Potenciales en Python

(Práctica 1)

Estudiante:

Víctor Mira Ramírez

Profesor:

J. Fernández-Rossier y Mar Ferri Cortés

Universidad de Alicante
Facultad de Ciencias: Departamento de Física Aplicada
Mecánica Cuántica 1

Resumen

El objetivo de esta práctica es aprender a resolver la ecuación de Schrödinger en una dimensión para potenciales arbitrarios de forma numérica usando *Python*. Para ello se estudiaran dos problemas propuestos y se discutirá brevemente la física que les concierne: el efecto túnel y túnel resonante, así como los estados ligados de potenciales en una dimensión.

Índice

Introducción

Práctica realizada durante el curso 23-24 para la asignatura de Mecánica Cuántica I de la Universidad de Alicante. Clases de prácticas impartidas por J. Fernández-Rossier y Mar Ferri Cortés. Mi DNI es 74528754Z, cuyas tres últimas cifras son 754, que será relevante para las cuestiones que se discuten más adelante.

Es importante para el funcionamiento adecuado de los códigos proporcionados en este informe se cambie el nombre del archivo proporcionado en clase a 'PRAC12023.py', ya que el que viene por defecto al descargarlo no puede ser leído directamente por *python*. A lo largo del informe usamos como valores iniciales: $xmax = 50$, $m = 1$, $N = 150$.

1. Primer Problema (Modelo 1)

1.1. Apartado A

Ejercicio:

Encuentre el valor de la diferencia de energía, $\Delta \equiv E_1 - E_0$, para $p = 0$. (Observa que esta cantidad dependerá ligeramente de los valores de N y $xmax$ elegidos para los cálculos).

El primer paso es definir nuestra función de potencial, que siguiendo los parámentros del guión y mi DNI será:

$$V(x) = \frac{p}{2}x^2 - 10x \tag{1}$$

Ahora, nos haremos valer de la función *ham* proporcionada, así como de la función *eigh* de la librería *scipy.linalg* para crear una función *DeltaRespectoP* que nos devuelva la diferencia de energía entre el estado $E0$ y el estado $E1$ proporcionando un p a la función.

Ejecutando *DeltaRespectoP(0)* obtenemos el valor deseado, que aparece impreso por consola una vez ejecutado.

$$\Delta \equiv E_1 - E_0 = 126{,}697 \ meV \qquad \text{para } p = 0 \tag{2}$$

1.2. Apartado B

Ejercicio:

Encuentra el valor del parámetro p en el potencial $V(x)$ que te corresponde, tal que el cambio de diferencia de energía entre el estado fundamental E_0 y el primer excitado es E_1 con respecto al obtenido en el apartado anterior, $\Delta(p) - \Delta(0)$, sea igual tres últimas cifras de tu DNI en meV. Por ejemplo, DNI terminado en 437, $\Delta(p) - \Delta(0) = 437meV$.

Para la resolución de este apartado crearemos una función *BuscaP* que se hará valer de la función del apartado anterior *DeltaRespectoP* para encontrar el valor de p que haga que $\Delta(p)-\Delta(0) = 745\ meV$

$$p = 75{,}641\ \ \pm 0{,}009\ {}^{meV}/_{A^2} \tag{3}$$

1.3. Apartados C y D

Ejercicio:

(FIGURA 1) Dibuja la curva $\Delta(p)-\Delta(0)$ como función del parámetro variable de tu potencial, y haz visible la solución del problema en dicho punto.

Ejercicio:

(FIGURA 2) Dibuja las funciones de onda de los estados para el valor p obtenido.

Nos hacemos valer del valor de p obtenido en el apartado anterior así como de la función *plotstates* del código de la práctica.

FIGURA 1: Δ(p) − Δ(0) en función de P

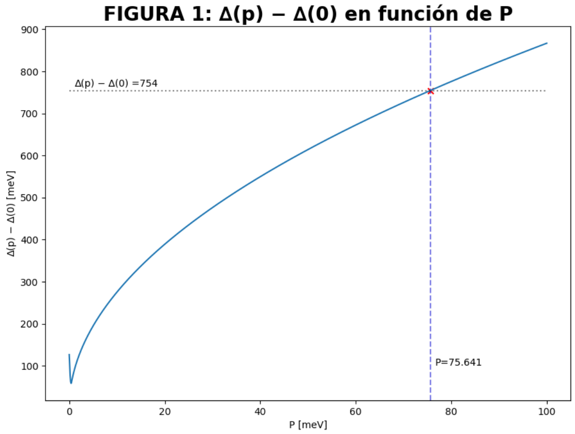
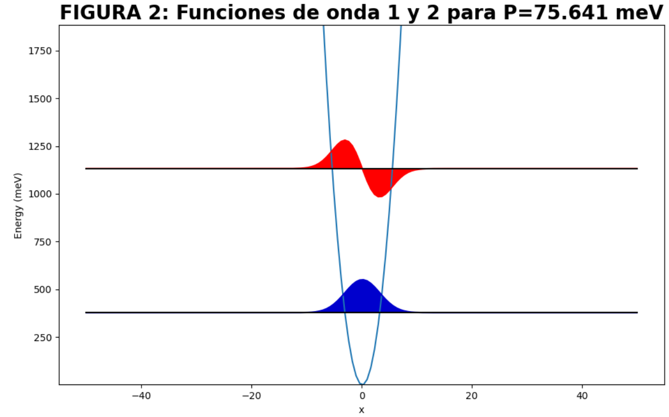


FIGURA 2: Funciones de onda 1 y 2 para P=75.641 meV



2. Segundo Problema (Modelo 2)

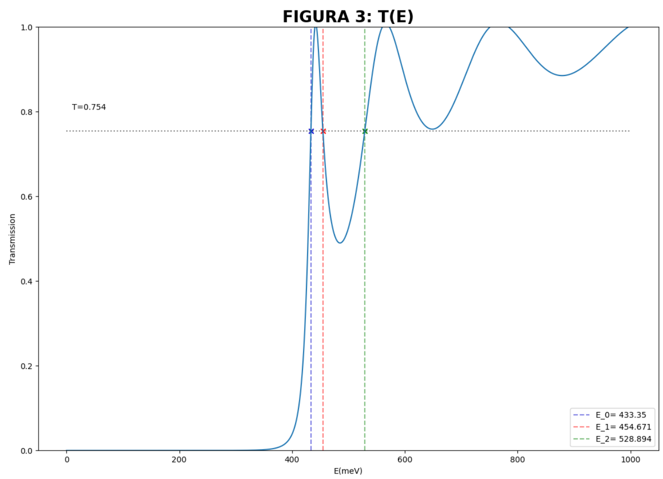
2.1. Apartados A y B

Ejercicio:

Determina la energía del electrón para que la probabilidad de transmisión T sea igual a las 3 últimas cifras de tu DNI, divididas por 1000 (Por ejemplo, para $DNI = 63290281, T = 0{,}281$). Nota que puede haber más de una energía que satisface la condición requerida.

Ejercicio:

(FIGURA 3) Grafica la curva $T(E)$ correspondiente en la que se vea la solución del apartado anterior.



Como en el problema anterior, lo primero que hacemos es definir nuestra función de potencial, haciendo uso de *doublebarrier* del código de la práctica. Ahora, crearemos una función *buscaE* que dada una transmisión objetivo y una precisión, calcula para qué energía (o energías) se da dicha precisión.

La función va calculando la transmisión haciendo uso de la función *trans* proporcionada por el guión de la práctica, y cuando la diferencia entre la transmisión calculada y la objetivo es menor a una tolerancia (por defecto de 10^{-4}) guarda dicho valor en una lista. Este método hace que para un mismo punto de corte podamos tener múltiples valores de energía que cumplan la condición de tolerancia, por lo que nos quedamos con el que menor error genere.

Finalmente el programa imprime por consola los valores de las energías que cumplen que $T = 0{,}754$:

$$E_0 = 433{,}350\ meV \qquad E_1 = 454{,}671\ meV \qquad E_2 = 528{,}894\ meV \tag{4}$$

Haremos uso de la función *plotT* para graficar la transmisión, añadiendo los datos obtenidos y obteniendo la figura deseada. Puede parecer debido a la baja resolución que hay un cuarto corte en la figura, pero con suficiente precisión no se da el caso.

3. Anexos

Hipervínculo al código de python que hace los cálculos y genera las gráficas:

MC1-P1.py

Hipervínculo al código de LaTeX que genera este documento:

MC1 - Práctica 1: Potenciales en Python.tex