

Guión das prácticas no IGFAE

Prácticas optativas de Grao · Curso 22-23
Universidade de Santiago de Compostela

Índice

1. Presentación	1
1.1. Obxectivos	1
1.2. Motivación	1
2. Principios físicos	2
2.1. Un detector gasoso	2
2.2. Perdas de enerxía na materia	2
2.3. Resolución en enerxía	3
3. Para comezar	4
4. Simulación	4
4.1. Xeometría	4
4.2. Cinemática	5
4.3. Perdas de enerxía (SRIM) e resolución	5
5. Algoritmo	7

1. Presentación

1.1. Obxectivos

O obxectivo destas prácticas vai ser reproducir a resposta real do detector *Active TARget and Time Projection Chamber* (ACTAR TPC) nun computador de cara a preparar a campaña experimental de 2025. Simularase unha reacción nuclear de cuxos resultados avaliaremos a resolución na reconstrución de ángulos.

Abarcaremos varias áreas:

- **Programación:** Aprenderemos a programar en C++ con ROOT. Utilizaremos programas sinxelos para que a adaptación sexa o máis liviana posíbel e fomentaremos o recurso á documentación na web.
- **Detectores:** ACTAR TPC é un detector gasoso moi recente que abre toda unha nova ventá de posibilidades no eido das reaccións nucleares. Introduciremos os conceptos básicos dun detector gasoso e complementaremos cos de detectores de Silicio.
- **Teoría:** Expoñeremos as leis básicas da cinemática que rixen calquera reacción entre partículas e introduciremos os fundamentos da interacción radiación-materia, básicos para a detección experimental.

Iremos adaptando o guión das prácticas en función das necesidades e do tempo dispoñíbel. Gran parte do código computacional está xa listo, e o que se requirirá será facer uso del en programas máis simples.

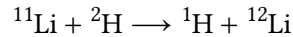
1.2. Motivación

En 2025 levarase a cabo un experimento coa reacción $^{11}\text{Li}(d, p)^{12}\text{Li}$ para estudar a estrutura nuclear do ^{12}Li . En particular, este tipo de reaccións de transferencia dun nucleón permiten acceder de forma pormenorizada á información espectroscópica do núcleo resultante; neste caso, co obxectivo principal de estudar o estado fundamental do ^{12}Li .

Para lograr este obxectivo, necesitamos varias compoñentes:

- **Detector:** O gas que encherá ACTAR TPC será unha mestura de D_2 e iC_4H_{10} (95 % e 5 %, respectivamente) a unha presión de 900 mbar. Será o D_2 o que provea os d da reacción.

- **Reacción:** A xa mencionada:



- **Feixe:** O ${}^{11}\text{Li}$ terá unha enerxía de 7,5 AMeV (é dicir, 82,5 MeV).

2. Principios físicos

2.1. Un detector gasoso

ACTAR TPC segue os principios de funcionamento dunha *time projection chamber*: as partículas ao pasaren polo gas ionizan os átomos, liberando electróns que **derivan** baixo aplicación dun campo eléctrico. Esta carga é recollida nun sensor amplamente fragmentado (), permitindo *seguir* as partículas no seu traspaso dentro do medio en 2D.

Pero é máis: pódese engadir a terceira coordenada coñecendo o tempo que lle leva aos electróns chegar ao *pad plane*, pois a **velocidade de deriva** é constante. Deste xeito, o seguimento das partículas faise en **3D**.

Finalmente, o principio de *active target* engade a vantaxe de ser o gas o propio albo da reacción: este actúa como medio de reacción e de detección. O seguinte esquema ilustra este modo de funcionamento.

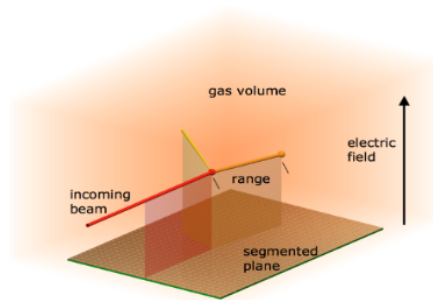


Figura 1: Ilustración do principio de funcionamento dun albo activo.

2.2. Perdas de enerxía na materia

As partículas ao propagarse a través do gas interaccionarán co campo de Coulomb dos átomos deste, perdendo enerxía e liberando electróns. Esta perda de enerxía en cada interacción é moi pequena, pero como é moi elevado o número de colisións no gas (é, polo tanto, un proceso estocástico), chega a ser moi importante. Nunha aproximación continua, está dada pola **ecuación de Bethe-Bloch**:

$$-\frac{dE}{dx} \cong \frac{4\pi e^4}{m_e} \left(\rho \frac{N_A}{M} \right) \frac{z^2 Z}{v^2} \ln \frac{2m_e v^2}{I}$$

Depende esencialmente da partícula a considerar (Z e A) e do gas (a través de ρ). É interesante definir:

- **Rango (R):** É o camiño percorrido por unha partícula ata que *case* se detén. É función da enerxía inicial e do material a atravesar. Existe unha **relación unívoca** $E \iff R$ que imos utilizar para estimar a perda da enerxía en función da distancia percorrida pola partícula.
- **Straggling:** Como a perda de enerxía é un **proceso estatístico**, dúas partículas coa mesma enerxía inicial non van depositar a mesma ΔE . O *straggling* mide, dalgunha forma, a σ desta distribución.

Esta información está tabulada nun programa que se coñece como **SRIM** e que utilizaremos sistematicamente nas prácticas. Basicamente, contén táboas cos parámetros que vamos necesitar para os cálculos: enerxías, rangos e straggling.

Unha pequena apreciación: SRIM vainos dar o *straggling* en posición, mais nós deberémolo converter a en enerxía. Aquí o algoritmo que imos usar:

1. Calculamos R_{ini} coa enerxía inicial, xunto co seu *straggling*.
2. Sabendo que a partícula percorre unha distancia d o rango nese punto será $R_L = R_{ini} - d$. Avaliamos o *stragg.* para este valor.
3. O *straggling* na distancia estará **correlacionado con ambos**, e sabendo que $u^2(R_{ini}) = u^2(R_{Left}) + u^2(d)$, despexamos $u(d)$.
4. Aleatorizamos o valor de d cunha gaussiana centrada no seu valor e de $\sigma = u(d)$. Recalculamos o valor de $R_{Left} \rightarrow R'_{Left}$ coa nova $d \rightarrow d'$.
5. Con este novo rango calculamos a enerxía final, efectivamente implementando o *straggling*!

2.3. Resolución en enerxía

Complementariamente acostuman situarse detectores de Si que nos van permitir medir a enerxía das partículas á súa saída do *pad plane* (seguindo os mesmos principios que na anterior Sección). Un parámetro importante destes é a **resolución en enerxía**, definida como a capacidade para *resolver* enerxías depositadas distintas: $R = \text{FWHM}/E$.

Para o noso experimento, esta resolución está tabulada en 50 keV a 5,5 MeV e podemos extrapolala ao resto de enerxías coa función:

$$R = \frac{2,35\sigma}{E} = \frac{K}{\sqrt{E}} \implies \sigma = \frac{K}{2,35} \sqrt{E} = \frac{0,0213 \text{ MeV}}{2,35} \sqrt{E}$$

Isto vai significar que a perda de enerxía ΔE que calcules nos Si deberala aleatorizar cunha gaussiana de σ obtida coa anterior fórmula.

En decembro de 2022 probamos ACTAR TPC (máis ben o prototipo, a versión *mini*) por primeira vez cun feixe de neutróns, para ver se era viábel utilizar este detector con este tipo de partículas e, idealmente, caracterizar o fluxo de neutróns incidente.

- **Detector:** Utilizouse *proto*ACTAR con isobutano como gas a 140 mbar de presión, e xunto a isto, dúas paredes de detectores de silicio. Estes miden a enerxía residual das partículas ao saír do gas, tal e como se amosa na [Figura 2](#).

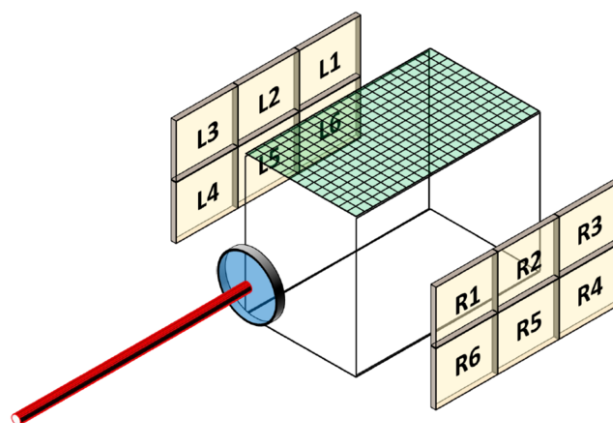
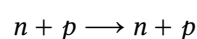


Figura 2: Disposición experimental dos detectores

- **Reacción:** A máis sinxela, a dispersión elástica



Como o gas tamén ten carbono, algunha reacción extra con este núcleo é esperábel.

- **Beam:** Neutróns cun espectro continuo entre 10-30 MeV. Coñecemos o fluxo teórico, pero o ideal sería verificalo con este experimento. Sen unha medida complementaria, a priori non coñecemos a enerxía do neutrón que reaccionou!
- **Análise:** Nun detector gaseoso a reacción ten lugar dentro do gas ao tempo que as partículas, ao propagarse por este medio, ionízano e deixan unha *traza* (electróns que se desprazan grazas a un campo eléctrico que aplicamos nós), que recolle a electrónica do detector en forma de *puntos 3D*. É o noso labor unilos para reconstruír a traza e obter así os ingredientes básicos da análise: ángulo e enerxía no vértice.

Aplicando as ecuacións de cinemática, podemos obter a enerxía que non sabíamos, a do feixe, e estimar posteriormente o fluxo! Pero reconstruír as trazas non é nada sinxelo... Por sorte nestas prácticas non imos facer isto, pero que saibas que constitúe unha das partes esenciais do traballo con estes detectores.

- **Simulacións:** Sempre é necesario reproducir computacionalmente un experimento para verificar que todos os aparellos están funcionando ben, e tamén para complementar a análise (hai partes desta que só se poden facer con datos extraídos de simulacións). É nesta parte na que nos centraremos nestas prácticas, realizando unha simulación moi sinxela, aínda que completa, dos contidos expostos na anterior Sección.

3. Para comezar

O primeiro é configurar o entorno de traballo. Para iso imos necesitar un entorno con GNU/Linux e unha versión de ROOT (<https://root.cern.ch/>) recente. Estas dúas presentacións mostran conceptos básicos de programación que che axudarán:

- C++. A sección 2 desta presentación (*Language basics*) cubre os conceptos básicos necesarios: tipos, referencias, *pointers*, *scopes* e estruturas de control. https://raw.githubusercontent.com/hsf-training/cpluspluscourse/download/talk/C%2B%2BCourse_essentials.pdf
- ROOT. Podes ver a seguinte presentación para ver as potencialidades de ROOT. Céntrate nos histogramas e nos gráficos, o resto non o imos necesitar. Practicaremos o código xa coa simulación. https://docs.google.com/presentation/d/1XUvgJ1IRNO-nIF7M9TErrMcyisLqhhZLB0L-M0AkWUo/edit#slide=id.g13ec28bae0e_0_63
- Tutoriais de ROOT: <https://root.cern/tutorials/>

4. Simulación

4.1. Xeometría

Como vimos na [Figura 2](#), a nosa simulación fai ter que representar fidedignamente a disposición real dos detectores. As trazas que simulemos dentro do gas terán que ser propagadas no espazo para ver se intersecan cos detectores de silicio. Desta forma determinaremos algo tan importante como a **eficiencia xeométrica**.

A xeometría está xa implementada, nunha clase coñecida como **SimGeometry**, que fai realizar a única tarefa que necesitaremos: propagar unha traza desde un *vértice* ata unha capa de silicios, seguindo unha **dirección** de traza. Isto faise co método

```
SimGeometry::PropagateTrackToSiliconArray(const XYZPoint& initPoint,
                                           const XYZVector& direction,
                                           int assemblyIndex,
                                           bool& isMirror,
                                           double& distance,
                                           int& silType,
                                           int& silIndex,
                                           XYZPoint& newPoint,
```

```
bool debug = false);
```

Listing 1: Interface á function de propagación de traza

Configurando o índice do plano de silicios aos que queremos ir (para nós será sempre 0), esta función devólvenos:

- Se está *espellado*: é dicir, se foi á *layer* esquerda (*false*) ou á dereita (*true*).
- A distancia (*distance*) do vértice ao punto de impacto no silicio (*newPoint*); é dicir, a **lonxitude de traza**.
- O número de silicio impactado, seguindo o código da [Figura 2](#). Se non houbo ningún impacto, devólvese o valor -1.
- A opción *debug* permite, se se activa, amosar o funcionamento interno da función.

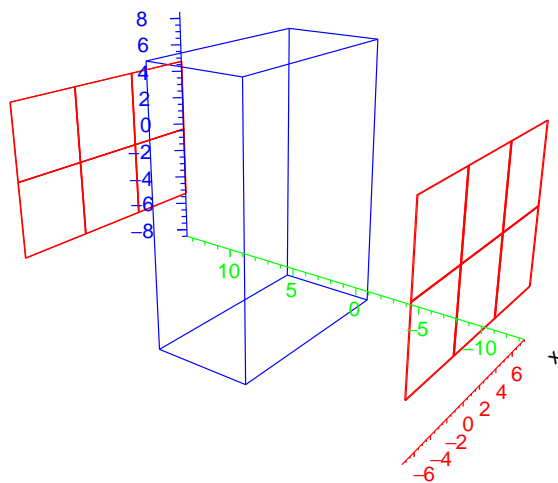


Figura 3: Visualización 3D da xeometría implementada no código

4.2. Cinemática

Aínda que a reacción a implementar aquí é a máis sinxela posíbel, imos resolver o caso máis xeral dunha reacción a dous corpos, asumindo que a partícula pesada final pode ter unha **enerxía de excitación**:

$$m_4 \longrightarrow m_4 + E_x$$

⇒ Aplica as leis de conservación de enerxía e momento para obter a enerxía E_3 e ángulo θ_3 da partícula saínte lixeira en función das variables no **CM**, como son a enerxía total E_{CM} , $m_{3,4}$ e θ_{CM} .

Esta variable θ_{CM} é o único *grao de liberdade* que temos, e que fisicamente está constrinxido pola **sección eficaz** da reacción, que determina a distribución angular dos produtos de reacción. Comezaremos cunha distribución uniforme no intervalo $[-\pi, \pi]$, pero máis tarde veremos como extraer a distribución angular real a partir da sección eficaz diferencial para a reacción de *scattering np*, na [Figura 4](#).

4.3. Perdas de enerxía (SRIM) e resolución

As partículas ao propagarse a través do gas interaccionarán co campo de Coulomb dos átomos deste, perdendo enerxía e liberando electróns. Esta perda de enerxía en cada interacción é moi pequena, pero como é moi elevado o número de átomos no gas, chega a ser moi importante. Nunha aproximación continua, está dada pola ecuación de Bethe-Bloch:

$$-\frac{dE}{dx} \cong \frac{4\pi e^4}{m_e} \left(\rho \frac{N_A}{M} \right) \frac{z^2 Z}{v^2} \ln \frac{2m_e v^2}{I}$$

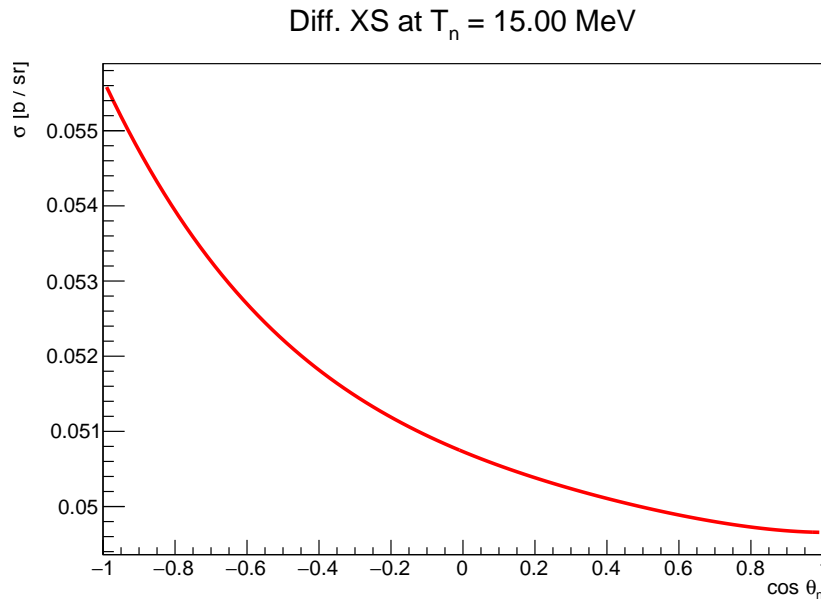


Figura 4: Exemplo real de sección eficaz diferencial para a dispersión np. Actúa como unha distribución de probabilidade dos valores de $\cos \theta$.

E depende esencialmente da partícula a considerar (Z e A) e do gas (esencialmente a través de ρ). É interesante definir:

- **Rango (R):** É o camiño percorrido por unha partícula ata que *case* se detén. É función da enerxía inicial e do material a atravesar. Existe unha **relación unívoca** $E \iff R$ que imos utilizar para estimar a perda da enerxía en función da distancia percorrida pola partícula (obtida coa xeometría!). Toda esta información aparece recollida no programa **SRIM** en forma de táboas.
- **Straggling:** Como a perda de enerxía é un **proceso estatístico**, dúas partículas coa mesma enerxía inicial non van depositar a mesma ΔE . O *straggling* mide, dalgunha forma, a σ desta distribución. SRIM daranos o *straggling* en rango, polo que o deberemos converter a enerxético co seguinte algoritmo:
 1. Calculamos R_{ini} coa enerxía inicial, xunto co seu *straggling*.
 2. Sabendo que a partícula percorre unha distancia d o rango nese punto será $R_L = R_{ini} - d$. Avaliamos o *stragg.* para este valor.
 3. O *straggling* na distancia estará **correlacionado con ambos**, e sabendo que $u^2(R_{ini}) = u^2(R_{Left}) + u^2(d)$, despexamos $u(d)$.
 4. Aleatorizamos o valor de d cunha gaussiana centrada no seu valor e de $\sigma = u(d)$. Recalculamos o valor de $R_{Left} \rightarrow R'_{Left}$ coa nova $d \rightarrow d'$.
 5. Con este novo rango calculamos a enerxía final, efectivamente implementando o *straggling*!

Unha última consideración a ter en conta é a **resolución dos detectores de Si**, definida como a súa capacidade para *resolver* enerxías depositadas distintas, froito tamén do proceso estocástico de perda de enerxía: $R = \text{FWHM}/E$. Sabendo que esta resolución está tabulada en 50 keV a 5,5 MeV podemos extrapolar ao resto de enerxías coa función:

$$R = \frac{2,35\sigma}{E} = \frac{K}{\sqrt{E}} \implies \sigma = \frac{K}{2,35} \sqrt{E} = \frac{0,0213 \text{ MeV}}{2,35} \sqrt{E}$$

A perda de enerxía ΔE que calcules nos Si deberala aleatorizar cunha gaussiana de σ obtida coa anterior fórmula.

5. Algoritmo

De momento imos seleccionar unha única configuración para simular:

- Enerxía do feixe: $T_n = 15$ MeV.
- Presión do gas: $p = 150$ mbar.
- Gas: 100 % de iC_4H_{10} .

Unha síntese dos pasos para realizar a simulación de N eventos é a seguinte:

1. Xerar o **vértice** de reacción. Para iso, asumiremos unha distribución gaussiana en yz, de raio (σ) o do beam, de 2 cm. A compoñente x será uniforme ao longo de toda a lonxitude da cámara. A variable será de tipo `auto vertex = XYZPoint(x, y, z)`.
2. Samplear os ángulos no CM: θ_{CM} e ϕ_{CM} para a partícula 3. ϕ_{CM} seguirá unha distribución uniforme entre $[0, 2\pi]$ e θ_{CM} xerárase a partir doutra uniforme en $\cos\theta_{CM} \in [-1, 1]$.
3. Calcúlase a cinemática no LAB usando o valor de θ_{CM} obtido. Con θ_{Lab} e $\phi_{Lab} = \phi_{CM}$ (xa que se define nunha dirección perpendicular ao *boost*) creamos o vector **dirección da partícula**:

```
auto direction = XYZVector(TMath::Cos(theta),
                           TMath::Sin(theta) * TMath::Sin(phi),
                           TMath::Sin(theta) * TMath::Cos(phi));
```

4. Inicializa as variables que pasas como referencia na [Listing 1](#) e pásaa a dita función, recordando que o *assemblyIndex* vai ser sempre 0. Se a partícula **non impactou** ningún silicio (*silIndex* = -1), ignoramos este evento.
5. Se a traza impactou un silicio, é o momento de aplicar as **perdas de enerxía** usando SRIM, pois o protón terá que atravesar a cantidade de gas (*distance*) ata chegar ao outro detector. Obtén a enerxía do protón ás portas do silicio.
6. Se esta enerxía é 0, esquivamos o evento, pois non o poderemos detectar (dado que requerimos que *impacte* no silicio).
7. Se non, aplicamos SRIM nos Si coñecendo o seu **espesor**, de 500 μm , obtendo ΔE e a enerxía á saída.
8. Crearemos unha bandeira en cada evento para determinar se o protón se parou no Si ou se aínda ten enerxía após el. É o que se coñece como **punchthrough**.

Todo isto farase creando un `TTree` e gardando nun ficheiro as variables máis interesantes para a representación gráfica posterior:

- Os ángulos θ_{CM} e θ_{Lab}
- As enerxías no vértice, ao chegar ao silicio e a ΔE neste.
- A distancia ao Si, o seu índice e se é *left* ou *right* (`std::string side = "left" ou "right";`).
- As *flags*: se impacta ou non no silicio, se chega a este ou se para no gas e se se detén no Si ou aínda posúe enerxía despois. Para isto dáte conta de que debes gardar todos os teus eventos, aínda que non superen os cortes! É útil que en cada iteración re-inicialices todas as variables ao 0. Pode serche útil o tipo `bool flag1 = true; bool flag2 = false;`.

Os histogramas a facer nun segundo programa de análise da simulación serán:

1. Unidimensionais:
 - Os ángulos θ **antes** dos cortes (as *flags*).
 - Os ángulos θ só dos eventos que se **deteñen** no silicio. Para iso, podes *cortar* nun `RDataFrame` sabendo que:

```
//Partindo dun RDataFrame df
auto hThetaCMCut = df.Filter("impactaSil == true && chegaAoSil == true
                             && paraseNoSil == true")
                             .Histo1D("nome da columna");
```

O importante é que representes só os eventos que se paran no Si (pero que chega a el, claro).

- A **eficiencia** angular no CM, sabendo que é o cociente entre os dous histogramas anteriores. Podes facelo como:

```
//Sera hThetaCM os ángulos xerados e hThetaCMCut o mesmo
//pero tras superar os cortes
auto* hThetaCMEff {(TH1D*)hThetaCM->Clone("hThetaCMEff")};
hThetaCMClone->Divide(hThetaCMCut);
//Clonamos = copiamos o primeiro histograma para evitar modificalo e
//ter un histograma co mesmo binning
```

- O número de silicio impactado (idealmente dous histogramas, un para a dereita e outro para a esquerda).

2. Bidimensionais:

- Os planos cinemáticos dos eventos que medimos tanto no CM como no Lab: E_3 vs θ_{CM} e θ_{Lab} . Representa tamén θ_{Lab} vs θ_{CM} .