

Memoria prácticas externa: simulacion ACTAR TPC

Daniel Vázquez Lago

18 de julio de 2024

Índice

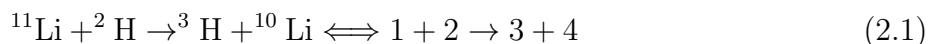
1. Información de las prácticas	3
2. Memoria de Actividades	3
2.1. Objetivos	3
2.2. Trabajo Realizado	3
2.2.1. Cálculo de la cinemática	3
2.2.2. Introducción a C++	4
2.2.3. Implementación de la simulación	4
2.3. Resultados	4
3. Conexión con el grado en físico	4
4. Valoración personal	4

1. Información de las prácticas

2. Memoria de Actividades

2.1. Objetivos

El objetivo principal de estas prácticas es reproducir la respuesta real del detector ACTAR TPC para una reacción nuclear usando el lenguaje de programación C++ y apoyándose en la herramienta ROOT diseñada por el CERN. La reacción nuclear estudiada es la siguiente:



El motivo por el cual esta reacción es de interés para la física nuclear moderna es que nos permitirá comprender mejor la estructura atómica del litio 10, un isótopo muy inestable. Conocer con rigor la estructura para isótopos inestables es fundamental en la física nuclear, ya que nos permitirá crear mejores modelos para las “posiciones” cuánticas de los protones y neutrones dentro del núcleo.

La estructura atómica podrá hallarse en función del ángulo de salida de las partículas pesadas. Consecuentemente es de vital importancia no solo tener los mejores detectores, si no poder predecir cuales cual es la mejor estructura para el detector (tamaño, grosor de los detectores de silicio...), de tal manera que podamos hallar la energía y ángulo de las partículas resultantes de la colisión con la mayor precisión y eficiencia posible.

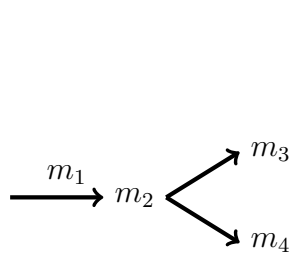
2.2. Trabajo Realizado

2.2.1. Cálculo de la cinemática

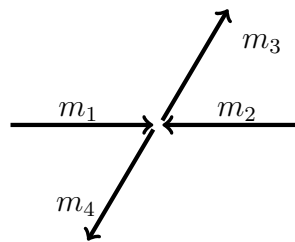
La primera tarea realizada fue calcular la cinemática de las partículas. Una buena aproximación de la colisión se puede hacer suponiendo una colisión elástica dentro de la relatividad especial, ya que debido a las altas energías y la baja masa de las partículas, usar la relatividad especial se hace obligatorio. Una mejor sería suponer que existe una energía perdida debido a la excitación de protones/neutrones en el núcleo de los átomos.

Resolver la cinemática de la colisión consiste en calcular las energías de las diferentes partículas. El sistema de referencia de más interés es el nuestro, que llamamos laboratorio (figura 2.1a), y es aquel en el que la partícula del litio 11 incide con una energía T_{beam} sobre la partícula de deuterio que suponemos quieta. El otro sistema de referencia de interés es el sistema centro de masas (figura 2.1b).

Tras un cálculo podemos obtener la energía cinética de las partículas y su ángulo de salida (respecto al eje de incidencia) en el sistema laboratorio a partir de dos parámetros: el ángulo de salida de la partícula en el sistema de centro de masas, y de T_{beam} . Dado que este último es un valor conocido, podríamos obtener θ_4 y θ_3 a partir de T_3 y T_4 .



(a) Sistema Laboratorio



(b) Sistema Centro de Masas

2.2.2. Introducción a C++

Nuestro segundo paso fue aprender la sintaxis básica de C++ y ROOT. Para esto realizamos algunos programas básicos, que consistían en realizar histogramas y algunas gráficas; así como la definición de variables y de funciones.

2.2.3. Implementación de la simulación

Una vez acabamos de introducirnos a C++ empezamos a implementar la simulación. Lo primero que hicimos fue comprobar si los cálculos de la cinemática realizados eran correctos. Para la implementación de estas ecuaciones se crearon diferentes clases, modularizando el programa principal, de tal modo que tuvimos bien organizadas las diferentes funciones. Dado que no conocemos el ángulo con el que van a salir las partículas, creamos un bucle for para recorrer todos los posibles ángulos. Así podemos ver que:

Una vez comprobamos que la cinemática se calculó correctamente, podemos comenzar a construir la simulación. Esto implica implementar a la colisión diferentes partes. La primera de ellas son las pérdidas de energía de las partículas pesadas en el gas de deuterio.

2.3. Resultados

3. Conexión con el grado en físico

4. Valoración personal