

Práctica de Laboratorio **Simulación Monte Carlo de detectores**

Tutor: Alejandro Algora / Dolores Jordán

El objetivo de esta práctica es iniciar al alumno en la utilización de métodos de simulación Monte Carlo (MC) para estudiar la respuesta de detectores a la radiación y familiarizarle con algunos conceptos fundamentales, usando el código Geant4 como ejemplo. Geant4 es un código de última generación, y consiste en un paquete de librerías C++ que el usuario puede llamar con gran libertad para construir un código de simulación específico. Esto le concede gran flexibilidad y versatilidad, comparado con otros códigos de simulación MC de uso general, a costa de exigir mayor esfuerzo por parte del usuario. Las librerías incluyen diversos paquetes de física electromagnética y hadrónica, potentes herramientas para la definición de geometrías y la posibilidad de controlar exhaustivamente el seguimiento de las partículas y la recolección de la información.

Bibliografía:

1. Alex Bielajew, *Fundamentals of the Monte Carlo method for neutral and charged particle transport*,
<http://www-personal.umich.edu/~bielajew/MCBook/book.pdf>
2. Geant4: <http://geant4.web.cern.ch/geant4/>
3. ROOT: <http://root.cern.ch/>
4. Linux: <http://www.debianhelp.co.uk/commands.htm>
5. <http://www.thegeekstuff.com/2010/11/50-linux-commands/>
6. Editors: <http://www.nedit.org/>
<http://www.gnu.org/software/emacs/>
<http://kate-editor.org/>

Materiales:

El alumno dispone del siguiente material:

1. Varios “scripts” de configuración de Geant4 (`geant4.9.6_enviroment.sh`) y de otros programas necesarios para la práctica como ROOT y programas de visualización (`root_enviroment.com`, `dawn_visualizer.com`, `freewrl_visualizer.com`) en el directorio *work*. También el usuario encontrará un script de compilación (`GNUmakefile`) y tres macros de ejecución del programa (`vis.mac`, `print-track.mac`, `spec-eff.mac`) en el directorio *test*.
2. Un programa simple de simulación basado en Geant4: `test.cc`, situado en el directorio `/home/Escuela2012/work/test/` y las subrutinas relacionadas que se encuentran en el subdirectorio `./src/` y sus cabeceras correspondientes en el subdirectorio `./include/` (o las clases y su definición en lenguaje C++)
3. Herramientas de visualización (`dawn`, `freewrl`) y análisis de datos ROOT. Se incluye un macro de ejecución (`hplot.cpp`) para este último.
4. Entorno de trabajo en Linux (`bash`)

Realización:

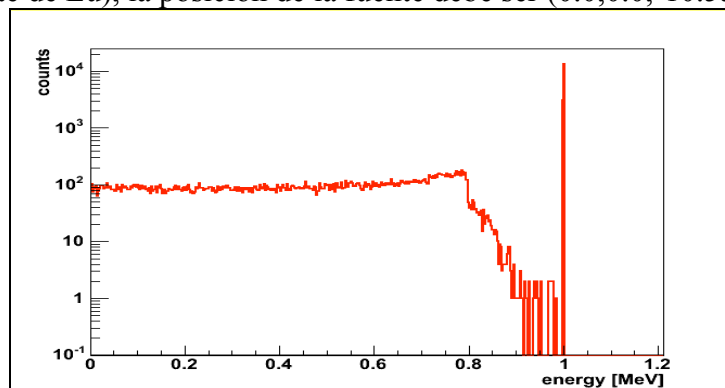
1. Entrar en el directorio *work* y configurar las variables de entorno de Geant4, ROOT y visualizadores

```
source geant4.9.6_enviroment.sh
source root_enviroment.com
source dawn_visualizer.com
source freewrl_visualizer.com
```
2. Copiar el directorio *test* al directorio *test_MC_Nombre_Apellido*

```
cp -r test test_MC_Nombre_Apellido
```
3. En el directorio *test_MC_Nombre_Apellido* familiarizarse con la estructura del programa y las subrutinas relevantes, identificando: la descripción geométrica, el generador de eventos, las partículas y procesos incluidos y la recogida de información a nivel de traza, evento y simulación.
4. Compilar el programa: *gmake*, y ejecutar el programa: *bin/Linux-g++/test*
5. Visualizar la geometría ejecutando el macro correspondiente: */control/execute vis.mac*. Cambiar el visualizador.
6. Estudiar e interpretar la información detallada de las trazas (posición, energía depositada, procesos secundarios, ...) generadas lanzando fotones, electrones y positrones de diferentes energías, usando el macro *print-track.mac* (*utilizar gammas de 1 y 5 MeV, electrones de 1 MeV y positrones de 1 MeV*).
7. Visualizar la información de las trazas usando el macro *vis.mac* modificado adecuadamente, utilizando un gamma de 5 MeV y un positrón de 1 MeV.
8. Cambiar la posición de la fuente a (0,0,10cm) en el *src/testPrimaryGeneratorAction.cc*. Obtener espectros de energía depositada usando el macro *spec-eff.mac* para fotones, electrones y positrones de diversas energías (*utilizar gammas de 1 y 5 MeV, electrones de 1 MeV y positrones de 1 MeV*). Los histogramas se visualizarán con ROOT (*root*) ejecutando el macro correspondiente: *./x hplot.cpp*. Determinar las eficiencias de detección (totales y de pico). Es importante después de cada simulación (run) copiar los ficheros *test.vec* con otro nombre, por ejemplo *test_Ge_distancia_fuente_gamma_Egamma.vec*, a un fichero que contenga la información relevante del espectro.
9. Modificar el material del detector (p.e. BGO, Si) y observar el efecto sobre el espectro de energía depositada para un gamma de 5 MeV y para electrones de 1 MeV.
10. Calcular la eficiencia del detector de Ge que se utiliza en el laboratorio. Para esto es necesario utilizar la geometria *testDetectorConstruction.cc.Ge*.

```
cp testDetectorConstruction.cc.Ge testDetectorConstruction.cc
```

Para el calculo de la eficiencia utilizar las energias 1.332 MeV (fuente de Co), 0.122 MeV(fuente de Eu), la posición de la fuente debe ser (0.0,0.0,-10.55 cm).



11. Si el tiempo lo permite, utilizar la geometria del detector simple, cambiar la posición de la fuente a (0,0,10.0 cm) e introducir un material absorbente (aluminio) entre fuente y detector y observar el efecto en el espectro (gamma de 1 MeV, electron de 1 MeV).