Práctica de Laboratorio nº 4:
Simulación Monte Carlo de detectores

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. Geant4 es un código de última generación y consiste en un paquete de librerías C++ que el usuario puede utilizar con gran libertad para construir un código de simulación específico. Esto le concede la 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 recogida de la información. Para el tratamiento de los datos obtenidos de la simulación, se hará uso del código ROOT que se presentará con un breve manual.

Bibliografía

1.-Alex Bielajew, Fundamentals of the Monte Carlo method for neutral and charged particle transport: http://www-personal.engin.umich.edu/~bielajew/MCBook/book.pdf

2.- Geant4 : http://geant4.cern.ch/

3.-Linux: http://www.debianhelp.co.uk/commands.htm

4.-ROOT: http://root.cern.ch

Materiales

El alumno dispone del siguiente material:

1. Ordenador con entorno de trabajo Linux o Windows. En el caso de Linux, los siguientes pasos se seguirán dentro de un terminal. Si tenemos Windows, debemos instalarnos el software "MobaXterm (Portable edition)". Los siguientes pasos se realizarán dentro de este software.
2. Acceder via ssh al ordenador de cálculo mediante el comando:
   ssh -XYC -p8080 master2021@161.111.23.237 (pedir password)
   En MobaXterm:
   Clicar en "Session", luego en "SSH" y poner los siguientes parámetros:
   Remote host: 161.111.23.237
   Username: master2021
   Port: 8080
3. El Directorio de trabajo */home/master2021/Geant4/* Accede a él haciendo
   cd Geant4
   En él, veréis haciendo
   ls

que se ubica un directorio llamado *IEMLOAX_Master_Template* que debes copiar usando el comando:

cp -rp IEMLOAX_Master_Template Simulacion_TuNombre

Este directorio copiado con tu nombre será en el que trabajes. Si accedes a él mediante el comando:

cd Simulacion_TuNombre

verás ejecutando el comando

ls -althr

que dentro de él existen, entre otros, un programa principal simple de simulación Monte Carlo basado en Geant4, llamado *GeLOAX.cc*, y las clases relacionadas, que se encuentran en los subdirectorios *src* e *include*. El subdirectorio *data* almacenará los ficheros de salida y el programa de análisis.

4. Macros para la ejecución del programa y variables de entorno (*vis.mac*, *env.sh*...)
5. Hoja anexa, donde se especificarán las dimensiones, geometría y materiales del detector de Germanio GEM que se debe simular.
6. ROOT para el análisis y representación de los datos.

Realización

1. Empezamos mirando el código principal: GeLOAX.cc haciendo:
   kate -n GeLOAX.cc &
2. Posteriormente, desde kate podréis abrir los ficheros de la carpeta *src* e *include* desde el menú "File" del programa kate. Allí, se puede ver la estructura del programa de Geant4 y las clases relevantes identificando: la descripción geométrica, el generador de eventos, las partículas y procesos incluidos y la recogida de información a nivel de paso, evento y simulación.
3. Una vez hemos visto todos los ficheros del código de Geant4 que vamos a utilizar, volvemos al terminal. Hay que cargar las variables de entorno de Geant4 que se encuentran en el archivo env.sh:
   source env.sh
4. A continuación compilamos el programa GeLOAX.cc: ejecutaremos
   make clean
   para asegurarnos que no haya previas compilaciones que enturbien la nuestra, luego compilaremos:
   make
5. Ejecutar el programa:
   ./bin/Linux-g++/GeLOAX
6. Visualizar la geometría indicada en el anexo ejecutando la macro de visualización:
   /control/execute vis.mac
7. Podemos rotar el objeto que estamos viendo. Para ver el interior del detector clicar en "View" y luego desactivar la opción "Lighting Calculation". Al cerrar la ventana se nos abre otra esta vez con partículas lanzadas en donde podemos ver sus trayectorias. Podemos cambiar orientación de la visualización, lanzar partículas… todo ello se define en la macro vis.mac
8. Los dos comandos anteriores se pueden ejecutar a la vez haciendo:
   ./bin/Linux-g++/GeLOAX vis.mac
9. Ejecutar la macro: ***energystore.mac***
10. Para salir de la ejecución de nuestro código de Geant4 debemos hacer: **exit**
11. La información de salida de la simulación de GEANT4 se guarda automáticamente en el directorio *data* en un fichero llamado *runOutput.dat*. Entramos en data haciendo:
    cd data

Hay que asegurarse después de cambiar el nombre a este fichero, puesto que cada vez que ejecutemos la simulación el fichero runOutput.dat se sobreescribirá con nuevos datos. Editar el fichero de datos para entender su estructura.

12. Ejecutar ***root Visualize.C*** para visualizar el espectro de energía obtenido para rayos gamma de 5.0 MeV con la simulación. Para salir de ROOT simplemente hacer: ***.q***
13. Llegados aquí, veamos una breve introducción al código de análisis y tratamiento de datos ROOT (ver y seguir pasos en Anexo II). Este se ejecuta dentro de la subcarpeta ROOT.
14. En este punto el alumno debe volver al código de Geant4, ubicado fuera del directorio ROOT (ejecutando ***cd ..*** para volver a la carpeta original).
15. Cambiar así mismo la posición de la fuente a 10 cm y la apertura a 180° para simular el experimento real que se realiza en la práctica 1 (Espectroscopía Gamma con detectores de Ge). Compilar de nuevo: ***make***
    Podemos visualizar los cambios volviendo a lanzar la macro vis.mac:
    ./bin/Linux-g++/GeLOAX vis.mac
16. Modificar las dimensiones de nuestra geometría como indican las especificaciones del detector de germanio GEM (anexo) y visualizarla.
17. Modificar y ejecutar la macro ***energystore.mac*** con esta nueva geometría, lanzando 50000 rayos gamma con cada una de las energías siguientes: 121.8 keV, 244.7 keV, 344.3 keV, 411.1 keV, 444.0 keV, 778.9 keV y 1408.0 keV (correspondientes a la desintegración del ^{152}Eu). Asegurarse de cambiar el nombre a los ficheros que se guardan (*runOutput.dat*) para poder analizarlos después.
18. Ejecutar de nuevo **root *Visualize.C*** para visualizar el espectro de salida de nuestras nuevas simulaciones y poder calcular la eficiencia absoluta de fotopico a las energías simuladas (y poder así compararlas a posteriori con las medidas experimentalmente en la práctica del detector de HPGe).
19. Para hacer la comparativa con el espectro experimental de 152Eu, hacer uso de lo aprendido en el anexo II sobre ROOT para sumar los espectros simulados para las energías anteriores y obtener un espectro de 152Eu "simulado" que podáis representar junto con el experimental y comentar las diferencias.

Informe de Prácticas

Para la evaluación de la realización-comprensión de la práctica el alumno deberá presentar un informe que enviará por correo electrónico a la dirección: jose.briz@csic.es. En el mismo se debe explicar la metodología seguida durante la práctica y se presentará, al menos, lo siguiente:

a) **Imagen de la geometría** utilizada a partir del apartado 12 incluyendo trazas de rayos gamma. Los colores de los distintos volúmenes y la orientación de la visualización serán elegidos por el alumno y no podrán ser los mismos que los definidos por defecto.

b) **Espectros**[1] obtenidos en las simulaciones en el apartado 7 para 5.0 MeV y en el apartado 12 para 344.3 keV.

c) **Tabla y gráfica**[2] **de eficiencias absolutas de detección** para todas las energías simuladas en el apartado 12. Discutir cómo se comparan con las eficiencias medidas experimentalmente en la práctica de Espectroscopia Gamma con detectores de Ge.

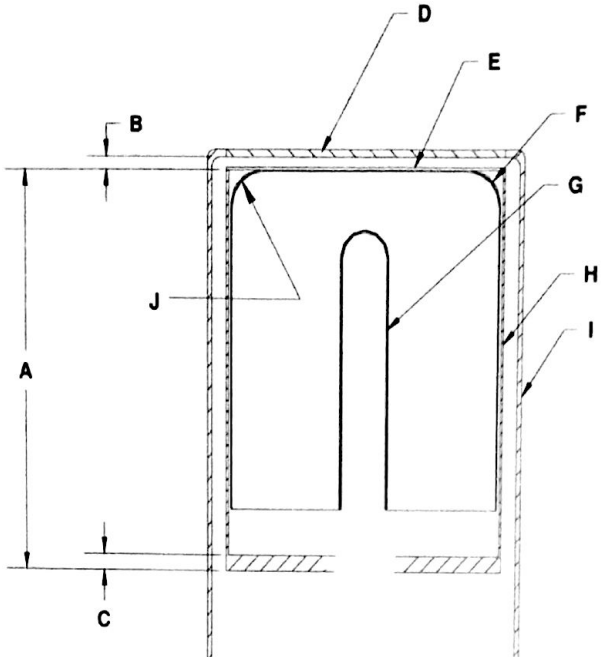
d) **Gráfica con los espectros experimental** (medido durante la práctica con el detector de HPGe con la fuente de ^{152}Eu) **y simulado** (obtenido a partir de los espectros individuales anteriormente

[1] Se recomienda representar todos los espectros incluidos en el informe en escala logarítmica en el eje Y
[2] La gráfica de eficiencias se debe representar con ambos ejes en escala logarítmica

obtenidos en simulación para cada energía y sumados[3,4]) representados sobre el mismo gráfico[1] para poder apreciar visualmente las diferencias entre simulación y experimento.

e) El alumno además contestará a las siguientes **cuestiones**:
   a. En el espectro obtenido para 5.0 MeV se observan distintas partes: un pico muy intenso, varios picos menos intensos, un continuo… ¿a qué se deben? ¿Por qué aparecen dos picos en el espectro de 5.0 MeV que no aparecen en el de 344.3 keV?
   b. Si en lugar de simular la respuesta del detector a la radiación gamma quisiéramos simular la deposición de energía de protones en éste, ¿qué ficheros deberíamos cambiar y como los cambiaríamos? (dar una repuesta cualitativa, sin incluir líneas de código).
   c. ¿Cuál es la diferencia más notable entre los espectros simulados y los que se obtienen al medir fuentes radioactivas reales con un detector de Ge real? ¿A qué se debe?

ANEXO: geometría detallada del detector de Ge real



BASIC DETECTOR DIMENSIONS	
DETECTOR DIAMETER	67 mm
DETECTOR LENGTH	73.4 mm
DETECTOR END RADIUS (J)	8 mm, NOMINAL
HOLE DIAMETER	12.5 mm
HOLE DEPTH	62.5 mm
HOLE BOTTOM RADIUS	HOLE DIAMETER / 2, NOMINAL

MISCELLANEOUS DETECTOR ASSEMBLY DIMENSIONS AND MATERIALS			
IDENTIFIER	DIMENSION	DESCRIPTION	MATERIAL(S)
A	105 mm	MOUNT CUP, LENGTH	ALUMINUM
B	4 mm	END CAP TO CRYSTAL GAP	N.A.
C	3.2 mm	MOUNT CUP BASE	ALUMINUM
D	0.8 mm	END CAP WINDOW	CARBON FIBER
E	0.03/0.03 mm/mm	INSULATOR/SHEILD	MYLAR/ALUMINIZED MYLAR
F	700 microns	OUTSIDE CONTACT LAYER	LITHIUM
G	0.3 microns	HOLE CONTACT LAYER	BORON
H	0.76 mm	MOUNT CUP WALL	ALUMINUM
I	1.5 mm	END CAP WALL	ALUMINUM

[3] Para sumar espectros en ROOT se recomienda hacer uso de la función Add() que permite sumar histogramas.
[4] Prestar atención a los factores de sumado, deben obtenerse a partir de las características de la fuente de ^{152}Eu

ANEXO II: Breve introducción al código ROOT

Encontraréis una subcarpeta llamada ROOT. Podéis acceder a ella haciendo:

cd ROOT

En ella hay varios ficheros pero de momento nos interesan estos:

- *60Co.dat*: espectro experimental tomado con el detector de HPGe de la práctica 1 con una fuente radioactiva de ^{60}Co.
- *macro.cpp*: macro de ROOT con instrucciones para leer el fichero 60Co.dat, convertirlo en un histograma de ROOT y guardarlo en un fichero de ROOT de salida que se llama *output.root*
- *inputlist*: fichero que contiene la lista de ficheros que la macro macro.cpp va a leer y convertir a ROOT.

Si abrís el fichero inputlist con un editor de textos:

kate -n inputlist &

veréis que contiene el nombre del fichero 60Co.dat que vamos a convertir. Ver además el código de la macro que vamos a ejecutar: macro.cpp que podéis abrir desde el menú "File" del programa kate.
Para ejecutar la macro de ROOT simplemente hay que hacer en el terminal:

root -l macro.cpp -q

Una vez ejecutado se tiene que haber creado el fichero output.root. Para abrirlo y ver lo que contiene:

root -l output.root

Una vez dentro de ROOT, hacer:

new TBrowser

y se abrirá una ventana en donde aparecerá en la parte izquierda una lista de ficheros. Allí debe aparecer el fichero *output.root* que habéis abierto. Si hacéis doble click sobre ese fichero se abrirá y mostrará lo que contiene. En este caso 2 histogramas: h1 y hcal1.

- El primero, h1, se puede ver haciendo doble click en él. Se dibujará en el panel derecho, llamado Canvas. Se trata del histograma original, sin calibrar.
- El otro espectro, llamado hcal1, es el mismo espectro pero ya calibrado en energía (atención, está generado con una calibración que no es la correcta, como veremos, hay que recalcular los parámetros y cambiarlos en el código macro.cpp).

Para ver más detalle de los picos, podéis hacer **ZOOM** en los espectros clicando justo debajo del eje X y dejando pulsado arrastrar para marcar la región del espectro sobre la que queremos hacer ZOOM. Otra forma de hacer ZOOM es clicando sobre el eje con el botón derecho del ratón y marcar la función "SetRangeUser" Y luego le damos los valores mínimo y máximo del eje sobre los que queremos hacer ZOOM.

Ajuste de picos a función gaussiana

Para analizar los picos, normalmente se hacen ajustes a funciones gaussianas para conocer las características del pico (posición, área, anchura). Para hacer este ajuste debemos hacer lo siguiente:

1.- Cargar el fichero *RootFit.C* haciendo sobre el terminal o en el campo "Command (local)" de la ventana del TBrowser situado debajo del histograma representado:

.L RootFit.C

2.- Para hacer el ajuste, hay que ejecutar:

FitSinglePeak(h1,min,max)

En donde h1 es el nombre del histograma que queremos ajustar, y min y max son los valores mínimo y máximo del eje X en donde queremos hacer el ajuste (hay que elegir valores cercanos al pico para que el ajuste se haga correctamente).

3.- Una vez ejecutado debe aparecer el histograma con un ajuste a una gaussiana más fondo lineal con los valores de los parámetros indicados sobre el cuadro situado a la derecha arriba dentro del Canvas donde estaba representado el histograma:

Sigma (parámetro de la gaussiana que caracteriza su anchura): FWHM = 2.355* sigma

Area: Área entre la curva azul y la recta verde, que es lo que se considera fondo

Position: Posición central de la gaussiana

4.- Debemos hacer los ajustes de los dos picos del espectro correspondientes a 60Co y anotar los parámetros del ajuste encontrados.

Podemos salir de ROOT simplemente ejecutando ***.q*** en el terminal o en el campo "Command (local)".

Calibración de espectros

En la carpeta ROOT, hay además otra macro llamada calib.cpp que lee el fichero calib.dat. Se trata de una macro que se puede utilizar para obtener los parámetros de una calibración lineal. Lee los datos del fichero calib.dat, en donde hay dos columnas, en la primera se indica la energía y en la segunda la posición en canales de cada pico. En este ejemplo están indicados los valores para los picos de la fuente de ^{60}Co, modificadlos por los valores que habéis obtenido en vuestros ajustes. Para ejecutar la macro, simplemente hacemos:

root -l calib.cpp

Y nos aparecerá un Canvas con dos puntos en rojo y una línea recta que es el ajuste a dichos puntos. Los parámetros p0 y p1 aparecen en el cuadro de datos en la esquina superior-derecha. Con estos valores, se pueden generar los espectros calibrados. Para ello habría que modificar los valores dados en la macro "macro.cpp" por estos nuevos valores. Para abrir la macro y ver su código hacemos:

kate -n macro.cpp &

Vemos en las líneas 12 y 13 que se definen los valores de los parámetros p0 y p1. Cambiamos esos valores por los nuevos.

Ahora debemos volver a lanzar la macro macro.cpp para generar los espectros calibrados con la nueva calibración:

root -l macro.cpp -q

Si abrimos otra vez los espectros de salida:

root -l output.root

new TBrowser

podremos ver que el espectro hcal1 aparece calibrado y con los valores que esperamos para los picos de la fuente de ^{60}Co (1173.2 y 1332.5 keV).

A continuación, mencionar que en el cuadro superior-derecho del gráfico aparecen el nombre del histograma (el primero que dibujamos), el número de Entries y, muy importante, la integral del espectro en el rango que estamos visualizando. Si vamos haciendo ZOOM en diferentes regiones veremos que este valor va cambiando, como es de esperar. Otra forma de obtener la integral de un espectro es haciendo en el terminal o en el campo "Command (local)" lo siguiente:

h1->Integral(400,500);

En el caso en que quisiéramos integrar el espectro h1 desde el canal 400 al 500.

Los espectros se pueden "dibujar" también en el terminal sin entrar en TBrowser, para ello hay que hacer:

h1->Draw();

si quisiéramos representar el histograma h1. Ahora si quisiéramos representar otro histograma, por ejemplo llamado h2, sobre el mismo gráfico lo que deberíamos hacer es primero cambiar el color del segundo histograma para que se distingan los dos, haciendo:

hcal1->SetLineColor(kRed);

si queremos ponerlo rojo, o kBlue azul o kGreen verde, por ejemplo. Para más detalles, simplemente podéis acudir a la web de ROOT (CERN) o buscando en google. Y luego lo dibujamos sobre el mismo gráfico haciendo:

hcal1->Draw("same");

Por último, un histograma se puede sumar a otro haciendo (en este caso no tiene sentido hacerlo con h1 y hcal1 porque uno está calibrado y otro no):

h1->Add(h2,f);

En ese caso, el histograma h1 lo estaríamos sumando con h2 multiplicado por un factor f, que podemos elegir el valor que queramos, es decir, estamos haciendo: h1=h1+f*h2. Nota: f puede ser negativo. Si queremos ver el histograma h1 tras esta operación suma, debemos volver a dibujarlo (h1->Draw();)

Para hacer una copia del histograma original debéis saber que se pueden clonar los espectros:

TH1F* hcopy = (TH1F*) h1->Clone("hcopy");

Y podéis reescalar los espectros (multiplicar por un factor), haciendo:

hcopy->Scale(f);

siendo f el factor por el que multiplicamos el espectro original.

De esta forma hemos aprendido a convertir ficheros de texto (en formato ASCII) al formato de histogramas de ROOT, analizar picos del espectro mediante ajustes gaussianos, hacer un ajuste de calibración, generar los histogramas calibrados, representarlos, integrarlos en ciertas regiones y sumarlos o restarlos.