Simulaciones de centelladores PET con LITRANI

Adriana Martín de Aguilera Moreno
Trabajo Académicamente Dirigido por José Manuel Udías Moinelo y Samuel España Palomares

Grupo de Física Nuclear
Universidad Complutense de Madrid

26 de septiembre de 2008

Qué es la Tomografía por Emisión de Positrones

- Es una técnica de imágen diagnóstica *in vivo*.
- Utiliza radifármacos marcados con isótopos radiactivos de vida media corta que se desintegran vía β^+.
- Los positrones en la materia se encuentran y recombinan rápidamente con los electrones de ésta, dando lugar a dos fotones gamma de 511 KeV que siguen trayectorias opuestas.
- Podemos estudiar la llegada de estos fotones a un anillo de detectores...
- ... y analizar la señal en busca de fotones detectados en puntos diametralmente opuestos del anillo en una ventana reducida de tiempo.

Qué es la Tomografía por Emisión de Positrones

- Es una técnica de imágen diagnóstica *in vivo*.
- Utiliza radifármacos marcados con isótopos radiactivos de vida media corta que se desintegran vía β^+.
- Los positrones en la materia se encuentran y recombinan rápidamente con los electrones de ésta, dando lugar a dos fotones gamma de 511 KeV que siguen trayectorias opuestas.
- Podemos estudiar la llegada de estos fotones a un anillo de detectores...
- ... y analizar la señal en busca de fotones detectados en puntos diametralmente opuestos del anillo en una ventana reducida de tiempo.

Qué es la Tomografía por Emisión de Positrones

- Es una técnica de imágen diagnóstica *in vivo*.
- Utiliza radifármacos marcados con isótopos radiactivos de vida media corta que se desintegran vía β^+.
- Los positrones en la materia se encuentran y recombinan rápidamente con los electrones de ésta, dando lugar a dos fotones gamma de 511 KeV que siguen trayectorias opuestas.
- Podemos estudiar la llegada de estos fotones a un anillo de detectores...
- ... y analizar la señal en busca de fotones detectados en puntos diametralmente opuestos del anillo en una ventana reducida de tiempo.

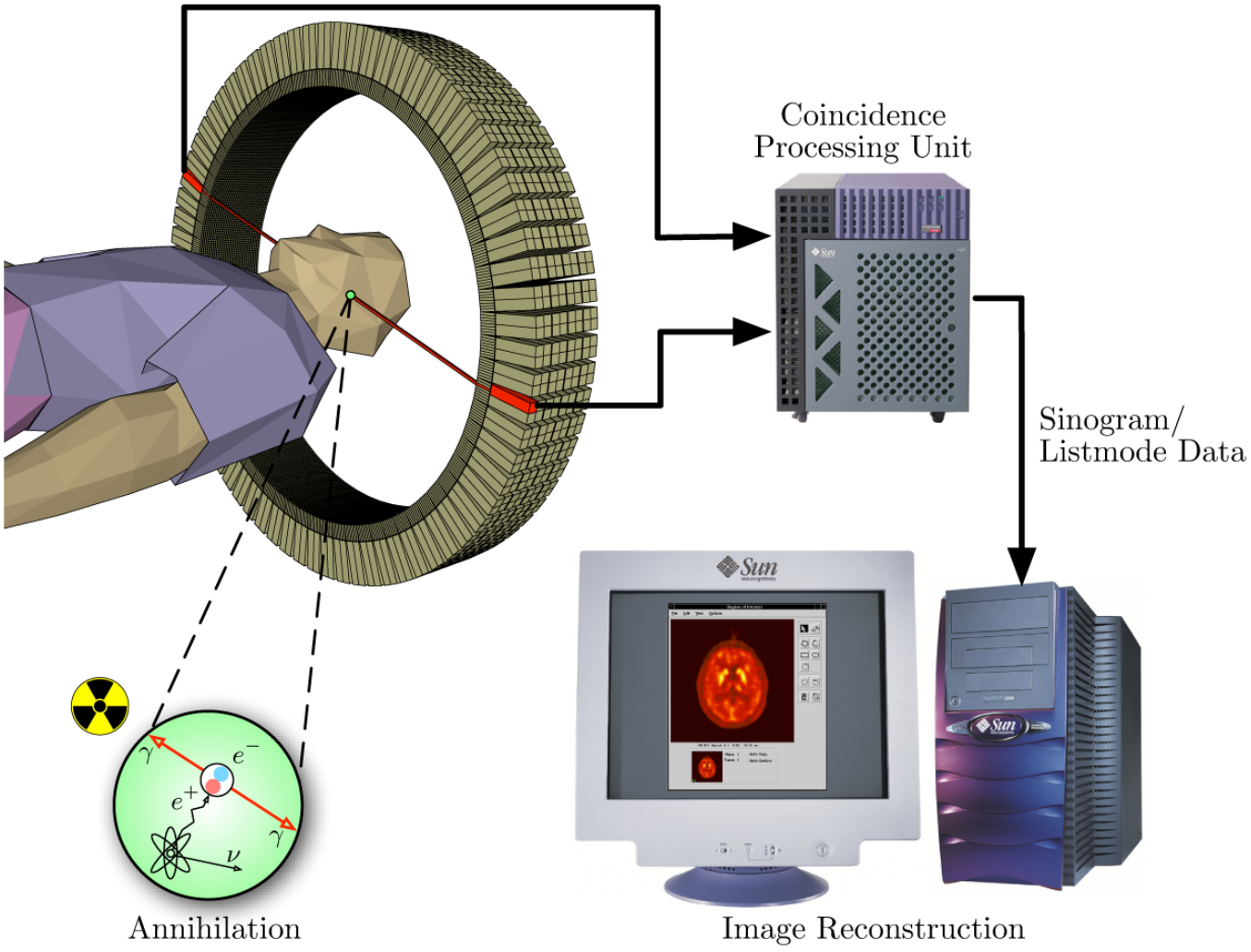
Qué es la Tomografía por Emisión de Positrones

- Es una técnica de imágen diagnóstica *in vivo*.
- Utiliza radifármacos marcados con isótopos radiactivos de vida media corta que se desintegran vía β^+.
- Los positrones en la materia se encuentran y recombinan rápidamente con los electrones de ésta, dando lugar a dos fotones gamma de 511 KeV que siguen trayectorias opuestas.
- Podemos estudiar la llegada de estos fotones a un anillo de detectores...
- ... y analizar la señal en busca de fotones detectados en puntos diametralmente opuestos del anillo en una ventana reducida de tiempo.

Qué es la Tomografía por Emisión de Positrones

- Es una técnica de imágen diagnóstica *in vivo*.
- Utiliza radifármacos marcados con isótopos radiactivos de vida media corta que se desintegran vía β^+.
- Los positrones en la materia se encuentran y recombinan rápidamente con los electrones de ésta, dando lugar a dos fotones gamma de 511 KeV que siguen trayectorias opuestas.
- Podemos estudiar la llegada de estos fotones a un anillo de detectores...
- ... y analizar la señal en busca de fotones detectados en puntos diametralmente opuestos del anillo en una ventana reducida de tiempo.

Representación esquemática de un aparato de PET para humanos



Cristales centelladores inorgánicos

Poseen una propiedad llamada **luminiscencia**. Los materiales luminiscentes absorben la energía a la que estén expuestos (luz, calor...) y la reemiten como fotones del espectro visible.

Características de estos materiales

- Convierten la energía cinética de las partículas cargadas en luz detectable.
- Esta conversión debe ser lineal.
- El medio ha de ser transparente a su propia longitud de emisión.
- La vida media de la luminiscencia inducida debe ser corto.
- Debe poder fabricarse con facilidad.
- Su índice de refracción debe ser parecido al del vidrio.

Cristales centelladores inorgánicos

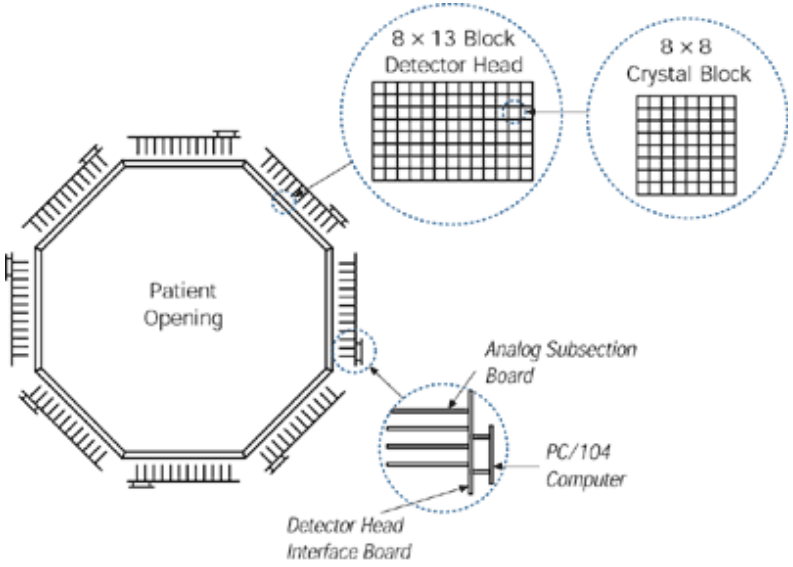
Poseen una propiedad llamada **luminiscencia**. Los materiales luminiscentes absorben la energía a la que estén expuestos (luz, calor...) y la reemiten como fotones del espectro visible.

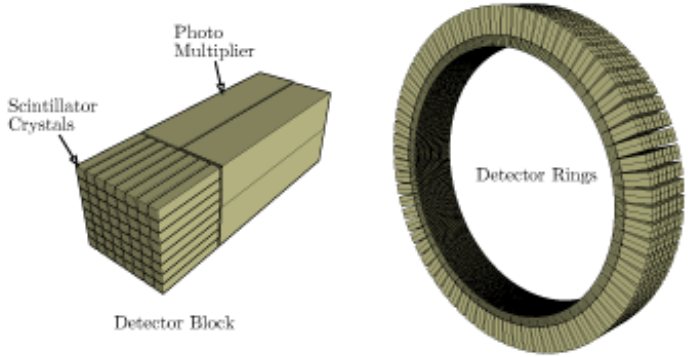
Características de estos materiales

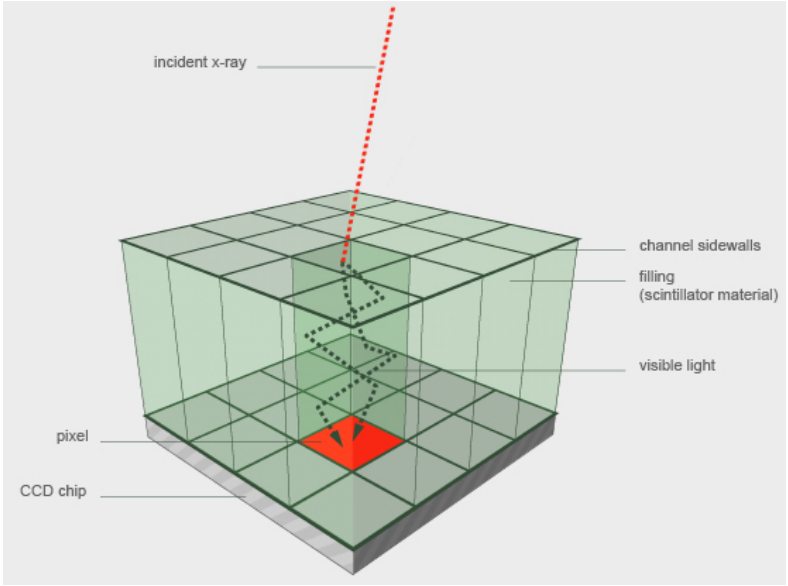
- Convierten la energía cinética de las partículas cargadas en luz detectable.
- Esta conversión debe ser lineal.
- El medio ha de ser transparente a su propia longitud de emisión.
- La vida media de la luminiscencia inducida debe ser corto.
- Debe poder fabricarse con facilidad.
- Su índice de refracción debe ser parecido al del vidrio.

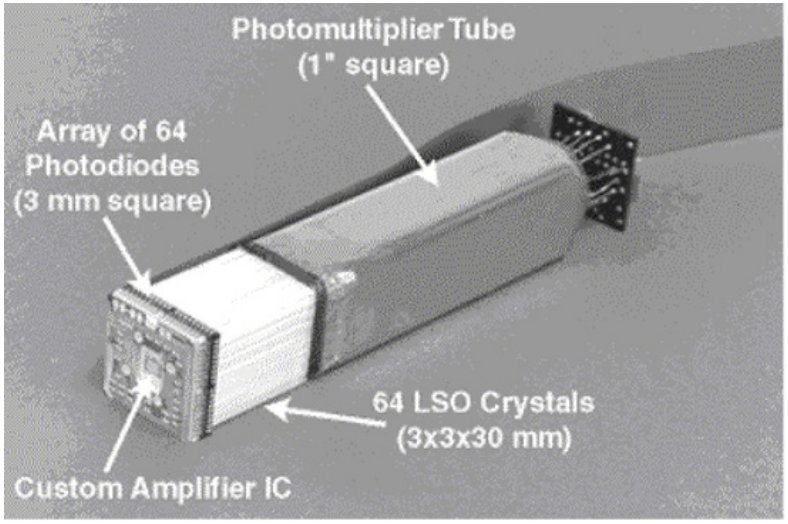
Distribución de los detectores

Según el aparato, podemos encontrarnos con soportes circulares, cuadrados o poligonales para los detectores de radiación. En todos los casos cada detector estará formado por varios bloques de cristales. Puesto que detectar los fotones y situar su origen son dos puntos fundamentales para obtener una buena imagen, el funcionamiento de los detectores debe ser óptimo.







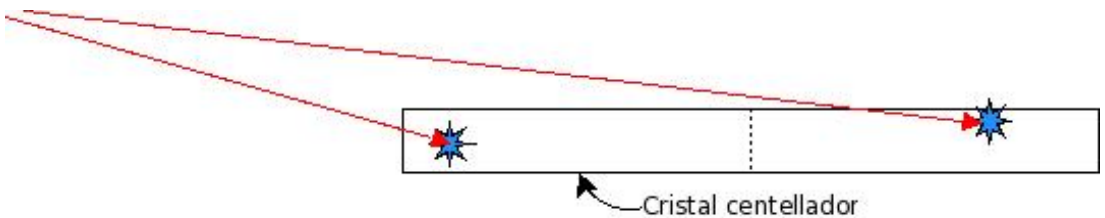


Matrices de centelladores

Los cristales están separados entre si por algún recubrimiento reflector, de modo que cada cristal será un píxel y estará en contacto con el fotodetector. Puesto que todos los cristales de cada matriz van a parar al mismo fototubo, no nos es posible saber a ciencia cierta en qué cristal se ha producido la llegada del fotón.

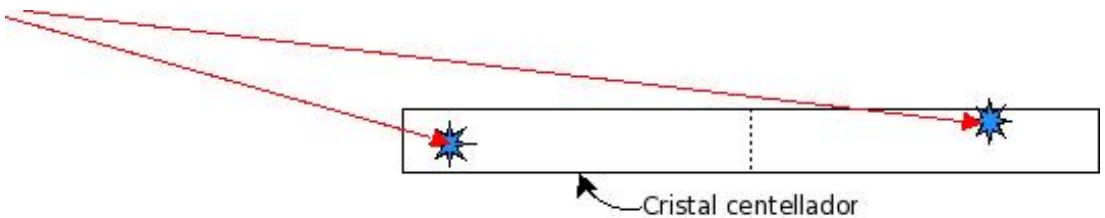
Qué problema queremos resolver

- Queremos incrementar la precisión a la hora de situar la aniquilación del par electrón-positrón a partir de la información de los fotomultiplicadores.
- Hay muchas formas de determinar el cristal al que ha llegado el fotón.
- Saber si el fotón que detectamos ha llegado a la parte delantera o a la trasera del cristal nos ayudaría a calcular con más precisión la trayectoria y el origen de cada fotón.



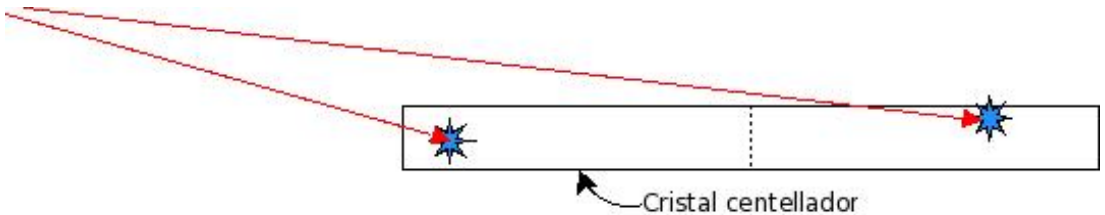
Qué problema queremos resolver

- Queremos incrementar la precisión a la hora de situar la aniquilación del par electrón-positrón a partir de la información de los fotomultiplicadores.
- Hay muchas formas de determinar el cristal al que ha llegado el fotón.
- Saber si el fotón que detectamos ha llegado a la parte delantera o a la trasera del cristal nos ayudaría a calcular con más precisión la trayectoria y el origen de cada fotón.



Qué problema queremos resolver

- Queremos incrementar la precisión a la hora de situar la aniquilación del par electrón-positrón a partir de la información de los fotomultiplicadores.
- Hay muchas formas de determinar el cristal al que ha llegado el fotón.
- Saber si el fotón que detectamos ha llegado a la parte delantera o a la trasera del cristal nos ayudaría a calcular con más precisión la trayectoria y el origen de cada fotón.



Idea

Los fotones que, aún habiéndose originado uno cerca del otro pero no en el mismo lugar, pueden llegar regiones distintas del mismo cristal. Saber si el fotón que detectamos ha llegado a la parte delantera o a la trasera del cristal nos ayudaría a calcular con más precisión la trayectoria y el origen de cada fotón.

Qué vamos a hacer

Muchas matrices para detectores de PET, como la de la figura 1.8, constan de dos bloques de cristales centelladores unidos mediante un gel o una grasa óptica, además del recubrimiento reflector que los separa. Nos interesa estudiar este sistema para saber si el fototubo apreciará alguna diferencia en la cantidad de fotones vistos si generamos fotones en el primer o el segundo cristal.

Idea

Los fotones que, aún habiéndose originado uno cerca del otro pero no en el mismo lugar, pueden llegar regiones distintas del mismo cristal. Saber si el fotón que detectamos ha llegado a la parte delantera o a la trasera del cristal nos ayudaría a calcular con más precisión la trayectoria y el origen de cada fotón.

Qué vamos a hacer

Muchas matrices para detectores de PET, como la de la figura 1.8, constan de dos bloques de cristales centelladores unidos mediante un gel o una grasa óptica, además del recubrimiento reflector que los separa. Nos interesa estudiar este sistema para saber si el fototubo apreciará alguna diferencia en la cantidad de fotones vistos si generamos fotones en el primer o el segundo cristal.

Introducción a LITRANI

¿Qué es LITRANI?

Las siglas LITRANI proceden de *LIght TRansmission in ANIsotropic media*. LITRANI es un paquete de simulación Monte Carlo construido sobre ROOT que simula el paso de luz a través de un montaje definido por formas y superficies previamente construidas en ROOT. Cada una de estas formas puede estar constituida por un material diferente, cuyas propiedades ópticas pueden ser constantes o depender de la longitud de onda. Además, permite simular materiales cuyo tensor dieléctrico es anisótropo además de los recubrimientos que los envuelven.

Ventajas

- Descripción detallada de las superficies: rugosidad, recubrimientos, huecos, etc.
- Es fácil de ampliar usando macros de ROOT.
- Incluye varios modelos de fuentes y detectores de fotones comerciales, pudiéndose incluir nuevos modelos fácilmente.
- Es válido en medios isótropos y anisótropos.

Ventajas

- Descripción detallada de las superficies: rugosidad, recubrimientos, huecos, etc.
- Es fácil de ampliar usando macros de ROOT.
- Incluye varios modelos de fuentes y detectores de fotones comerciales, pudiéndose incluir nuevos modelos fácilmente.
- Es válido en medios isótropos y anisótropos.

Ventajas

- Descripción detallada de las superficies: rugosidad, recubrimientos, huecos, etc.
- Es fácil de ampliar usando macros de ROOT.
- Incluye varios modelos de fuentes y detectores de fotones comerciales, pudiéndose incluir nuevos modelos fácilmente.
- Es válido en medios isótropos y anisótropos.

Ventajas

- Descripción detallada de las superficies: rugosidad, recubrimientos, huecos, etc.
- Es fácil de ampliar usando macros de ROOT.
- Incluye varios modelos de fuentes y detectores de fotones comerciales, pudiéndose incluir nuevos modelos fácilmente.
- Es válido en medios isótropos y anisótropos.

Inconvenientes

- Sólo se dispone de formas simples (no pueden extenderse). No se pueden situar unas formas dentro de otras.
- Sólo montajes fáciles (GEANT simula mejor volúmenes complejos).
- Tiene dificultades con las superficies curvas.
- El modelo es incorrecto para la propagación en el tiempo, pues sólo tiene en cuenta la velocidad de fase.
- Los datos necesarios para definir un material no son los que se utilizan normalmente en la vida real y es muy difícil encontrarlos.

Inconvenientes

- Sólo se dispone de formas simples (no pueden extenderse). No se pueden situar unas formas dentro de otras.
- Sólo montajes fáciles (GEANT simula mejor volúmenes complejos).
- Tiene dificultades con las superficies curvas.
- El modelo es incorrecto para la propagación en el tiempo, pues sólo tiene en cuenta la velocidad de fase.
- Los datos necesarios para definir un material no son los que se utilizan normalmente en la vida real y es muy difícil encontrarlos.

Inconvenientes

- Sólo se dispone de formas simples (no pueden extenderse). No se pueden situar unas formas dentro de otras.
- Sólo montajes fáciles (GEANT simula mejor volúmenes complejos).
- Tiene dificultades con las superficies curvas.
- El modelo es incorrecto para la propagación en el tiempo, pues sólo tiene en cuenta la velocidad de fase.
- Los datos necesarios para definir un material no son los que se utilizan normalmente en la vida real y es muy difícil encontrarlos.

Inconvenientes

- Sólo se dispone de formas simples (no pueden extenderse). No se pueden situar unas formas dentro de otras.
- Sólo montajes fáciles (GEANT simula mejor volúmenes complejos).
- Tiene dificultades con las superficies curvas.
- El modelo es incorrecto para la propagación en el tiempo, pues sólo tiene en cuenta la velocidad de fase.
- Los datos necesarios para definir un material no son los que se utilizan normalmente en la vida real y es muy difícil encontrarlos.

Inconvenientes

- Sólo se dispone de formas simples (no pueden extenderse). No se pueden situar unas formas dentro de otras.
- Sólo montajes fáciles (GEANT simula mejor volúmenes complejos).
- Tiene dificultades con las superficies curvas.
- El modelo es incorrecto para la propagación en el tiempo, pues sólo tiene en cuenta la velocidad de fase.
- Los datos necesarios para definir un material no son los que se utilizan normalmente en la vida real y es muy difícil encontrarlos.

Elementos que componen LITRANI

- **LITRANI (TwoPad):** haciendo uso de las librerías y elementos que ahora describiremos, permite crear el montaje que deseamos simular.
- **SplineFit:** módulo de interpolación de datos cuyo fin es, empleando datos dados por el usuario, realizar interpolaciones para su uso en otra simulación de TwoPad. Suele ser muy importante para introducir índices de refracción de materiales, permeabilidades o secciones eficaces en montajes con fuentes de luz no monocromáticas.
- **VisuLitrani:** encargado de mostrar histogramas y resultados producto de las simulaciones. El número y carácter de dichos resultados puede modificarse al escribir el programa.

Elementos que componen LITRANI

- **LITRANI (TwoPad):** haciendo uso de las librerías y elementos que ahora describiremos, permite crear el montaje que deseamos simular.
- **SplineFit:** módulo de interpolación de datos cuyo fin es, empleando datos dados por el usuario, realizar interpolaciones para su uso en otra simulación de TwoPad. Suele ser muy importante para introducir índices de refracción de materiales, permeabilidades o secciones eficaces en montajes con fuentes de luz no monocromáticas.
- **VisuLitrani:** encargado de mostrar histogramas y resultados producto de las simulaciones. El número y carácter de dichos resultados puede modificarse al escribir el programa.

Elementos que componen LITRANI

- **LITRANI (TwoPad):** haciendo uso de las librerías y elementos que ahora describiremos, permite crear el montaje que deseamos simular.
- **SplineFit:** módulo de interpolación de datos cuyo fin es, empleando datos dados por el usuario, realizar interpolaciones para su uso en otra simulación de TwoPad. Suele ser muy importante para introducir índices de refracción de materiales, permeabilidades o secciones eficaces en montajes con fuentes de luz no monocromáticas.
- **VisuLitrani:** encargado de mostrar histogramas y resultados producto de las simulaciones. El número y carácter de dichos resultados puede modificarse al escribir el programa.

Formas que podemos utilizar

- TSBRIK: caja de caras perpendiculares entre sí.
- TSTRD1: caja trapezoidal con la dimensión x variable a lo largo de z.
- TSTRD2: caja trapezoidal con las dimensiones x e y variando a lo largo de z.
- TSPARA: paralelepípedo.
- TSTRAP: trapezoide general.
- TS8PTS: volumen general que se determina mediante sus ocho vértices.
- TSCONE: cono.
- TSCYL: cilindro.
- TSTUBE: tubo con hueco para introducir un cilindro en su interior.

Fuentes de luz disponibles

- Espontánea: emite fotones desde cualquier punto de un volumen o una superficie.
- Fibra óptica: los fotones proceden de una fibra óptica en el interior del volumen.
- Haz de partículas: genera haces de, por ejemplo, muones, capaces de producir luz Cherenkov en el intetior del material.
- Gammas: fotones de entre 0,1 y 1 MeV, teniendo en cuenta el efecto Compton y el efecto fotoeléctrico en el interior del material.
- Cascadas electromagnéticas.

Detectores disponibles

- Detectores de superficie.
- Detectores de volumen.
- Phototubos.
- APD's.
- Diodos PIN.

Recubrimientos

Es posible añadir recubrimientos, es decir, simular la presencia de materiales metálicos adheridos a las caras de los volúmenes. La simulación de recubrimientos es uno de los puntos más dinámicos de LITRANI, puesto que pueden simularse distintas bandas de material sobre una cara o zonas recubiertas con formas geométricas como círculos de metal. Además, es posible añadir caras erosionadas a las superficies.

Detectores disponibles

- Detectores de superficie.
- Detectores de volumen.
- Phototubos.
- APD's.
- Diodos PIN.

Recubrimientos

Es posible añadir recubrimientos, es decir, simular la presencia de materiales metálicos adheridos a las caras de los volúmenes. La simulación de recubrimientos es uno de los puntos más dinámicos de LITRANI, puesto que pueden simularse distintas bandas de material sobre una cara o zonas recubiertas con formas geométricas como círculos de metal. Además, es posible añadir caras erosionadas a las superficies.

Cómo escribir un programa en LITRANI

Inicio del programa

En primer lugar, cargamos LITRANI y especificamos el título y características del programa. También necesitamos llamar a ROOT y establecer algunas de las características de la generación de números aleatorios.

Código

```
void ejemplo1a(void)

{
  char *name    = "EJEMPLO 1a. Un cristal de 1cm^3 de  LSO recubierto
de material totalmente reflector";
  char *listing = "La cara ultima del cristal es un fotocatodo ";
  char *upcom   = "La fuente es puntual, isotropica y espontanea, y la
situo dentro del cristal ";

  char *downcom = "¿Cuantos fotones detecto?";
gROOT->ProcessLine(".x InitLitrani.C(8,name,listing,upcom,downcom,kTRUE,kFALSE
,kTRUE)");
```

Cómo escribir un programa en LITRANI

Inicio del programa

En primer lugar, cargamos LITRANI y especificamos el título y características del programa. También necesitamos llamar a ROOT y establecer algunas de las características de la generación de números aleatorios.

Código

```
void ejemplo1a(void)

{
  char *name    = "EJEMPLO 1a. Un cristal de 1cm^3 de  LSO recubierto
de material totalmente reflector";
  char *listing = "La cara ultima del cristal es un fotocatodo ";
  char *upcom   = "La fuente es puntual, isotropica y espontanea, y la
situo dentro del cristal ";

  char *downcom = "¿Cuantos fotones detecto?";
gROOT->ProcessLine(".x InitLitrani.C(8,name,listing,upcom,downcom,kTRUE,kFALSE
,kTRUE)");
```

Geometría

Definimos las semilongitudes de cada uno de los cuerpos.

Código

```
const Double_t semilongitud_en_x = 0.5;
const Double_t semilongitud_en_y = 0.5;
const Double_t semilongitud_en_z = 0.5;
```

Definición de los materiales

1. Nombre.
2. Título.
3. Sensibilidad: kTRUE si es detector y kFALSE si no lo es.
4. Permeabilidad magnética.

Código

```
TOpticMaterial *LSO;
LSO = new TOpticMaterial("LSO","Titulito,¡es LSO!",kFALSE,1.0,21);
LSO->IsIsotropic(1.82);
```

Geometría

Definimos las semilongitudes de cada uno de los cuerpos.

Código

```
const Double_t semilongitud_en_x = 0.5;
const Double_t semilongitud_en_y = 0.5;
const Double_t semilongitud_en_z = 0.5;
```

Definición de los materiales

1. Nombre.
2. Título.
3. Sensibilidad: kTRUE si es detector y kFALSE si no lo es.
4. Permeabilidad magnética.

Código

```
TOpticMaterial *LSO;
LSO = new TOpticMaterial("LSO","Titulito,¡es LSO!",kFALSE,1.0,21);
LSO->IsIsotropic(1.82);
```

Geometría

Definimos las semilongitudes de cada uno de los cuerpos.

Código

```
const Double_t semilongitud_en_x = 0.5;
const Double_t semilongitud_en_y = 0.5;
const Double_t semilongitud_en_z = 0.5;
```

Definición de los materiales

1. Nombre.
2. Título.
3. Sensibilidad: kTRUE si es detector y kFALSE si no lo es.
4. Permeabilidad magnética.

Código

```
TOpticMaterial *LSO;
LSO = new TOpticMaterial("LSO","Titulito,¡es LSO!",kFALSE,1.0,21);
LSO->IsIsotropic(1.82);
```

Geometría

Definimos las semilongitudes de cada uno de los cuerpos.

Código

```
const Double_t semilongitud_en_x = 0.5;
const Double_t semilongitud_en_y = 0.5;
const Double_t semilongitud_en_z = 0.5;
```

Definición de los materiales

1. Nombre.
2. Título.
3. Sensibilidad: kTRUE si es detector y kFALSE si no lo es.
4. Permeabilidad magnética.

Código

```
TOpticMaterial *LSO;
LSO = new TOpticMaterial("LSO","Titulito,¡es LSO!",kFALSE,1.0,21);
LSO->IsIsotropic(1.82);
```

Definición de los recubrimientos

1. Nombre.
2. Título.
3. Declaramos si hay una capa fina de algún material entre el cristal y el recubrimiento.
4. Proporción de fotones que se difunden en vez de reflejarse.
5. Parte real del índice de refracción.
6. Parte imaginaria del índice de refracción.
7. Permeabilidad magnética.
8. Absorción extra.
9. Ángulo máximo de difusión de los fotones.

Código

```
        TRevetment *reflectortotal;
reflectortotal = new TRevetment("reflectortotal","Recubrimiento reflector total",
"none",0.00, 0.0,-1.0,1.0,0.0,90.0);
```

Definición de los recubrimientos

1. Nombre.
2. Título.
3. Declaramos si hay una capa fina de algún material entre el cristal y el recubrimiento.
4. Proporción de fotones que se difunden en vez de reflejarse.
5. Parte real del índice de refracción.
6. Parte imaginaria del índice de refracción.
7. Permeabilidad magnética.
8. Absorción extra.
9. Ángulo máximo de difusión de los fotones.

Código

```
        TRevetment *reflectortotal;
reflectortotal = new TRevetment("reflectortotal","Recubrimiento reflector total",
"none",0.00, 0.0,-1.0,1.0,0.0,90.0);
```

Definiendo las formas

1. Nombre.
2. Título.
3. Material.
4. Revestimiento.
5. Semilongitudes de los lados (x, y, z).

Código

```
TSBRIK *prisma;
prisma = new TSBRIK("prisma","Cristal de LSO","LSO","reflectortotal",
semilongitud_en_x,semilongitud_en_y,semilongitud_en_z);
```

Definiendo las formas

1. Nombre.
2. Título.
3. Material.
4. Revestimiento.
5. Semilongitudes de los lados (x, y, z).

Código

```
TSBRIK *prisma;
prisma = new TSBRIK("prisma","Cristal de LSO","LSO","reflectortotal",
semilongitud_en_x,semilongitud_en_y,semilongitud_en_z);
```

Asignar propiedades a caras concretas

Muchas veces necesitamos declarar las propiedades de alguna cara en particular como, por ejemplo, rugosidades (`SetDepolished()`), biselados (`SetBevellings()`) o si actúa como detector, pudiendo definir detectores de superficie y fototubos (estos últimos solamente para formas cilíndricas).

1. Nombre.
2. Título.
3. Número de la cara que queremos establecer como detector.
4. Eficiencia cuántica: si es total basta con poner `"none"`.

Código

```
prisma->fSuppl->SetSurfDet("detector","Detector de superficie",1,"none");
```

Asignar propiedades a caras concretas

Muchas veces necesitamos declarar las propiedades de alguna cara en particular como, por ejemplo, rugosidades (`SetDepolished()`), biselados (`SetBevellings()`) o si actúa como detector, pudiendo definir detectores de superficie y fototubos (estos últimos solamente para formas cilíndricas).

1. Nombre.
2. Título.
3. Número de la cara que queremos establecer como detector.
4. Eficiencia cuántica: si es total basta con poner `"none"`.

Código

```
prisma->fSuppl->SetSurfDet("detector","Detector de superficie",1,"none");
```

Organización de los elementos del montaje

En este caso hemos definido `prisma1` como centro del montaje, en torno al cual colocaremos los demás cuerpos. Los datos que debemos introducir en este caso son:

1. Nombre.
2. Título.
3. Nombre del volumen que vayamos a colocar.
4. Distancia entre el centro del cuerpo en cuestión y el cuerpo que hemos definido como origen del montaje.

Código

```
TSNode *nodo1;
    nodo1 = new TSNode("nodo1","Nodo asociado al cristal 1",prisma1);
    nodo1->cd();

    TSNode *nodo2;
    nodo2 = new TSNode("nodo2","Nodo asociado al cristal 2",prisma2,0.5,0.0,0.0 );
```

Organización de los elementos del montaje

En este caso hemos definido `prisma1` como centro del montaje, en torno al cual colocaremos los demás cuerpos. Los datos que debemos introducir en este caso son:

1. Nombre.
2. Título.
3. Nombre del volumen que vayamos a colocar.
4. Distancia entre el centro del cuerpo en cuestión y el cuerpo que hemos definido como origen del montaje.

Código

```
TSNode *nodo1;
    nodo1 = new TSNode("nodo1","Nodo asociado al cristal 1",prisma1);
    nodo1->cd();

    TSNode *nodo2;
    nodo2 = new TSNode("nodo2","Nodo asociado al cristal 2",prisma2,0.5,0.0,0.0 );
```

Contacto entre los volumenes

El penúltimo argumento indica qué tipo de contacto es. El último parámetro nos dice si existe una capa fina de material entre las caras. Es importante tener en cuenta que:

1. ThinSlice no utiliza las ecuaciones de Fresnel.
2. Es mejor, si tenemos 3 elementos en contacto, poner como nodo de referencia el elemento central.
3. Al poner las caras rugosas, si hay recubrimiento, hay que añadir una capa fina de aire.
4. Al poner los cristales en contacto el revestimiento se elimina en las caras en contacto.

Código

```
TContact *contacto;
contacto = new TContact("entre_cristales","Contacto entre cristales",
"prisma1", "prisma2",identical,"none");
```

Contacto entre los volumenes

El penúltimo argumento indica qué tipo de contacto es. El último parámetro nos dice si existe una capa fina de material entre las caras. Es importante tener en cuenta que:

1. ThinSlice no utiliza las ecuaciones de Fresnel.
2. Es mejor, si tenemos 3 elementos en contacto, poner como nodo de referencia el elemento central.
3. Al poner las caras rugosas, si hay recubrimiento, hay que añadir una capa fina de aire.
4. Al poner los cristales en contacto el revestimiento se elimina en las caras en contacto.

Código

```
TContact *contacto;
contacto = new TContact("entre_cristales","Contacto entre cristales",
"prisma1", "prisma2",identical,"none");
```

Fuentes de luz

Han de introducirse después de establecer todos los nodos. En este caso hemos definido una fuente de luz puntual y espontánea, en el centro del cristal. El tercer argumento está relacionado con el parámetro x usado como abcisa al dibujar los resultados por la clase TPublication. Los argumentos cuarto y quinto están relacionados con si se muestran o no los histogramas y estadísticas de cada paso, i.e., las asociadas al pointer 'gGp'. Las globales sí que se muestran y no están relacionadas con estos argumentos.

Código

```
TSpontan *fuente;
fuente = new TSpontan("fuente","fuente","prisma2",-0.2499,0.0,0.0,600.0);
for (Int_t i=1;i<=100;i++) {
fuente->Gen(i,100,-2.0,true,false);
 }
//Generación de 100 fotones en 100 pasos (i.e., de 10000 fotones)
```

Fuentes de luz

Han de introducirse después de establecer todos los nodos. En este caso hemos definido una fuente de luz puntual y espontánea, en el centro del cristal. El tercer argumento está relacionado con el parámetro x usado como abcisa al dibujar los resultados por la clase TPublication. Los argumentos cuarto y quinto están relacionados con si se muestran o no los histogramas y estadísticas de cada paso, i.e., las asociadas al pointer 'gGp'. Las globales sí que se muestran y no están relacionadas con estos argumentos.

Código

```
TSpontan *fuente;
fuente = new TSpontan("fuente","fuente","prisma2",-0.2499,0.0,0.0,600.0);
for (Int_t i=1;i<=100;i++) {
fuente->Gen(i,100,-2.0,true,false);
 }
//Generación de 100 fotones en 100 pasos (i.e., de 10000 fotones)
```

Publicación de resultados

Por último, podemos especificar los resultados que queremos ver impresos en la pantalla al finalizar el programa.

Código

```
cout << "fNbPart:    " << gGs->fNpGener    << endl;//Nº total de fotones generados.
cout << "fNpSeen:    " << gGs->fNpSeen    << endl;//Nº de fotones detectados.
cout << "fNpLossAny: " << gGs->fNpLossAny << endl;//Nº de fotones perdidos por alguna razón.
cout << "fNpAbsMat:  " << gGs->fNpAbsMat  << endl;//Nº de fotones absorbidos en los materiales.
cout << "fNpAbsRvt:  " << gGs->fNpAbsRvt  << endl;//Nº de fotones perdidos en el revesitmiento.
cout << "fNpOutSide: " << gGs->fNpOutSide << endl;//Nº de fotones que abandonan el montaje.
cout << "fNpAbnorm:  " << gGs->fNpAbnorm  << endl;//Nº de fotones que son destruidos antes
de generarse.
cout << "fNpLossQE:  " << gGs->fNpLossQE  << endl;//Nº de fotones perdidos porque el detector
 no es  perfecto

  exit();

  }
```

Publicación de resultados

Por último, podemos especificar los resultados que queremos ver impresos en la pantalla al finalizar el programa.

Código

```
cout << "fNbPart:    " << gGs->fNpGener    << endl;//Nº total de fotones generados.
cout << "fNpSeen:    " << gGs->fNpSeen     << endl;//Nº de fotones detectados.
cout << "fNpLossAny: " << gGs->fNpLossAny  << endl;//Nº de fotones perdidos por alguna razón.
cout << "fNpAbsMat:  " << gGs->fNpAbsMat   << endl;//Nº de fotones absorbidos en los materiales.
cout << "fNpAbsRvt:  " << gGs->fNpAbsRvt   << endl;//Nº de fotones perdidos en el revesitmiento.
cout << "fNpOutSide: " << gGs->fNpOutSide  << endl;//Nº de fotones que abandonan el montaje.
cout << "fNpAbnorm:  " << gGs->fNpAbnorm   << endl;//Nº de fotones que son destruidos antes
de generarse.
cout << "fNpLossQE:  " << gGs->fNpLossQE   << endl;//Nº de fotones perdidos porque el detector
 no es  perfecto

  exit();

  }
```

LITRANI en acción

```
[adri@nuc9 programas]$ q
bash: q: command not found
[adri@nuc9 programas]$ litrani gammas04.C
  ******************************************
  *                                        *
  *       W E L C O M E  to  R O O T       *
  *                                        *
  *   Version   5.19/02     11 March 2008  *
  *                                        *
  *  You are welcome to visit our Web site *
  *          http://root.cern.ch           *
  *                                        *
  ******************************************

ROOT 5.19/02 (trunk@22600, Mar 12 2008, 09:24:00 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.16.29, Jan 08, 2008
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing gammas04.C...

size     : 5
name     : gammas04
listing  : Paso de fotones gamma por un centellador cilíndrico
upcom    : La última cara es un detector de superficie
downcom  : ¿Qué detecto?
otherseq : 0
indebug  : 0
WithDate : 1

Start from Litrani, libLitrani is already loaded
calling TLitGlob

  ******************************************
```

Interpolaciones con SplineFit

Inicio de la interpolación

En primer lugar, llamamos al programa y le asignamos un nombre a la interpolación.

1. todraw: kTRUE si queremos dibujar la interpolación.
2. infile: kTRUE si la interpolación va a crear un archivo SplineFitDB.rdb.
3. firstinfile: kTRUE si queremos borrar todas las interpolaciones presentes en el archivo SplineFitDB.rdb y sustuirlos por la interpolación presente.

Código

```
TSplineFit* nai_cross_section(Bool_t todraw = kFALSE, Bool_t infile = kFALSE,
Bool_t firstinfile = kFALSE)
```

Interpolaciones con SplineFit

Inicio de la interpolación

En primer lugar, llamamos al programa y le asignamos un nombre a la interpolación.

1. todraw: kTRUE si queremos dibujar la interpolación.
2. infile: kTRUE si la interpolación va a crear un archivo SplineFitDB.rdb.
3. firstinfile: kTRUE si queremos borrar todas las interpolaciones presentes en el archivo SplineFitDB.rdb y sustuirlos por la interpolación presente.

Código

```
TSplineFit* nai_cross_section(Bool_t todraw = kFALSE, Bool_t infile = kFALSE,
Bool_t firstinfile = kFALSE)
```

Código: creación de la tabla de datos

```
Int_t k1;
  Int_t k2 = -100;
  k1 = TClassTable::GetID("TSplineFit");
  if (k1<0) k2 = gSystem.Load("libSplineFit");
  const Int_t M = 74;
  Int_t i;
```

Código: introducción de los datos

```
TSplineFit *nai_cross_section;
  Double_t x[M] = { 0.001,  0.001017,  0.001035,  0.001053 };

  Double_t y[M] = { 602625.1, 581738.2, 560851.3, 540738.0, 521398.3 };
  nai_cross_section = new TSplineFit("PhotoEl_nai", "Photo-Electric Cross Section | nai",
 18, M, x, y, 0.001,  1.4);
```

Código: definición de unidades, ejes y comentarios

```
 nai_cross_section->SetSource("http://physics.nist.gov/PhysRefData/Xcom/Text/XCOM.html");
  nai_cross_section->SetMacro("nai_cross_section.C");
  nai_cross_section->SetXLabel("Gamma Energy [MeV]");
  nai_cross_section->SetYLabel("Cross Section x10-24 cm2");
  return nai_cross_section;
}
```

Código: creación de la tabla de datos

```
Int_t k1;
  Int_t k2 = -100;
  k1 = TClassTable::GetID("TSplineFit");
  if (k1<0) k2 = gSystem.Load("libSplineFit");
  const Int_t M = 74;
  Int_t i;
```

Código: introducción de los datos

```
TSplineFit *nai_cross_section;
  Double_t x[M] = { 0.001,  0.001017,  0.001035,  0.001053 };

  Double_t y[M] = { 602625.1, 581738.2, 560851.3, 540738.0, 521398.3 };
  nai_cross_section = new TSplineFit("PhotoEl_nai", "Photo-Electric Cross Section | nai",
 18, M, x, y, 0.001,  1.4);
```

Código: definición de unidades, ejes y comentarios

```
 nai_cross_section->SetSource("http://physics.nist.gov/PhysRefData/Xcom/Text/XCOM.html");
  nai_cross_section->SetMacro("nai_cross_section.C");
  nai_cross_section->SetXLabel("Gamma Energy [MeV]");
  nai_cross_section->SetYLabel("Cross Section x10-24 cm2");
  return nai_cross_section;
}
```

Código: creación de la tabla de datos

```
Int_t k1;
  Int_t k2 = -100;
  k1 = TClassTable::GetID("TSplineFit");
  if (k1<0) k2 = gSystem.Load("libSplineFit");
  const Int_t M = 74;
  Int_t i;
```

Código: introducción de los datos

```
TSplineFit *nai_cross_section;
  Double_t x[M] = { 0.001,  0.001017,  0.001035,  0.001053 };

  Double_t y[M] = { 602625.1, 581738.2, 560851.3, 540738.0, 521398.3 };
  nai_cross_section = new TSplineFit("PhotoEl_nai", "Photo-Electric Cross Section | nai",
 18, M, x, y, 0.001,  1.4);
```

Código: definición de unidades, ejes y comentarios

```
 nai_cross_section->SetSource("http://physics.nist.gov/PhysRefData/Xcom/Text/XCOM.html");
  nai_cross_section->SetMacro("nai_cross_section.C");
  nai_cross_section->SetXLabel("Gamma Energy [MeV]");
  nai_cross_section->SetVLabel("Cross Section x10-24 cm2");
  return nai_cross_section;
}
```

Representación de resultados con VisuLitrani y TwoPadDisplay

Llamar a VisuLitrani

En todo lo que hemos hecho hasta ahora no hemos empleado VisuLitrani, puesto que es posible extraer resultados del programa sin necesidad de ponerlo en marcha.En primer lugar, para que VisuLitrani se ponga en marcha hay que dibujar el montaje desde LITRANI. Para ello debemos añadir algunos comandos adicionales a la hora de definir los nodos.

Código

```
  TSNode *node1;
  node1=new TSNode("node1","node1",centellador);
  node1->SetLineColor(1);
  node1->SetLineWidth(2);
  node1->cd();
\tiny
```

Representación de resultados con VisuLitrani y TwoPadDisplay

Llamar a VisuLitrani

En todo lo que hemos hecho hasta ahora no hemos empleado VisuLitrani, puesto que es posible extraer resultados del programa sin necesidad de ponerlo en marcha.En primer lugar, para que VisuLitrani se ponga en marcha hay que dibujar el montaje desde LITRANI. Para ello debemos añadir algunos comandos adicionales a la hora de definir los nodos.

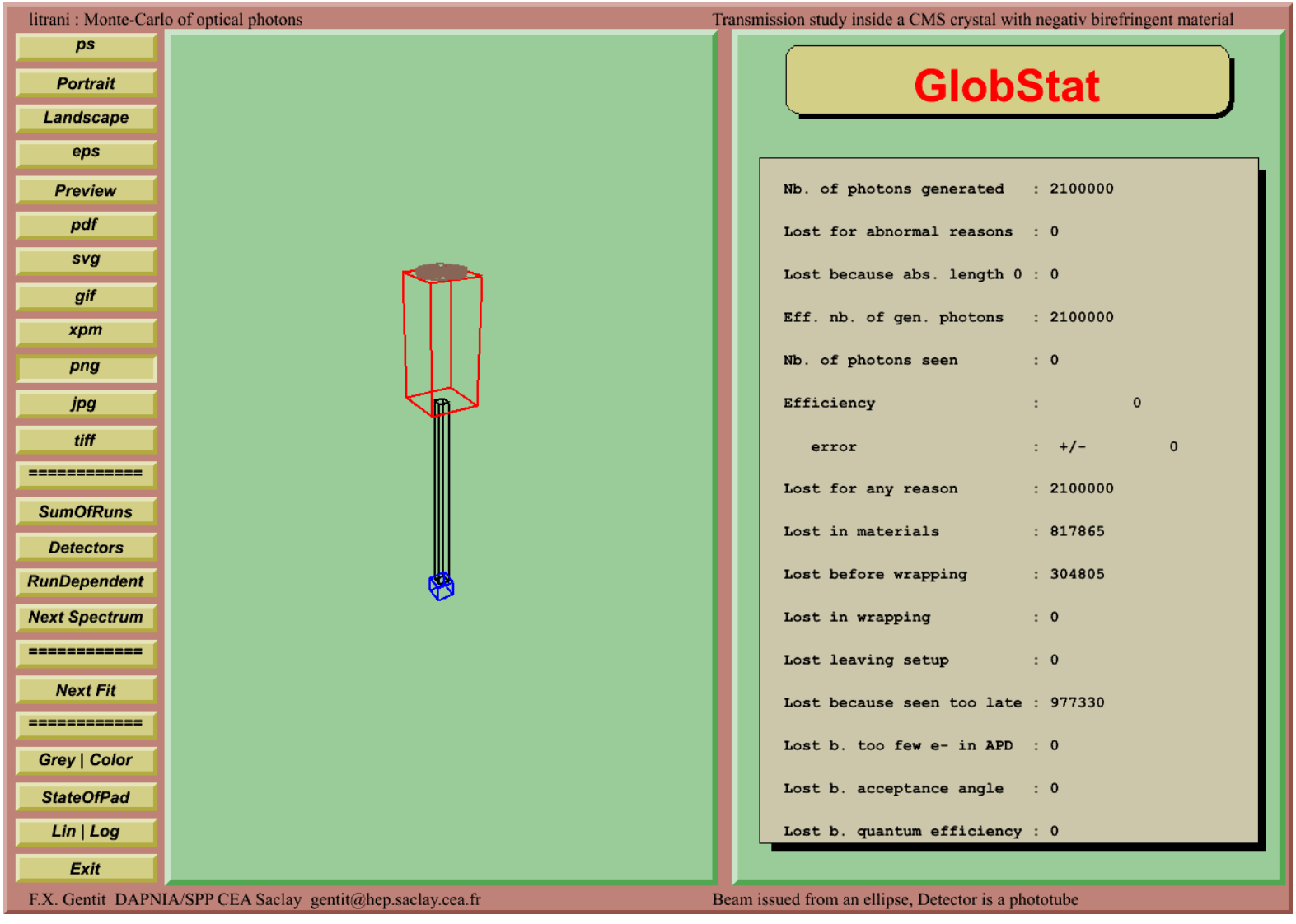
Código

```
  TSNode *node1;
  node1=new TSNode("node1","node1",centellador);
  node1->SetLineColor(1);
  node1->SetLineWidth(2);
  node1->cd();
\tiny
```

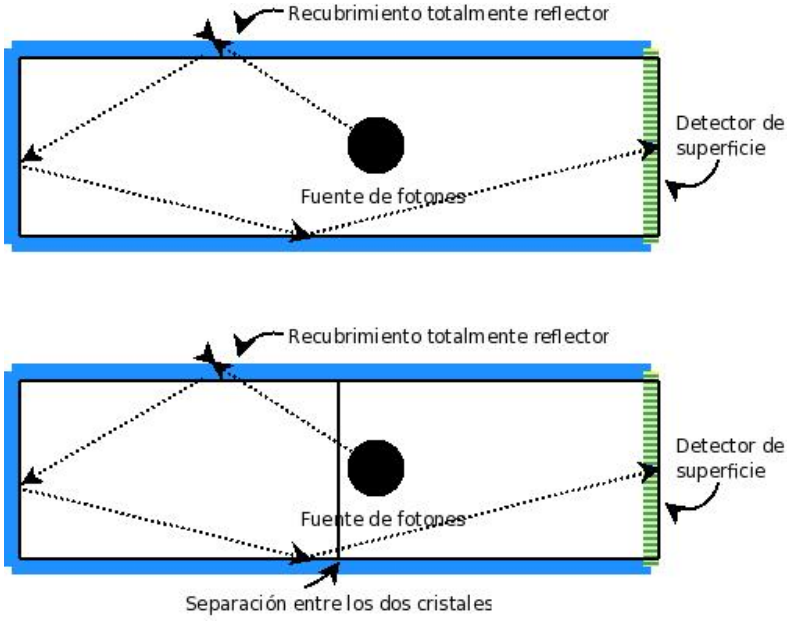
Histogramas

- Histogramas de detector: Según el tipo de detectores que tengamos en cada montaje podemos llamar a diversas clases que permitan buscar los histogramas más adecuados para cada detector:
  1. **TStatSurfD:** Para detectores de superficie y fototubos.
  2. **TStatVolD:** Para detectores de volumen.
  3. **TStatAPD:** Para APD's.

VisuLitrani en acción



Validación de problemas sencillos con LITRANI



Equivalencia de montajes

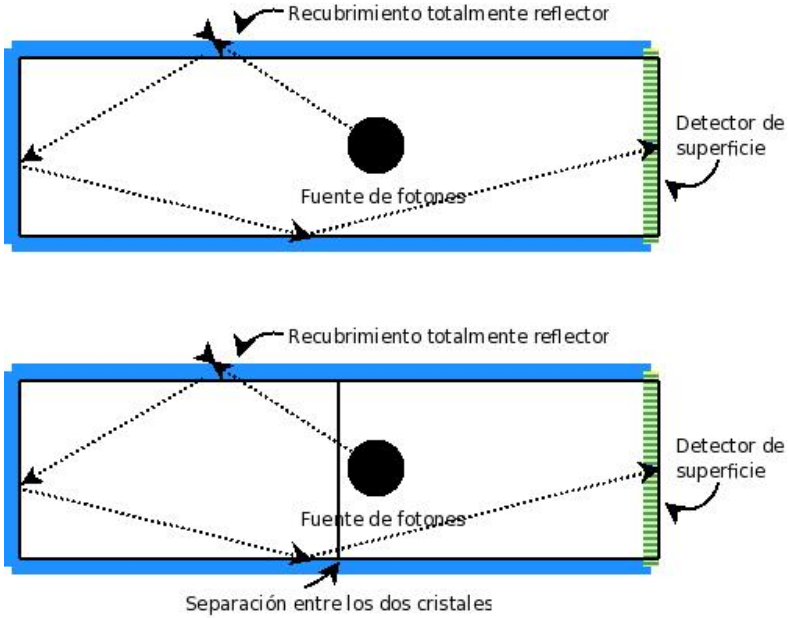
- Una fuente de fotones se encuentra en el interior de un cristal recubierto de un material totalmente reflector y con un detector de superficie en una de sus caras.
- Dos cristales cuya longitud es la mitad del anterior pegados perfectamente y en las mismas condiciones.

En principio estos dos montajes tienen que ser completamente equivalentes.

Resultados

	Un solo cristal	Dos cristales	Fluctuación aproximada
Fotones emitidos	10000	10000	-
Fotones vistos	8376	8396	91
Fotones perdidos en el material	1624	1604	40

Validación de problemas sencillos con LITRANI



Equivalencia de montajes

- Una fuente de fotones se encuentra en el interior de un cristal recubierto de un material totalmente reflector y con un detector de superficie en una de sus caras.
- Dos cristales cuya longitud es la mitad del anterior pegados perfectamente y en las mismas condiciones.

En principio estos dos montajes tienen que ser completamente equivalentes.

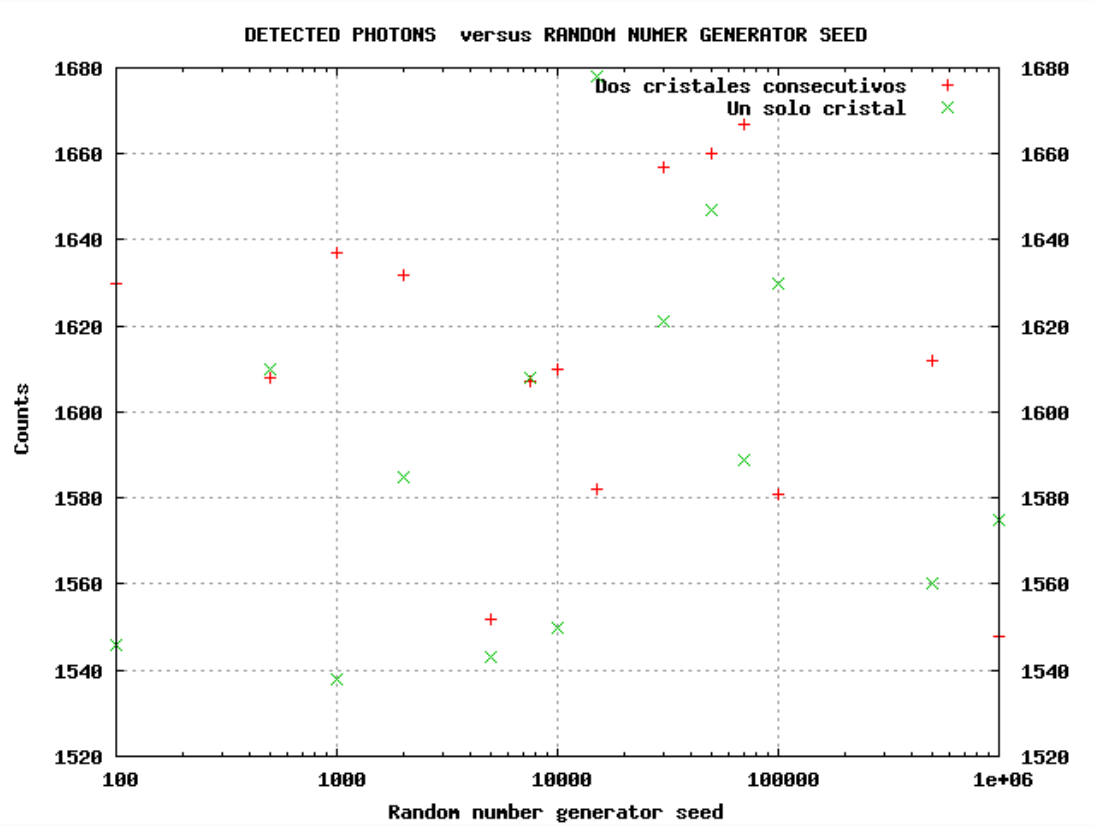
Resultados

	Un solo cristal	Dos cristales	Fluctuación aproximada
Fotones emitidos	10000	10000	-
Fotones vistos	8376	8396	91
Fotones perdidos en el material	1624	1604	40

Modificamos la semilla que genera los números aleatorios

```
gROOT->ProcessLine(".x InitLitrani.C(8,name,listing,upcom,downcom,kTRUE,kFALSE,kTRUE)");
gRandom3->SetSeed(seed);
    \tiny
```

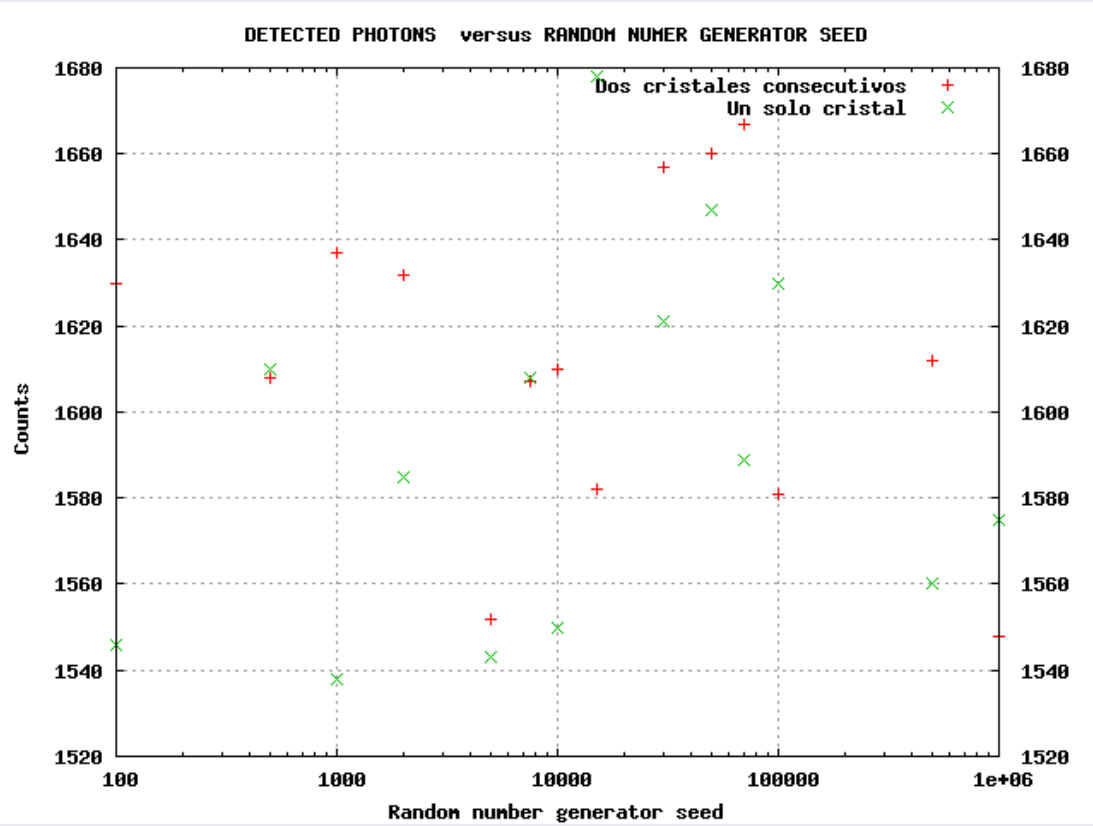
Resultado de las simulaciones

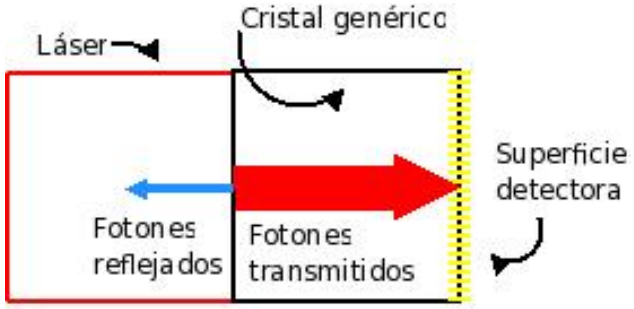


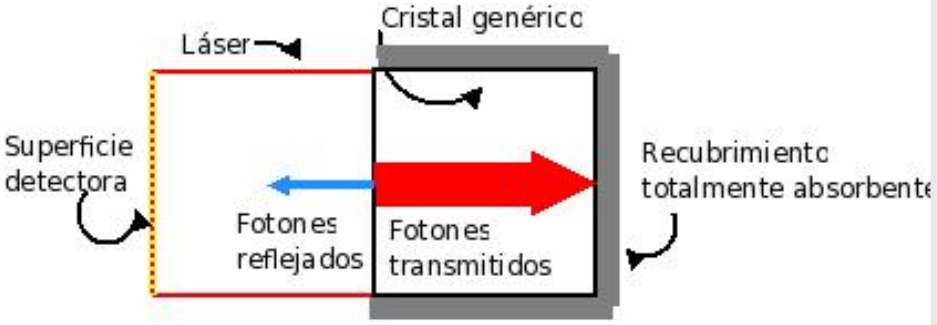
Modificamos la semilla que genera los números aleatorios

```
gROOT->ProcessLine(".x InitLitrani.C(8,name,listing,upcom,downcom,kTRUE,kFALSE,kTRUE)");
gRandom3->SetSeed(seed);
    \tiny
```

Resultado de las simulaciones







Reflectancia y transmitancia

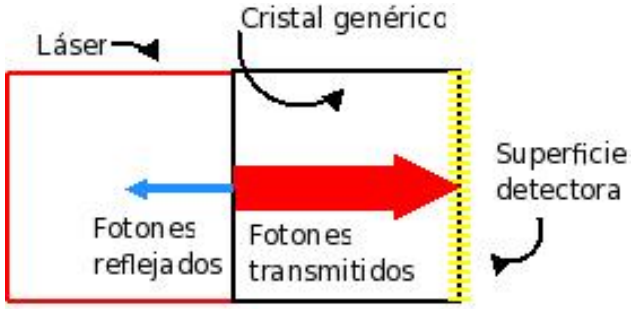
En incidencia normal, la reflectancia y la transmitancia son

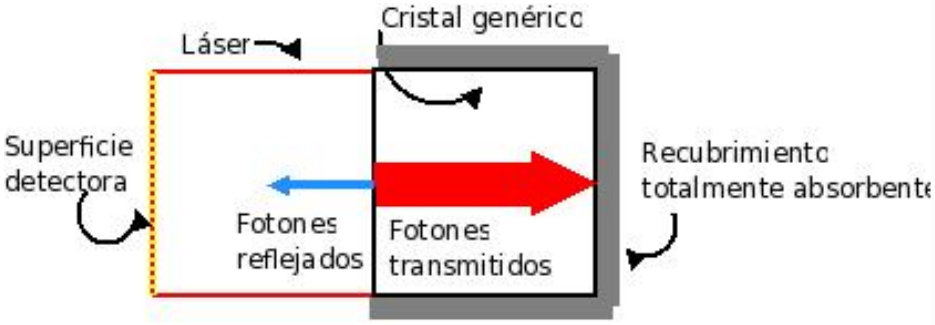
$$R = \left(\frac{n_t - n_i}{n_t + n_i}\right)^2 ; T = \frac{4 n_t n_i}{(n_t + n_i)^2}$$

Comparemos los resultados de la simulación en LITRANI con los dados por las fórmulas de Fresnel.

Resultados de la simulación

	LITRANI	Resultado teórico
Fotones emitidos	10000	10000
Fotones transmitidos	9600	9600
Fotones reflejados	400	400





Reflectancia y transmitancia

En incidencia normal, la reflectancia y la transmitancia son

$$R = \left(\frac{n_t - n_i}{n_t + n_i}\right)^2 ; T = \frac{4 n_t n_i}{(n_t + n_i)^2}$$

Comparemos los resultados de la simulación en LITRANI con los dados por las fórmulas de Fresnel.

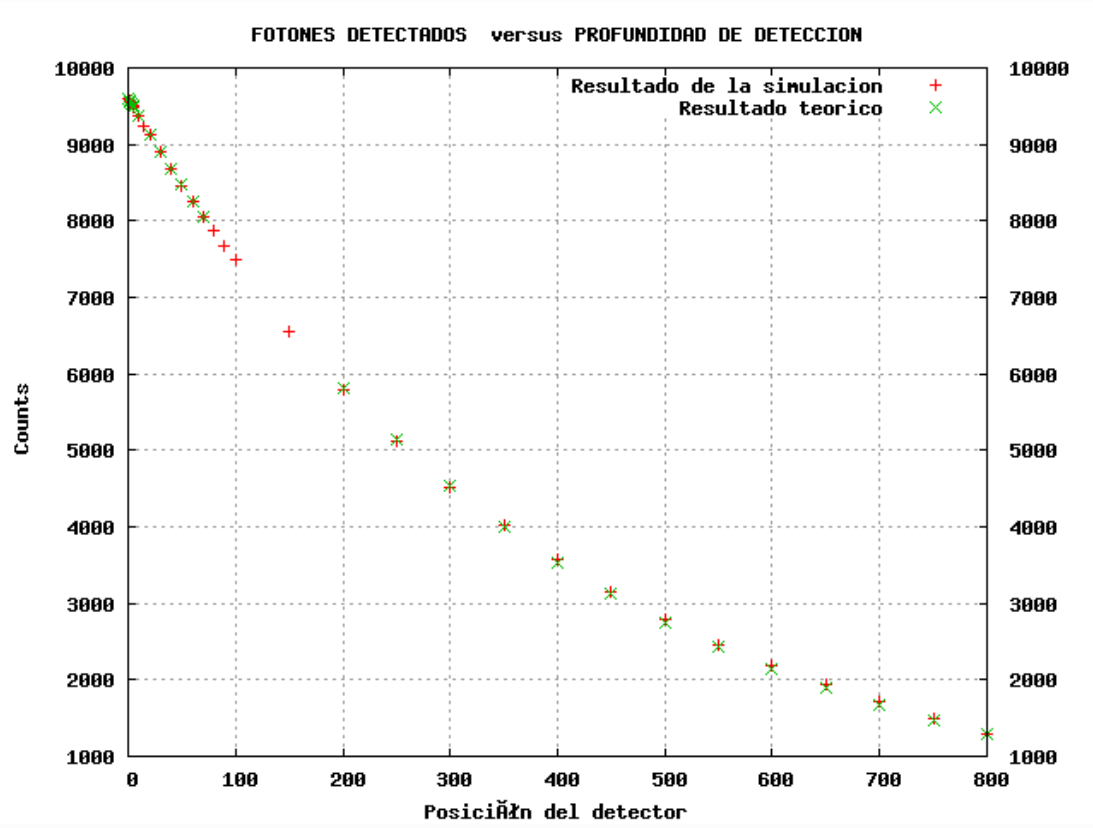
Resultados de la simulación

	LITRANI	Resultado teórico
Fotones emitidos	10000	10000
Fotones transmitidos	9600	9600
Fotones reflejados	400	400

Absorción en un dieléctrico isótropo

$L_a = \frac{c}{2\omega n_I} = \frac{\lambda}{4\pi n_I}$, de modo que la intensidad decrecerá a medida que penetre en el material según $I(x) = I_0 e^{x/L_a}$

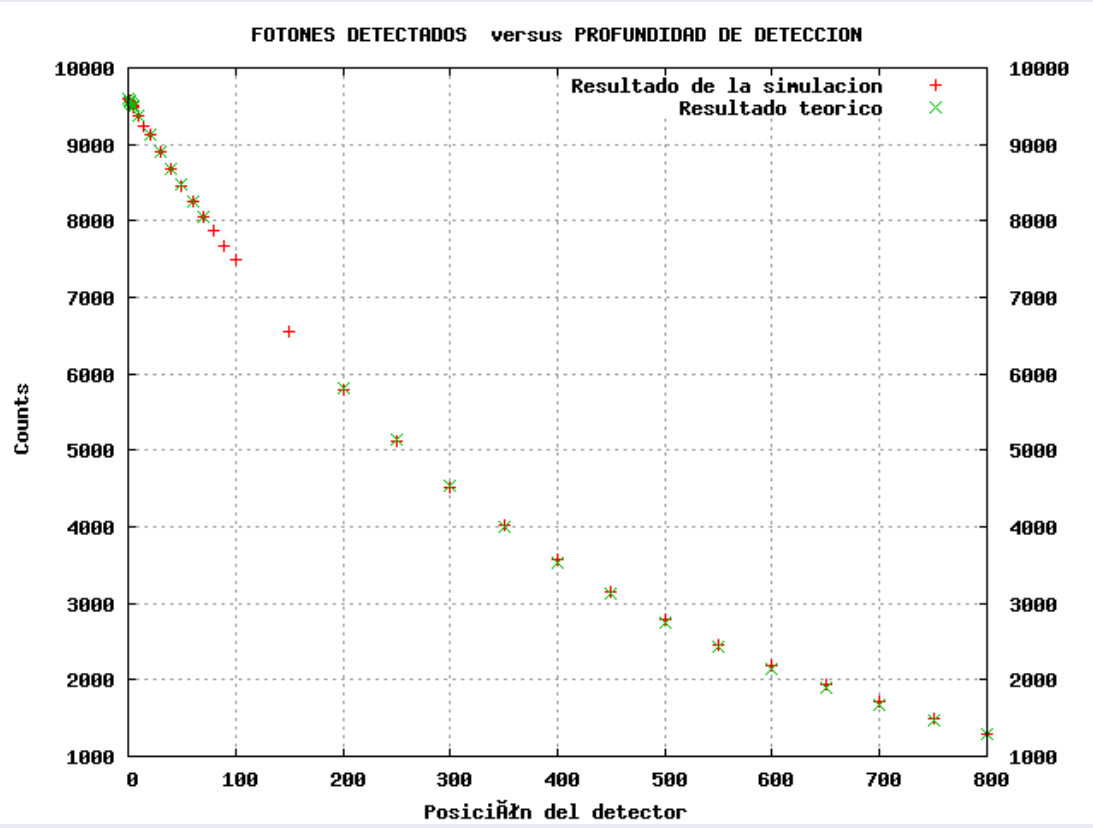
Resultado de la simulación



Absorción en un dieléctrico isótropo

$L_a = \frac{c}{2\omega n_I} = \frac{\lambda}{4\pi n_I}$, de modo que la intensidad decrecerá a medida que penetre en el material según $I(x) = I_0 e^{x/L_a}$

Resultado de la simulación

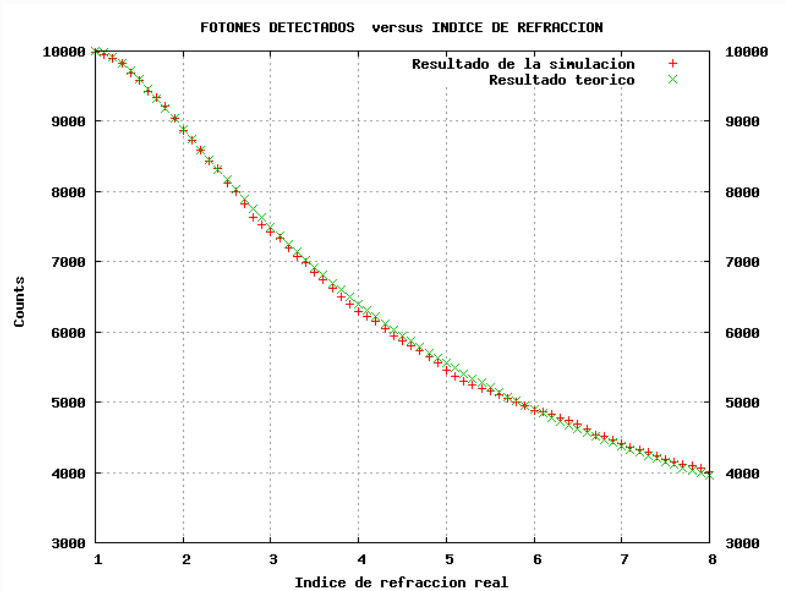


Ecuaciones de Fresnel

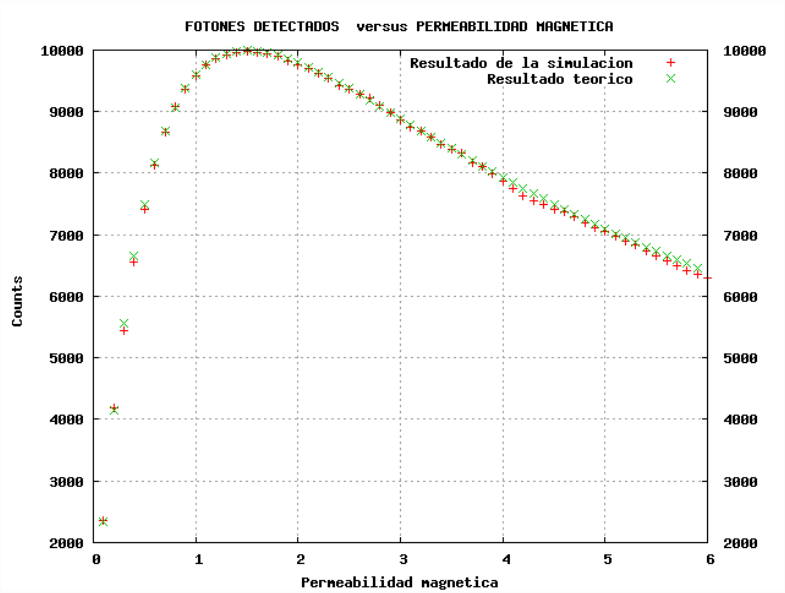
$$r_\perp \equiv \left(\frac{E_{0r}}{E_{0i}}\right)_\perp = \frac{\frac{n_i}{\mu_i}\cos\theta_i - \frac{n_t}{\mu_t}\cos\theta_t}{\frac{n_i}{\mu_i}\cos\theta_i + \frac{n_t}{\mu_t}\cos\theta_t} \qquad r_\parallel \equiv \left(\frac{E_{0r}}{E_{0i}}\right)_\parallel = \frac{\frac{n_t}{\mu_t}\cos\theta_i - \frac{n_i}{\mu_i}\cos\theta_t}{\frac{n_i}{\mu_i}\cos\theta_t + \frac{n_t}{\mu_t}\cos\theta_i}$$

$$t_\perp \equiv \left(\frac{E_{0t}}{E_{0i}}\right)_\perp = \frac{2\frac{n_i}{\mu_i}\cos\theta_i}{\frac{n_i}{\mu_i}\cos\theta_i + \frac{n_t}{\mu_t}\cos\theta_t} \qquad t_\parallel \equiv \left(\frac{E_{0t}}{E_{0i}}\right)_\parallel = \frac{2\frac{n_i}{\mu_i}\cos\theta_i}{\frac{n_i}{\mu_i}\cos\theta_t + \frac{n_t}{\mu_t}\cos\theta_i}$$

Variación del índice de refracción



Variación de la permeabilidad magnética

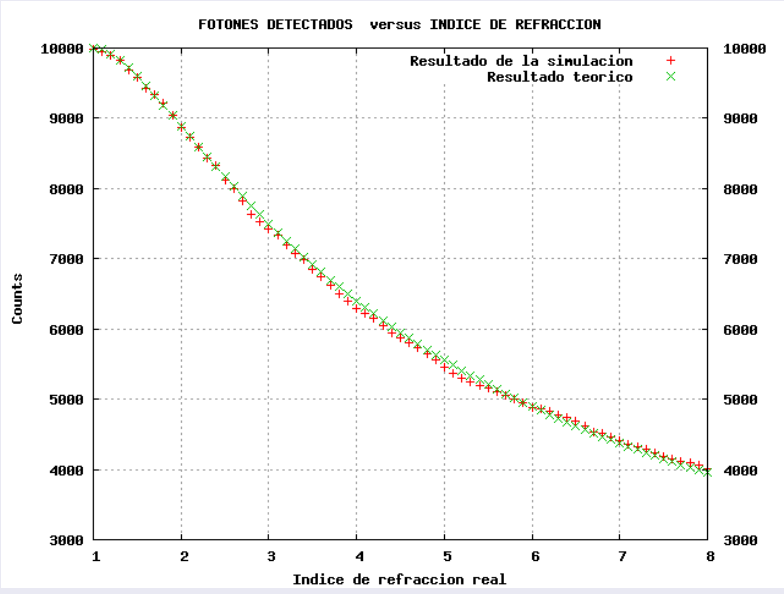


Ecuaciones de Fresnel

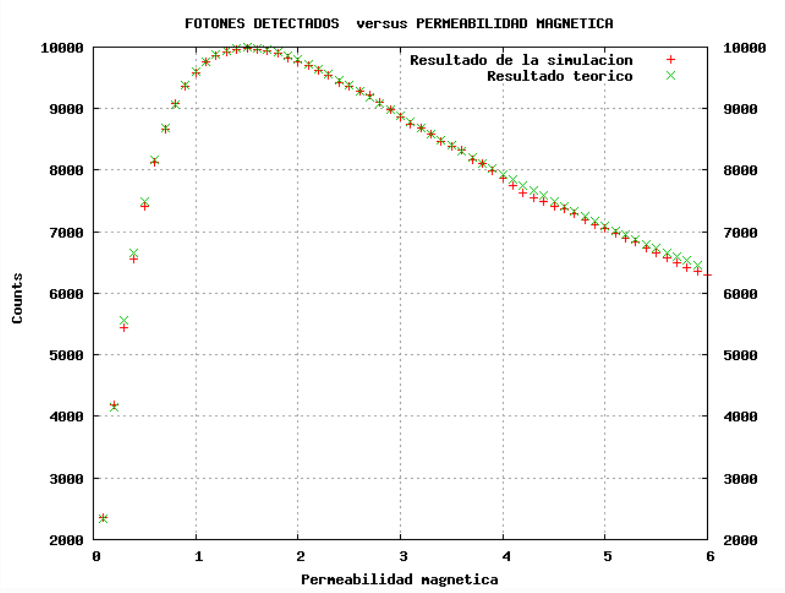
$$r_\perp \equiv \left(\frac{E_{0r}}{E_{0i}}\right)_\perp = \frac{\frac{n_i}{\mu_i}\cos\theta_i - \frac{n_t}{\mu_t}\cos\theta_t}{\frac{n_i}{\mu_i}\cos\theta_i + \frac{n_t}{\mu_t}\cos\theta_t} \qquad r_\parallel \equiv \left(\frac{E_{0r}}{E_{0i}}\right)_\parallel = \frac{\frac{n_t}{\mu_t}\cos\theta_i - \frac{n_i}{\mu_i}\cos\theta_t}{\frac{n_i}{\mu_i}\cos\theta_t + \frac{n_t}{\mu_t}\cos\theta_i}$$

$$t_\perp \equiv \left(\frac{E_{0t}}{E_{0i}}\right)_\perp = \frac{2\frac{n_i}{\mu_i}\cos\theta_i}{\frac{n_i}{\mu_i}\cos\theta_i + \frac{n_t}{\mu_t}\cos\theta_t} \qquad t_\parallel \equiv \left(\frac{E_{0t}}{E_{0i}}\right)_\parallel = \frac{2\frac{n_i}{\mu_i}\cos\theta_i}{\frac{n_i}{\mu_i}\cos\theta_t + \frac{n_t}{\mu_t}\cos\theta_i}$$

Variación del índice de refracción



Variación de la permeabilidad magnética



Ecuaciones de Fresnel

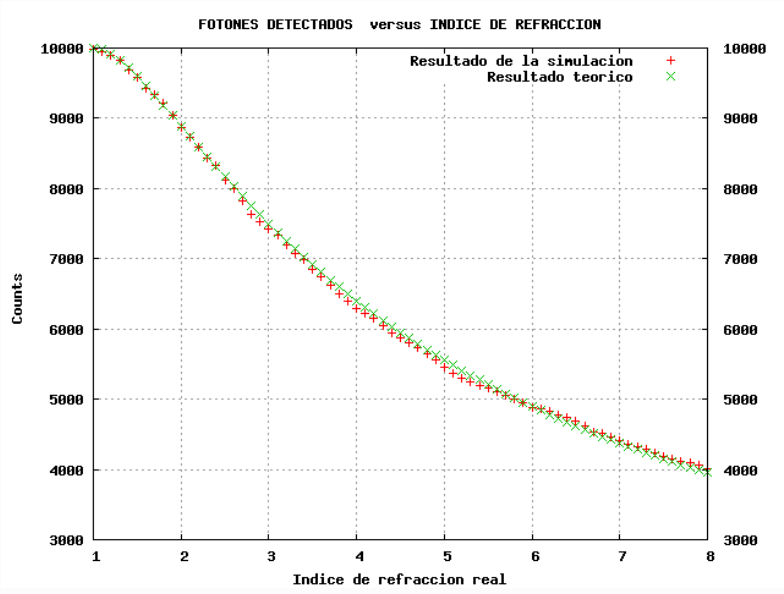
$$r_\perp \equiv \left(\frac{E_{0r}}{E_{0i}}\right)_\perp = \frac{\frac{n_i}{\mu_i}\cos\theta_i - \frac{n_t}{\mu_t}\cos\theta_t}{\frac{n_i}{\mu_i}\cos\theta_i + \frac{n_t}{\mu_t}\cos\theta_t}$$

$$r_\parallel \equiv \left(\frac{E_{0r}}{E_{0i}}\right)_\parallel = \frac{\frac{n_t}{\mu_t}\cos\theta_i - \frac{n_i}{\mu_i}\cos\theta_t}{\frac{n_i}{\mu_i}\cos\theta_t + \frac{n_t}{\mu_t}\cos\theta_i}$$

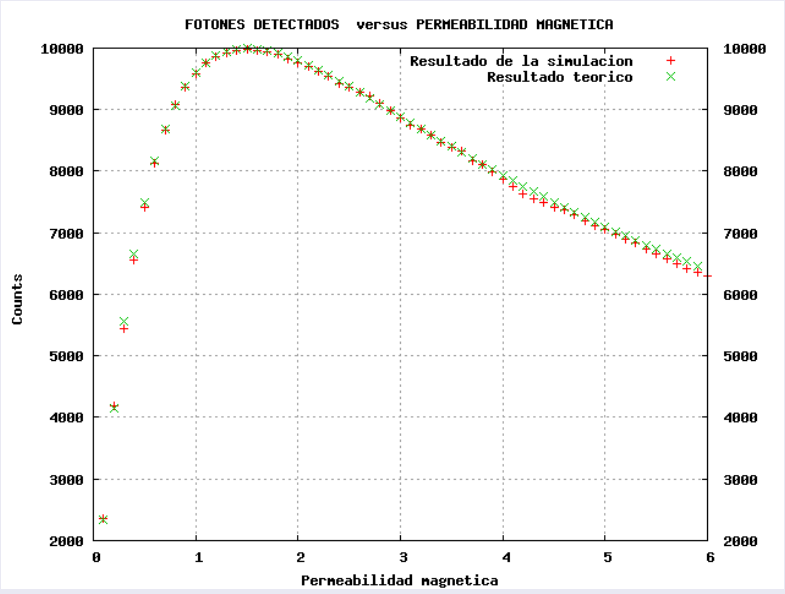
$$t_\perp \equiv \left(\frac{E_{0t}}{E_{0i}}\right)_\perp = \frac{2\frac{n_i}{\mu_i}\cos\theta_i}{\frac{n_i}{\mu_i}\cos\theta_i + \frac{n_t}{\mu_t}\cos\theta_t}$$

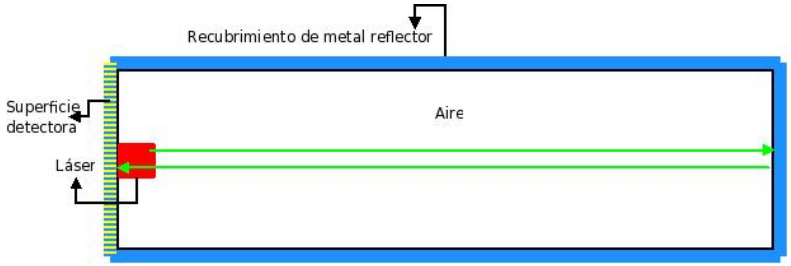
$$t_\parallel \equiv \left(\frac{E_{0t}}{E_{0i}}\right)_\parallel = \frac{2\frac{n_i}{\mu_i}\cos\theta_i}{\frac{n_i}{\mu_i}\cos\theta_t + \frac{n_t}{\mu_t}\cos\theta_i}$$

Variación del índice de refracción



Variación de la permeabilidad magnética



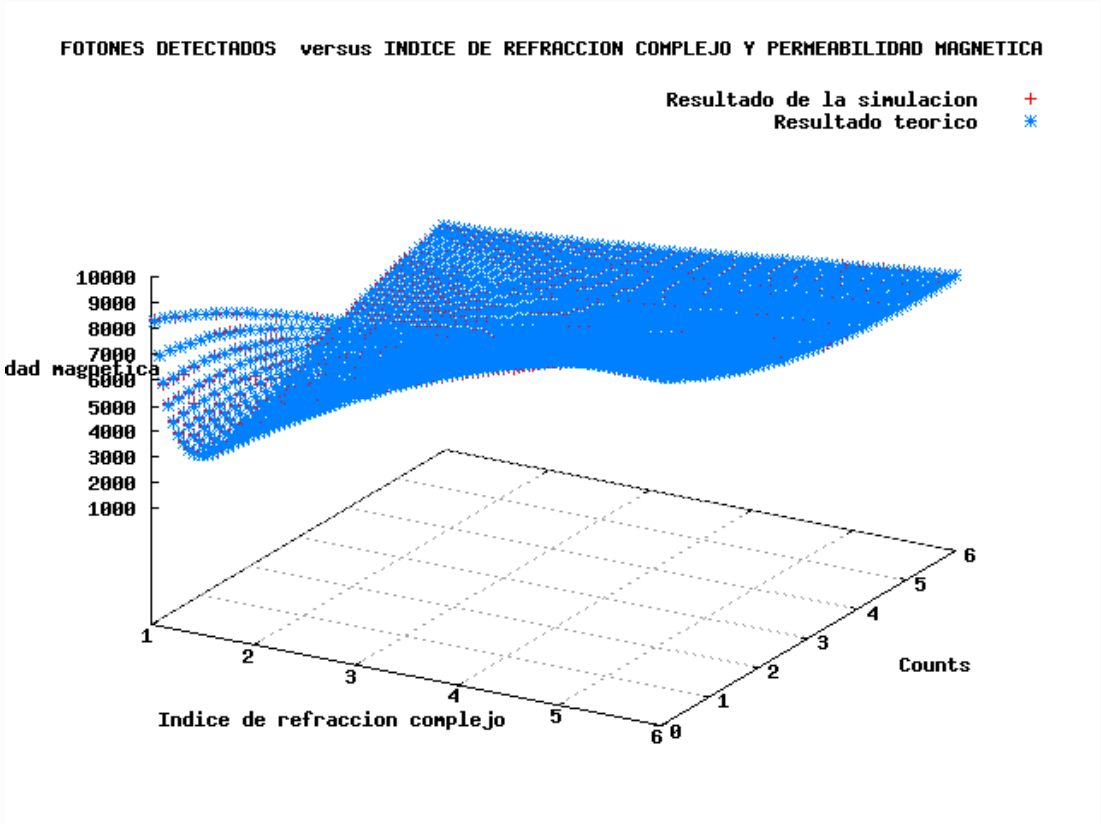


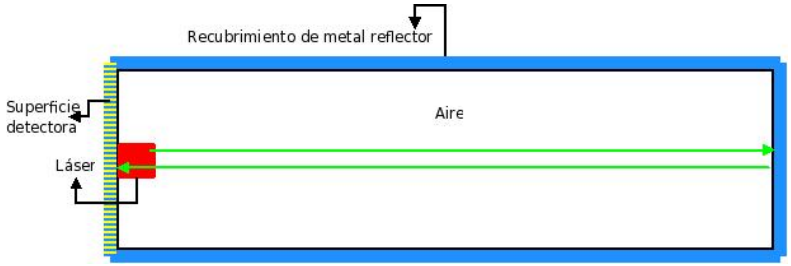
Reflexión en metales

En incidencia normal, la reflectividad de un metal es

$$R = \frac{(n_R - \mu_t)^2 + n_I}{(n_R + \mu_t)^2 + n_I}$$

Resultado de las simulaciones



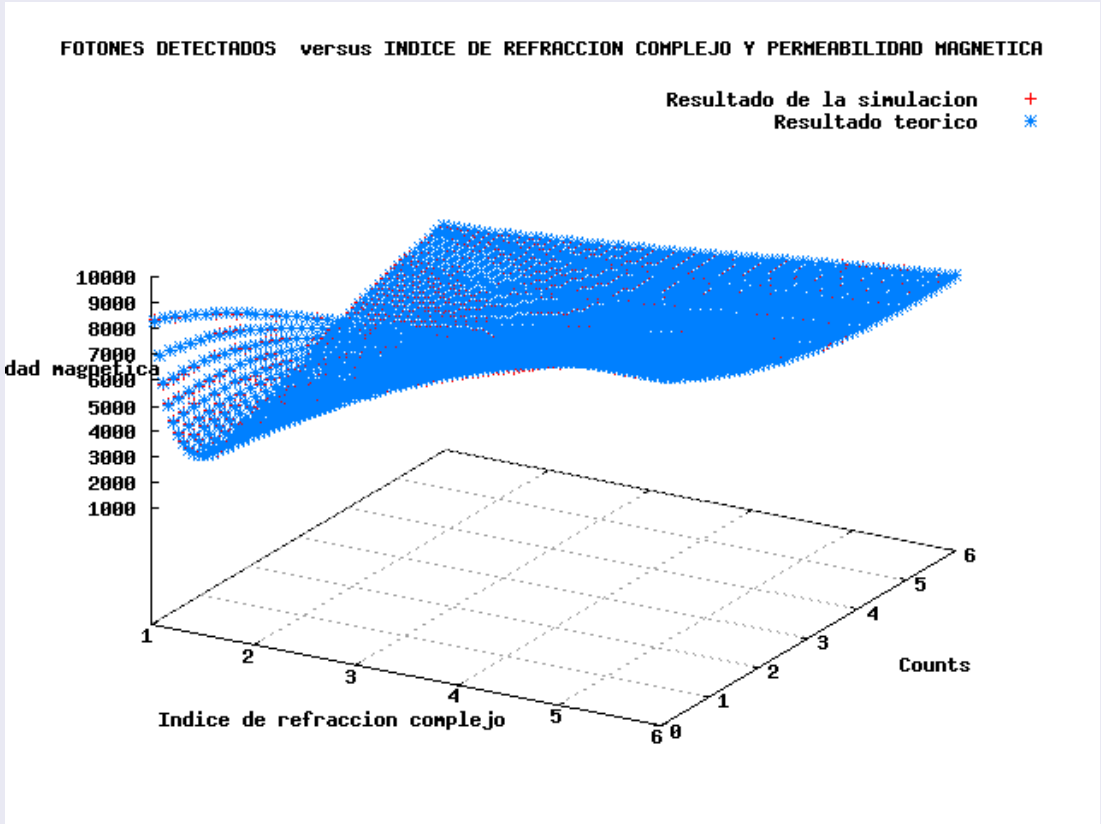


Reflexión en metales

En incidencia normal, la reflectividad de un metal es

$$R = \frac{(n_R - \mu_t)^2 + n_I}{(n_R + \mu_t)^2 + n_I}$$

Resultado de las simulaciones

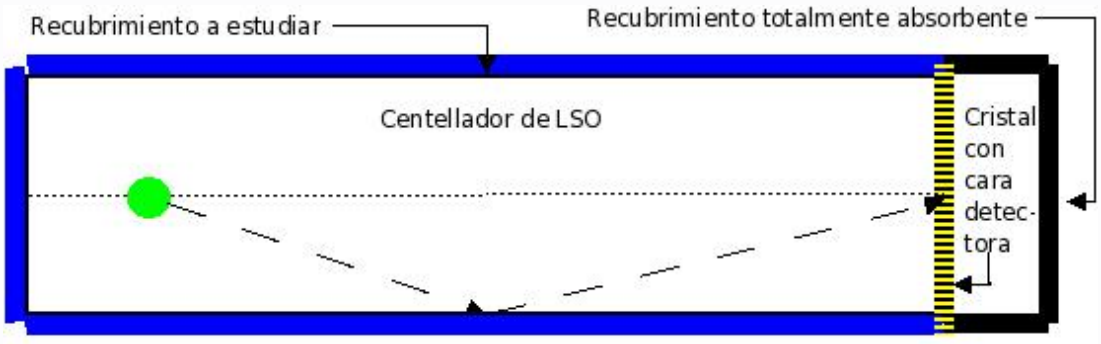


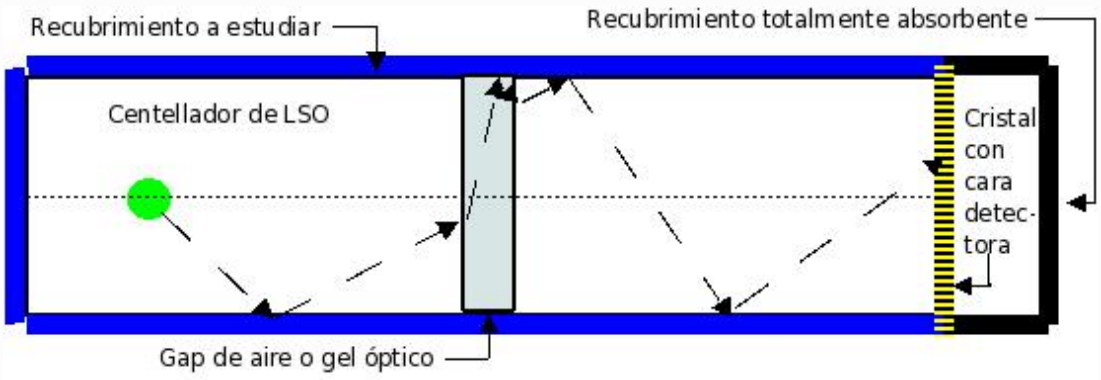
Simulaciones en el interior de centelladores de LSO

Procedimiento

Disponemos de dos tipos de cristales: unos de 2cmx1cmx1cm y otros de 7mmx1,4mmx1,4mm, ambos de LSO. Estudiaremos cómo varía la cantidad de fotones detectados en la última cara en función de la posición a lo largo del eje x de una fuente isótropa de fotones de 600 y 400 nm en función del recubrimiento que rodee al montaje.

Montajes



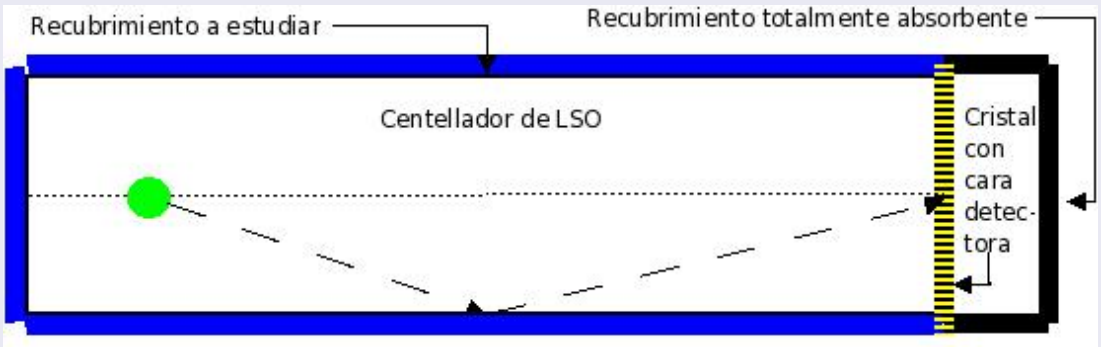


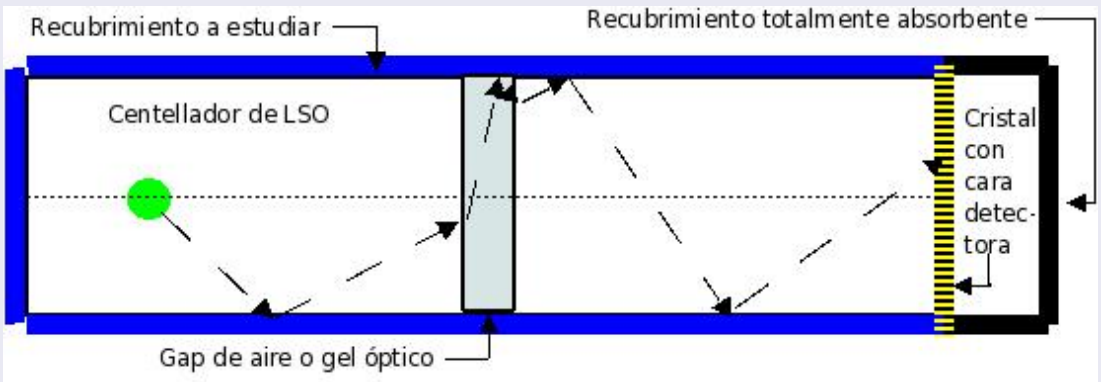
Simulaciones en el interior de centelladores de LSO

Procedimiento

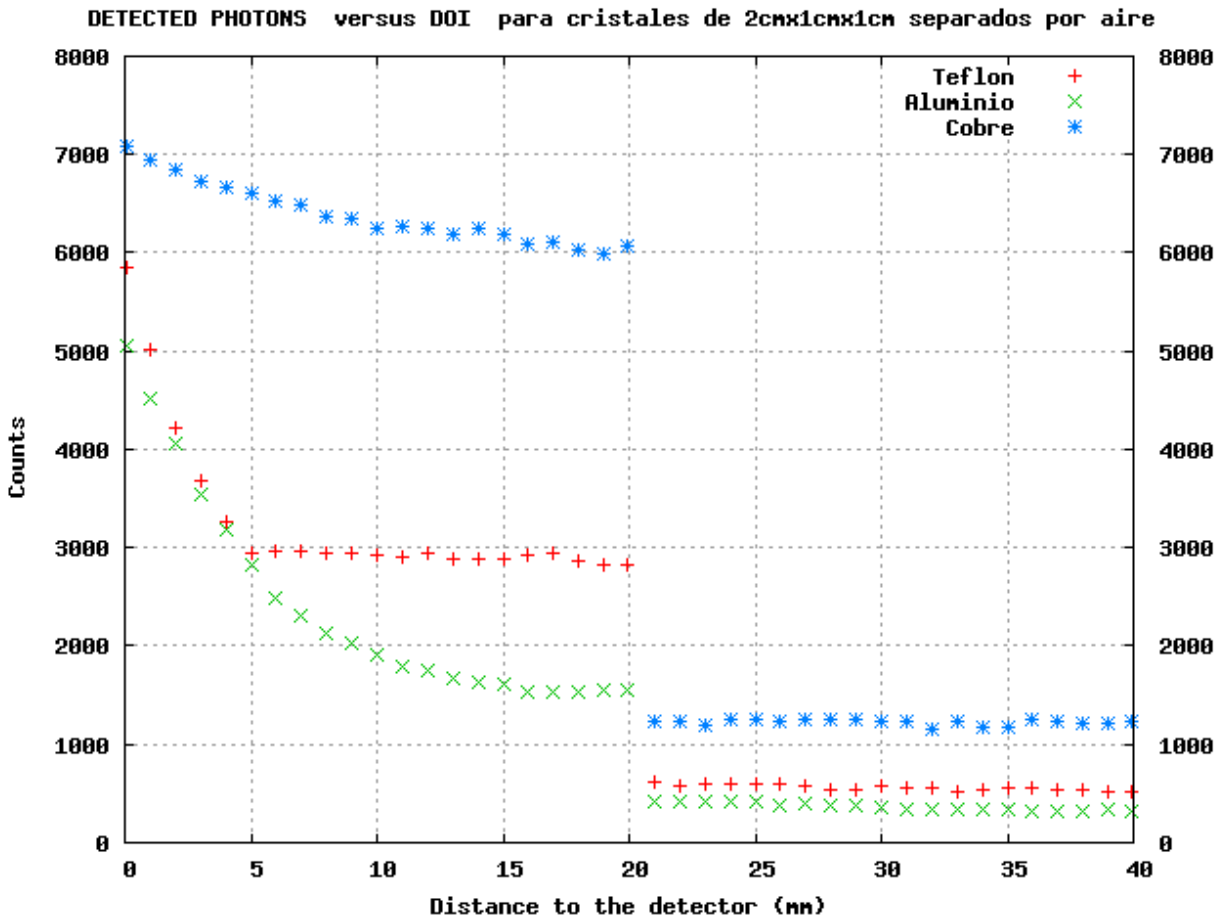
Disponemos de dos tipos de cristales: unos de 2cmx1cmx1cm y otros de 7mmx1,4mmx1,4mm, ambos de LSO. Estudiaremos cómo varía la cantidad de fotones detectados en la última cara en función de la posición a lo largo del eje x de una fuente isótropa de fotones de 600 y 400 nm en función del recubrimiento que rodee al montaje.

Montajes

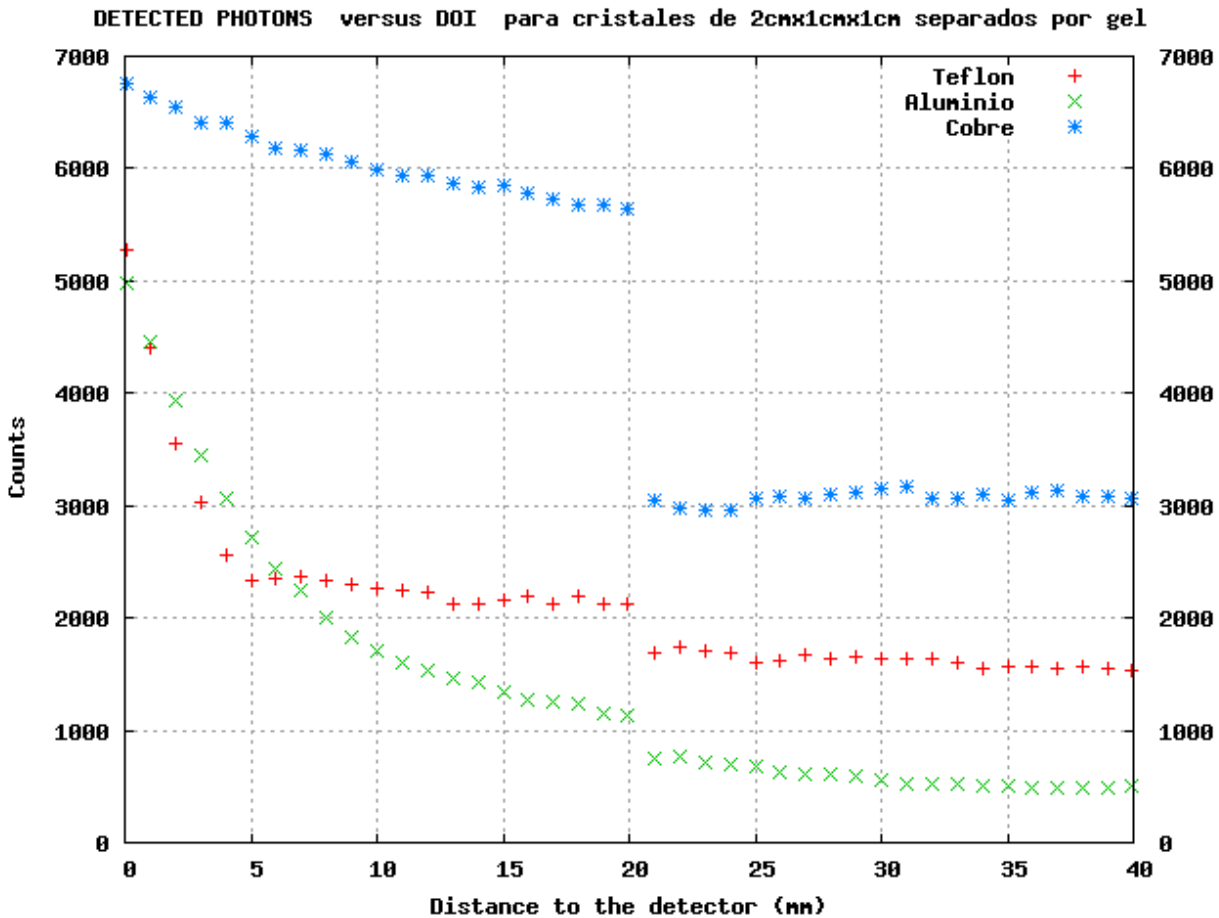




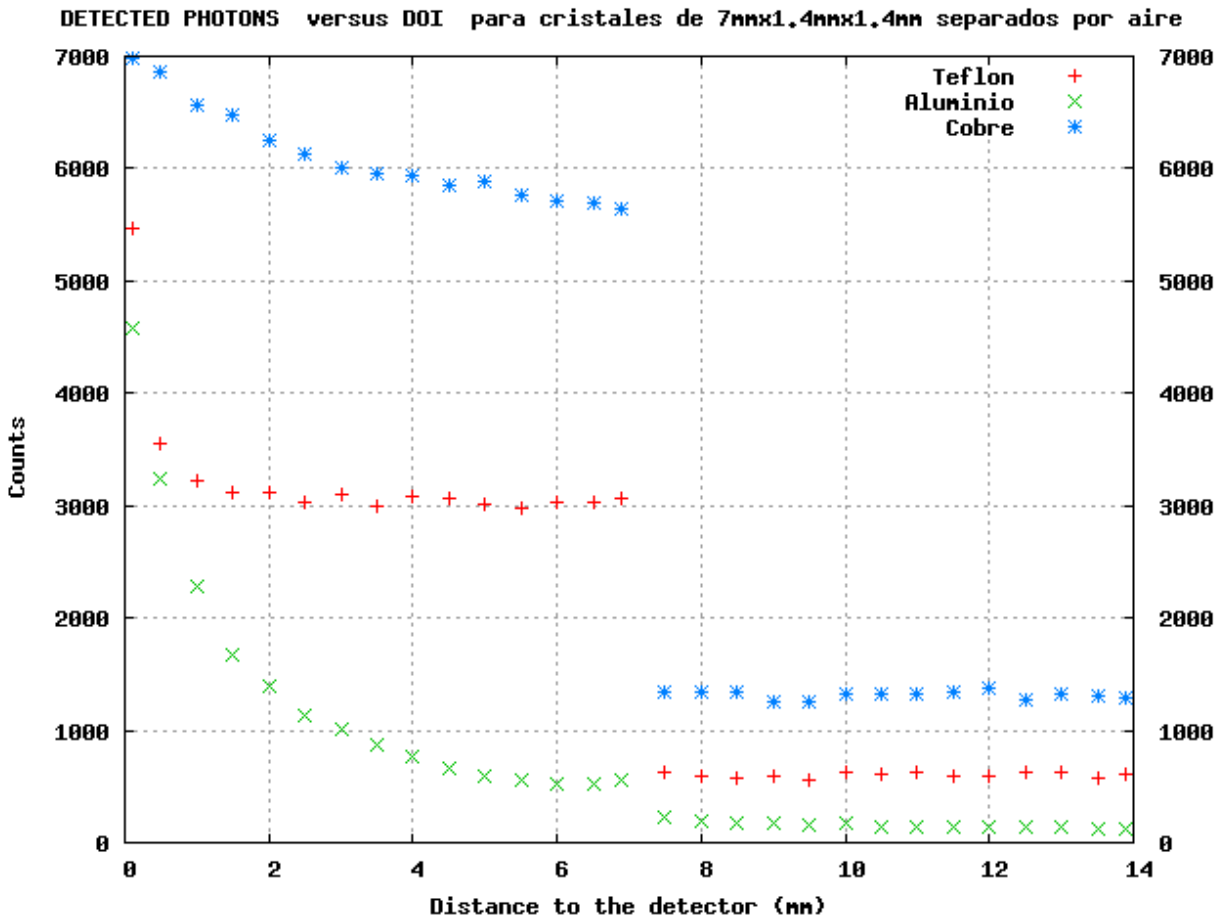
Fotones de 400 nm en cristales de 2cmx1cmx1cm separados por aire



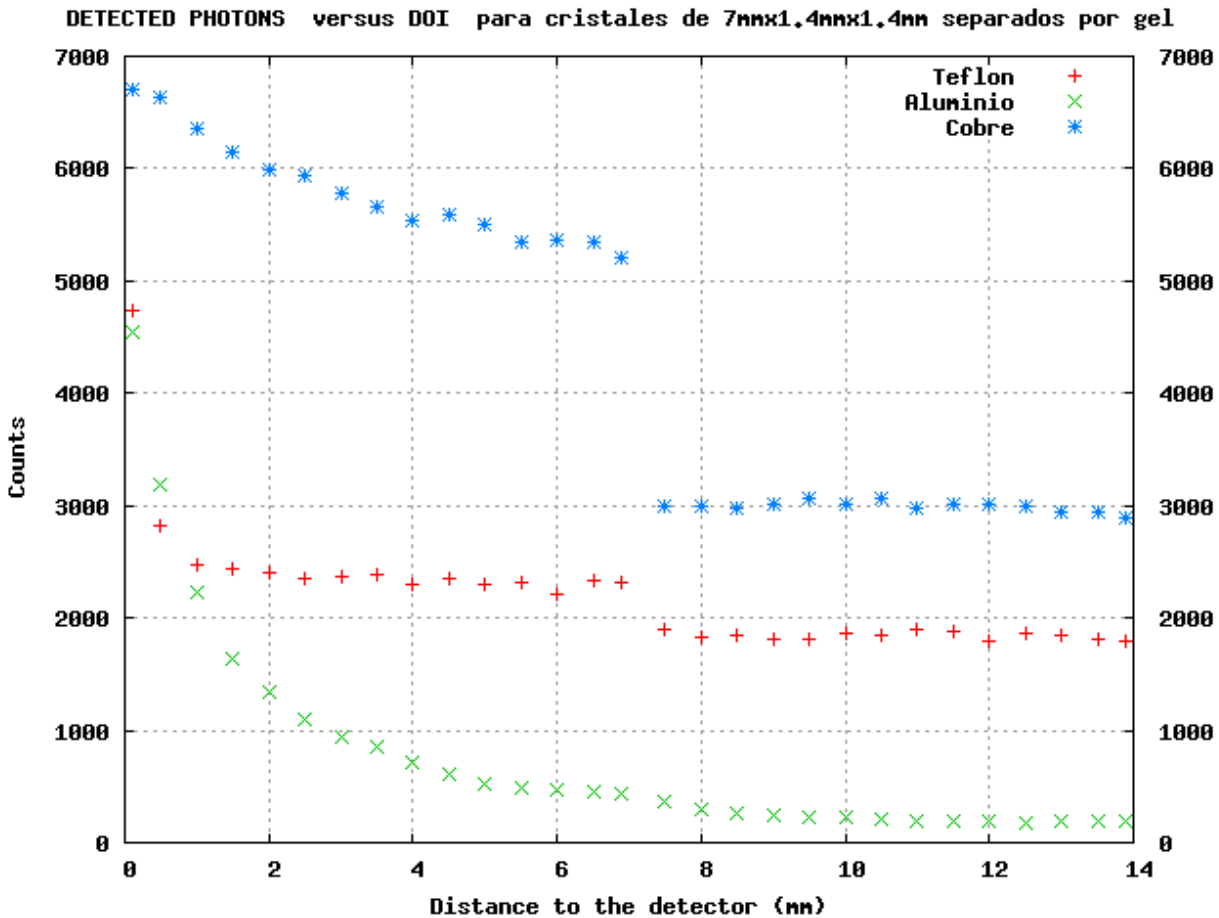
Fotones de 400 nm en cristales de 2cmx1cmx1cm separados por gel



Fotones de 400 nm en cristales de 7mmx1,4mmx7,4mm separados por aire



Fotones de 400 nm en cristales de 7mmx1,4mmx1,4mm separados por gel



Comparación con los resultados experimentales

Procedimiento

Cristales de LYSO de 5cmx1cmx1cm individualmente. La longitud de onda de los fotones que ha estudiado está en torno a los 400 nm, y ha comparado teflón, aluminio y un reflector perfecto. Tomando el número de fotones vistos con un recubrimiento de teflón como la unidad, los resultados son:

Resultados experimentales

Recubrimiento	Fotones vistos respecto al teflón
Totalmente absorbente	0.45
Vicuiti (reflector total)	0.95
Aluminio	0.73
Teflón	1.00

Comparación con los resultados experimentales

Procedimiento

Cristales de LYSO de 5cmx1cmx1cm individualmente. La longitud de onda de los fotones que ha estudiado está en torno a los 400 nm, y ha comparado teflón, aluminio y un reflector perfecto. Tomando el número de fotones vistos con un recubrimiento de teflón como la unidad, los resultados son:

Resultados experimentales

Recubrimiento	Fotones vistos respecto al teflón
Totalmente absorbente	0.45
Vicuiti (reflector total)	0.95
Aluminio	0.73
Teflón	1.00

Simulación de la llegada de fotones gamma a un centellador

- Es posible emplear LITRANI para simular el paso de fotones gamma de entre 0.1 y 1 MeV a través de un material.
- La clase `TPhotoElecCompton` es la encargada de simular la producción de efecto Compton y efecto fotoeléctrico como consecuencia del paso de los fotones gamma.
- Inconveniente principal: el cálculo sobre el que está basada está resuelto para una simetría cilíndrica.
- Simularemos centelladores de NaI dopado con talio, esto es, NaI(Tl) de forma cilíndrica con una pulgada de altura y diámetro. El recubrimiento que emplearemos será aluminio.
- La finalidad de este apartado es mostrar el funcionamiento de esta clase en LITRANI.

Simulación de la llegada de fotones gamma a un centellador

- Es posible emplear LITRANI para simular el paso de fotones gamma de entre 0.1 y 1 MeV a través de un material.
- La clase `TPhotoElecCompton` es la encargada de simular la producción de efecto Compton y efecto fotoeléctrico como consecuencia del paso de los fotones gamma.
- Inconveniente principal: el cálculo sobre el que está basada está resuelto para una simetría cilíndrica.
- Simularemos centelladores de NaI dopado con talio, esto es, NaI(Tl) de forma cilíndrica con una pulgada de altura y diámetro. El recubrimiento que emplearemos será aluminio.
- La finalidad de este apartado es mostrar el funcionamiento de esta clase en LITRANI.

Simulación de la llegada de fotones gamma a un centellador

- Es posible emplear LITRANI para simular el paso de fotones gamma de entre 0.1 y 1 MeV a través de un material.
- La clase `TPhotoElecCompton` es la encargada de simular la producción de efecto Compton y efecto fotoeléctrico como consecuencia del paso de los fotones gamma.
- Inconveniente principal: el cálculo sobre el que está basada está resuelto para una simetría cilíndrica.
- Simularemos centelladores de NaI dopado con talio, esto es, NaI(Tl) de forma cilíndrica con una pulgada de altura y diámetro. El recubrimiento que emplearemos será aluminio.
- La finalidad de este apartado es mostrar el funcionamiento de esta clase en LITRANI.

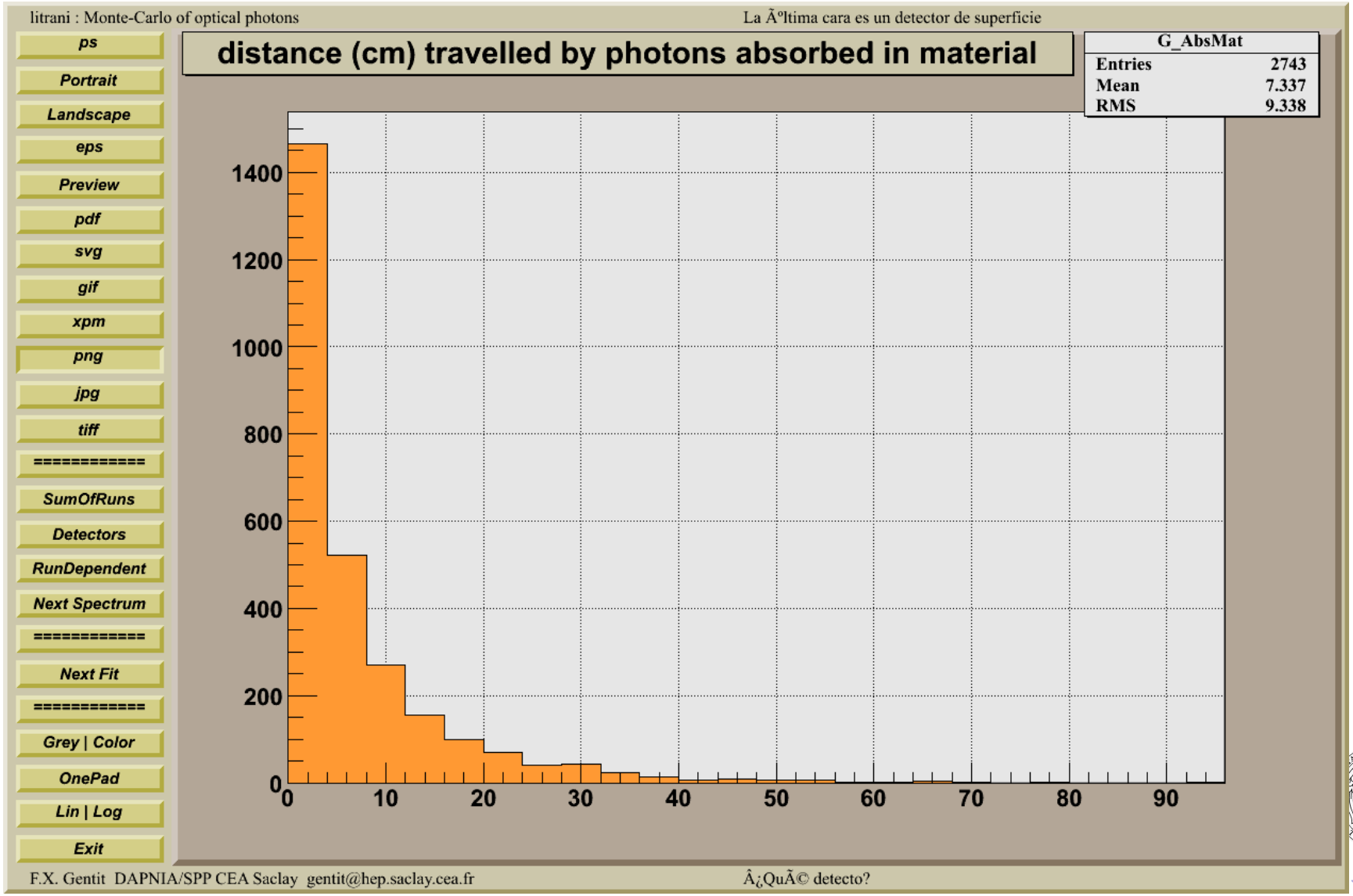
Simulación de la llegada de fotones gamma a un centellador

- Es posible emplear LITRANI para simular el paso de fotones gamma de entre 0.1 y 1 MeV a través de un material.
- La clase `TPhotoElecCompton` es la encargada de simular la producción de efecto Compton y efecto fotoeléctrico como consecuencia del paso de los fotones gamma.
- Inconveniente principal: el cálculo sobre el que está basada está resuelto para una simetría cilíndrica.
- Simularemos centelladores de NaI dopado con talio, esto es, NaI(Tl) de forma cilíndrica con una pulgada de altura y diámetro. El recubrimiento que emplearemos será aluminio.
- La finalidad de este apartado es mostrar el funcionamiento de esta clase en LITRANI.

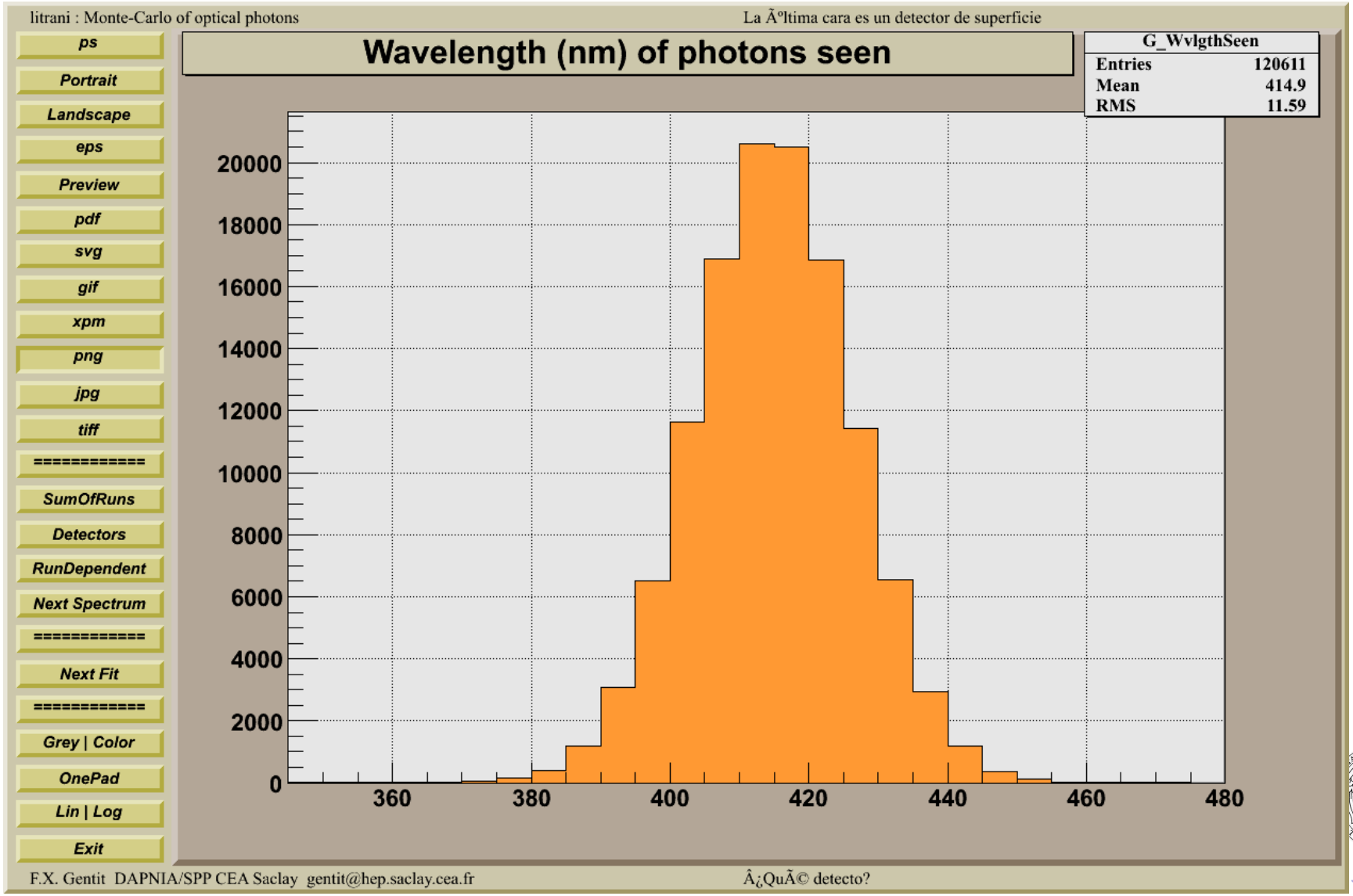
Simulación de la llegada de fotones gamma a un centellador

- Es posible emplear LITRANI para simular el paso de fotones gamma de entre 0.1 y 1 MeV a través de un material.
- La clase `TPhotoElecCompton` es la encargada de simular la producción de efecto Compton y efecto fotoeléctrico como consecuencia del paso de los fotones gamma.
- Inconveniente principal: el cálculo sobre el que está basada está resuelto para una simetría cilíndrica.
- Simularemos centelladores de NaI dopado con talio, esto es, NaI(Tl) de forma cilíndrica con una pulgada de altura y diámetro. El recubrimiento que emplearemos será aluminio.
- La finalidad de este apartado es mostrar el funcionamiento de esta clase en LITRANI.

Resultados


litrani : Monte-Carlo of optical photons
La Ãºltima cara es un detector de superficie
ps
Portrait
Landscape
eps
Preview
pdf
svg
gif
xpm
png
jpg
tiff
==========
SumOfRuns
Detectors
RunDependent
Next Spectrum
==========
Next Fit
==========
Grey | Color
OnePad
Lin | Log
Exit
distance (cm) travelled by photons absorbed in material
G_AbsMat
Entries 2743
Mean 7.337
RMS 9.338
1400
1200
1000
800
600
400
200
0
0 10 20 30 40 50 60 70 80 90
F.X. Gentit DAPNIA/SPP CEA Saclay gentit@hep.saclay.cea.fr
Â¿QuÃ© detecto?

Resultados


litrani : Monte-Carlo of optical photons
La Ãºltima cara es un detector de superficie
ps
Portrait
Landscape
eps
Preview
pdf
svg
gif
xpm
png
jpg
tiff
==========
SumOfRuns
Detectors
RunDependent
Next Spectrum
==========
Next Fit
==========
Grey | Color
OnePad
Lin | Log
Exit
Wavelength (nm) of photons seen
G_WvlgthSeen
Entries 120611
Mean 414.9
RMS 11.59
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
360
380
400
420
440
460
480
F.X. Gentit DAPNIA/SPP CEA Saclay gentit@hep.saclay.cea.fr
Â¿QuÃ© detecto?

Resultados

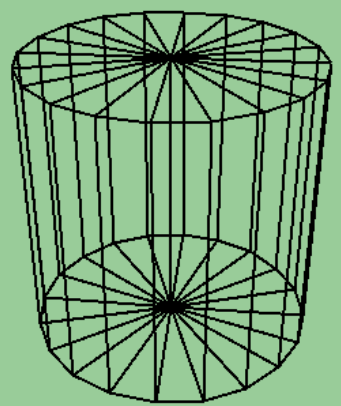
litrani : Monte-Carlo of optical photons

La Ãºltima cara es un detector de superficie

ps
Portrait
Landscape
eps
Preview
pdf
svg
gif
xpm
png
jpg
tiff
==========
SumOfRuns
Detectors
RunDependent
Next Spectrum
==========
Next Fit
==========
Grey | Color
StateOfPad
Lin | Log
Exit

GlobStat

```
Nb. of photons generated   : 184121
Lost for abnormal reasons  : 0
Lost because abs. length 0 : 0
Eff. nb. of gen. photons   : 184121
Nb. of photons seen        : 120611
Efficiency                 :   0.655064
   error                   :  +/-0.00188621
Lost for any reason        : 63510
Lost in materials          : 2743
Lost before wrapping       : 0
Lost in wrapping           : 60767
Lost leaving setup         : 0
Lost because seen too late : 0
Lost b. too few e- in APD  : 0
Lost b. acceptance angle   : 0
Lost b. quantum efficiency : 0
```



F.X. Gentit DAPNIA/SPP CEA Saclay gentit@hep.saclay.cea.fr

Â¿QuÃ© detecto?

Conclusiones

- Hemos aprendido a utiliar un software completamente desconocido para nosotros, hasta llegar a simular montajes de cierta complejidad con él.
- Además, de este aprendizaje hemos podido escribir un pequeño manual de iniciación para los posibles interesados en el manejo de LITRANI, así como una buena cantidad de programas sencillos para entrenarse en su utilización.
- Lamentablemente, los resultados que hemos obtenido en nuestras simulaciones son difícilmente contrastables y, a nuestro parecer, no demasiado fiables. Además, la pequeña comprobación experimental de la bondad de nuestros resultados ha arrojado conclusiones bastante negativas.
- La falta de seguridad con respecto a los datos que hemos utilizado es una de las principales causas de la poca fiabilidad de nuestros resultados.

Conclusiones

- Hemos aprendido a utiliar un software completamente desconocido para nosotros, hasta llegar a simular montajes de cierta complejidad con él.
- Además, de este aprendizaje hemos podido escribir un pequeño manual de iniciación para los posibles interesados en el manejo de LITRANI, así como una buena cantidad de programas sencillos para entrenarse en su utilización.
- Lamentablemente, los resultados que hemos obtenido en nuestras simulaciones son difícilmente contrastables y, a nuestro parecer, no demasiado fiables. Además, la pequeña comprobación experimental de la bondad de nuestros resultados ha arrojado conclusiones bastante negativas.
- La falta de seguridad con respecto a los datos que hemos utilizado es una de las principales causas de la poca fiabilidad de nuestros resultados.

Conclusiones

- Hemos aprendido a utiliar un software completamente desconocido para nosotros, hasta llegar a simular montajes de cierta complejidad con él.
- Además, de este aprendizaje hemos podido escribir un pequeño manual de iniciación para los posibles interesados en el manejo de LITRANI, así como una buena cantidad de programas sencillos para entrenarse en su utilización.
- Lamentablemente, los resultados que hemos obtenido en nuestras simulaciones son difícilmente contrastables y, a nuestro parecer, no demasiado fiables. Además, la pequeña comprobación experimental de la bondad de nuestros resultados ha arrojado conclusiones bastante negativas.
- La falta de seguridad con respecto a los datos que hemos utilizado es una de las principales causas de la poca fiabilidad de nuestros resultados.

Conclusiones

- Hemos aprendido a utiliar un software completamente desconocido para nosotros, hasta llegar a simular montajes de cierta complejidad con él.
- Además, de este aprendizaje hemos podido escribir un pequeño manual de iniciación para los posibles interesados en el manejo de LITRANI, así como una buena cantidad de programas sencillos para entrenarse en su utilización.
- Lamentablemente, los resultados que hemos obtenido en nuestras simulaciones son difícilmente contrastables y, a nuestro parecer, no demasiado fiables. Además, la pequeña comprobación experimental de la bondad de nuestros resultados ha arrojado conclusiones bastante negativas.
- La falta de seguridad con respecto a los datos que hemos utilizado es una de las principales causas de la poca fiabilidad de nuestros resultados.

¡Gracias!

Muchísimas gracias a Pablo García Corzo, Samuel España y a Marta Larranoa por su ayuda a lo largo del trabajo.