze and compare their GRS profiles.

## References

- [1] SHAHABINEJAD, H.; FEGHHI, S. A. H.; KHORSANDI, M. Structural inspection and troubleshooting analysis of a lab-scale distillation column using gamma scanning technique in comparison with Monte Carlo simulations. *Measurement*, Amsterdam, v. 55, [s.n], p. 375-381, 2014.
- [2] PELOWITZ, D. B. *MCNP-X TM User's Manual*. Version 2.5.0. New Mexico: Los Alamos National Laboratory Report, 2005. Não paginado.