

## Tarea 3

Brian Tapia Contreras <sup>†</sup>

### Pregunta 1

- (a) En primer lugar, se configuró el formato del archivo, de tal manera que en la primera línea el archivo almacena  $N$ , el número de cargas que formaran la distribución, y en las siguientes  $N$  líneas muestra el detalle de las cargas, escribiendo la carga  $Q$ , y la posición en  $x$  e  $y$  respectivamente. Para los incisos (b) y (c) se ocuparon inputs distintos, pero el código no se diferenció. En el mismo código se muestran los archivos necesarios para ejecutar cada uno de los incisos.

Teniendo listo el formato de archivo, se creó el programa **P1/P1A.f90**. En las primeras líneas del archivo, además de definir las variables a utilizar, se lee el input correspondiente a cada inciso. Para ello, primero lee la primera línea, definiendo  $N$ , y luego itera  $N$  veces en un ciclo DO, leyendo y almacenando la información de  $Q, x, y$  en vectores del mismo nombre.

Se definió un parámetro  $h$  correspondiente a la distancia entre nodos en la discretización de una grilla que resultará de tamaño  $N_x \times N_y$ . En este caso,  $N_x, N_y$  se definieron iguales y quedan sólo en función de un  $r_{max}$  a definir, que representará la mitad de la longitud de cada lado de la grilla. En esta ocasión usé  $r_{max} = 40$ , por lo que el espacio bidimensional que representa la grilla corresponderá a un intervalo de  $[-40, 40]$  tanto en  $x$  como en  $y$ .

Una vez definidos estos parámetros, se itera sobre la grilla, llamando en cada punto  $(x_i, y_j)$  a una subrutina llamada *efield\_and\_pot*. Esta subrutina calcula las componentes  $E_x$  y  $E_y$  del campo eléctrico en un punto dado, acorde a las discretizaciones mostradas en las siguientes ecuaciones:

$$E_x(x, y) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^N \frac{Q_i(x - x_i)}{((x - x_i)^2 + (y - y_i)^2)^{3/2}} \quad (1)$$

$$E_y(x, y) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^N \frac{Q_i(y - y_i)}{((x - x_i)^2 + (y - y_i)^2)^{3/2}} \quad (2)$$

$$V(x, y) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^N \frac{Q_i}{((x - x_i)^2 + (y - y_i)^2)^{1/2}} \quad (3)$$

En este caso, la suma sobre  $i$  representa a cada carga, con  $Q_i, x_i, y_i$  los atributos de dicha carga leídos desde el input. Los pares ordendos  $(x_i, y_j)$  de la grilla serían los valores de  $x, y$  en las ecuaciones anteriores, por lo que obtendremos la expresión del campo eléctrico y el potencial en

---

<sup>†</sup>brian.tapia@uc.cl

cada punto de la grilla.

Al terminar de ejecutarse la subrutina, se guardan directamente los datos del potencial en un archivo *potencial.dat*. Las componentes del campo se guardan en un archivo *efield.dat*, sometidos a una normalización dada por  $E = \sqrt{E_x^2 + E_y^2}$ .

- (b) Se consideraron 6 partículas cargadas con carga  $q = 10^{-9}C$ , distribuidas como indica la Figura 1. La información específica de posiciones y cargas se encuentra en el archivo *input.dat*, archivo que será leído por el programa descrito en el inciso (a).

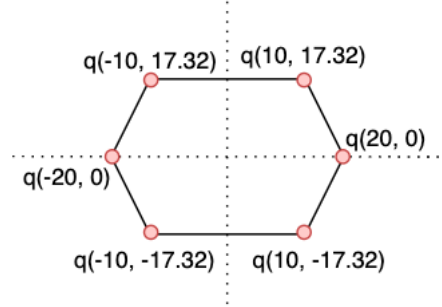


Figura 1: Distribución de las partículas en el espacio  $xy$ .

Se ejecutó el código a fin de obtener los datos necesarios para graficar el potencial y las líneas del campo eléctrico. Los resultados se muestran en la figura a continuación.

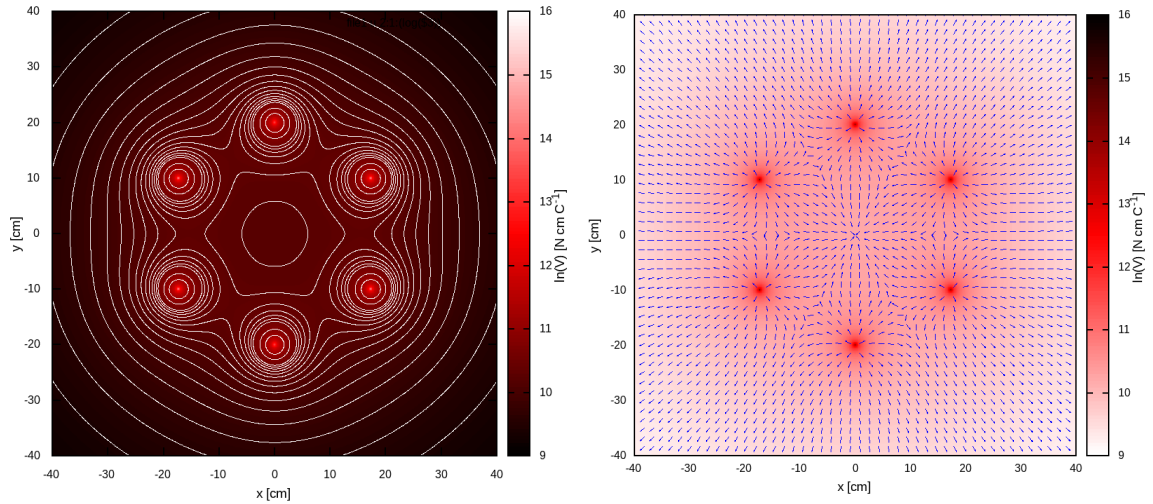


Figura 2: Logaritmo natural del potencial para cada punto del espacio en presencia de 6 cargas idénticas. Se denotan las líneas equipotenciales (fig. izq.) y las líneas del campo eléctrico (fig. der.).

Para graficarlos, se utilizó GNUPlot de manera no trivial. La figura a la izquierda, que muestra el potencial y las respectivas equipotenciales, se graficó utilizando el código **P1/plotA.gnu**. En él se utilizan varios parámetros que permiten la visualización de las líneas de contorno, y

se encuentra el detalle de los valores de estas líneas, arbitrarios y expresados como factores del potencial mínimo. En cuanto a la figura de la derecha, se utilizó un plot *with vectors*, que permite graficar un vector con las componentes normalizadas  $E_x, E_y$  en cada punto de la grilla. Se filtraron simétricamente los datos a fin de no saturar la vista en el plot. El detalle se encuentra en **P1/PlotC.gnu**. En ambos, la escala en el gradiente de color es la misma, que corresponde al logaritmo natural del potencial  $V$ , pero se invirtieron los colores para facilitar la visualización en cada caso.

Respecto a la física del problema: El potencial es esperado para una distribución de cargas iguales como la tenemos. Es pronunciado cerca de cada carga, pudiendo diferenciar perfectamente cuál es la carga que produce el potencial en su vecindad, sin embargo, al alejarnos de las cargas las curvas equipotenciales comienzan a ser más regulares, y en dicho caso es imposible definir qué carga tiene una mayor aportación. En este sentido, a una distancia muy grande sólo sentiríamos el potencial debido a una carga puntual con carga equivalente a la suma de las cargas individuales, como es esperado.

En cuanto a las líneas del campo eléctrico, notamos que son radiales y divergen desde cada una de las cargas. Entre cargas, notamos que hay una tendencia al equilibrio en los ejes de simetría del sistema, pero por supuesto la componente dada por las demás cargas tiene un efecto importante y hace que el campo en ese punto no se anule. El único punto completo de equilibrio sería el centro del hexágono, donde se recibirían aportaciones iguales de todas las cargas pero en direcciones opuestas. Todo esto hace sentido para la distribución de cargas que consideramos.

- (c) Ocupando la misma distribución espacial de cargas mostrada en la Figura 1, pero alternando los valores de las cargas, se obtuvieron los siguientes gráficos mostrando líneas equipotenciales y líneas de campo.

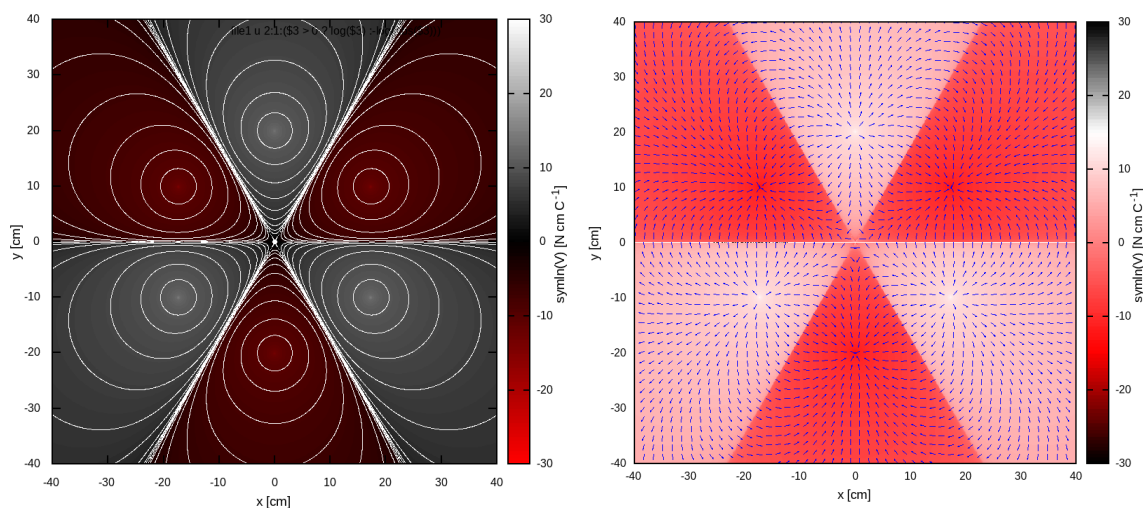


Figura 3: Logaritmo natural del potencial para cada punto del espacio en presencia de 6 cargas de mismo módulo y signo alternado. Se denotan las líneas equipotenciales (fig. izq.) y las líneas del campo eléctrico (fig. der.).

El gráfico de la derecha nos revela un complejo sistema de curvas equipotenciales, donde podemos apreciar seis regiones importantes (separadas por tres rectas equipotenciales que se extienden hacia el infinito), en cada una de estas regiones se encuentra una de las partículas cargadas, la cual tiene mayor relevancia en esta zona. En el centro del plano, en la posición (0,0), podemos encontrar un punto de equilibrio inestable que se vuelve muy oscuro, lo que nos muestra que la energía potencial en este punto es 0.

Finalmente, notamos que este sistema nos permite orbitas cerradas, pero que se mantienen en la región gobernada por una de las seis partículas cargadas, lo que nos señala que no es posible orbitar en torno a las seis partículas al mismo tiempo, ya que no hay curvas equipotenciales que encierren toda la figura.

Del análisis vectorial del campo eléctrico que nos proporciona el segundo gráfico podemos corroborar el punto de equilibrio inestable en el centro del plano y además notar que el campo eléctrico siempre cruza perpendicular en las rectas que separan una región de otra, dotándolas como intuimos antes de la característica de curva de nivel.

## Pregunta 2

Para resolver este problema, se tomó como punto de partida el código **10\_Poisson.f90** mostrado en clases.

En el sistema de conductores con el que se está trabajando, el potencial eléctrico en la región intermedia debe satisfacer la ecuación de Laplace:

$$\nabla^2 V = 0 \Rightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 \quad (4)$$

Ecuación que, de manera discreta, adopta la siguiente forma:

$$\nabla^2 V = 0 \Rightarrow V_{i,j} = \frac{1}{4}(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1}) \quad (5)$$

donde  $i, j$  representan los puntos de la grilla.

Dado que los conductores son cuadrados, tomaremos esta grilla como una grilla cuadrada, de tamaño  $N \times N$ .

Para los programas que resolverán los casos en los incisos siguientes, se crearon tres subrutinas genéricas:

- **red\_inicial:** Esta subrutina, que recibe el tamaño de la malla, las condiciones de borde y las variables que almacenarán la información, tiene como primer objetivo escribir la malla inicial, incluyendo únicamente los valores de las condiciones de borde. Esto se almacenará en la matriz  $V$ , de igual tamaño que la grilla. Adicionalmente se definirá la matriz *es\_conductor*, de igual tamaño pero con valores lógicos. Esta indicará si en el punto hay un conductor o no.
- **Poisson:** Recibe las dos matrices resultantes de la subrutina anterior,  $N$  y una precisión. En esencia, esta subrutina resuelve la ecuación de Laplace según lo mostrado en la Ecuación 5 (no es más que una ecuación de Poisson con  $f = 0$ ). La iteración ocurre para los puntos fuera del borde y que no son conductores. El método iterativo

permite resolver la ecuación en cada punto a partir de las condiciones iniciales. Al finalizar los ciclos, el programa verifica si la precisión se alcanza, y de ser así termina la ejecución.

- **guardar:** Simplemente guarda los datos obtenidos en un archivo, según el formato:  $x, y, V(x, y)$
- **Q:** Se utilizará en los incisos (b) y (c). Su intención es calcular las cargas en ambos conductores. Para ello, se iterará sobre estos buscando el punto más cercano, que siempre será el punto siguiente o anterior (no conductor), a excepción de aquellas cargas que estén en las esquinas. Tomando la diferencia de potencial entre ambos puntos y el tamaño  $h$  de la distancia entre nodos, podemos estimar  $E_n$ , para luego calcular la densidad superficial de carga e integrar para obtener la carga total.

Explicado esto, procedo con el problema.

- (a) Se definió  $L_1$  como un vector de largo 4, y se definieron los valores de las condiciones de borde en los conductores. La grilla a considerar será con  $N = 250$ , y precisión  $\varepsilon = 1 \times 10^{-6}$ .

Se llamó a las primeras dos subrutinas, que no dependen aún del valor de  $L_1$ , por lo que serán ejecutadas solo una vez.

Luego, se hizo un ciclo DO para recorrer el array  $L_1$ . En cada iteración, definimos el paso como  $h = L_1/N$  y ejecutamos la subrutina **guardar**. En este caso, además de sólo guardar los datos, esta subrutina realizará un mapeo que toma a la grilla y la lleva a un espacio bidimensional según el parámetro  $h$ .

Los resultados para el mínimo y máximo valor de  $L_1$  se observan en la Figura ?? . Los otros valores mantienen la misma forma, pero varían en el rango del espacio bidimensional.

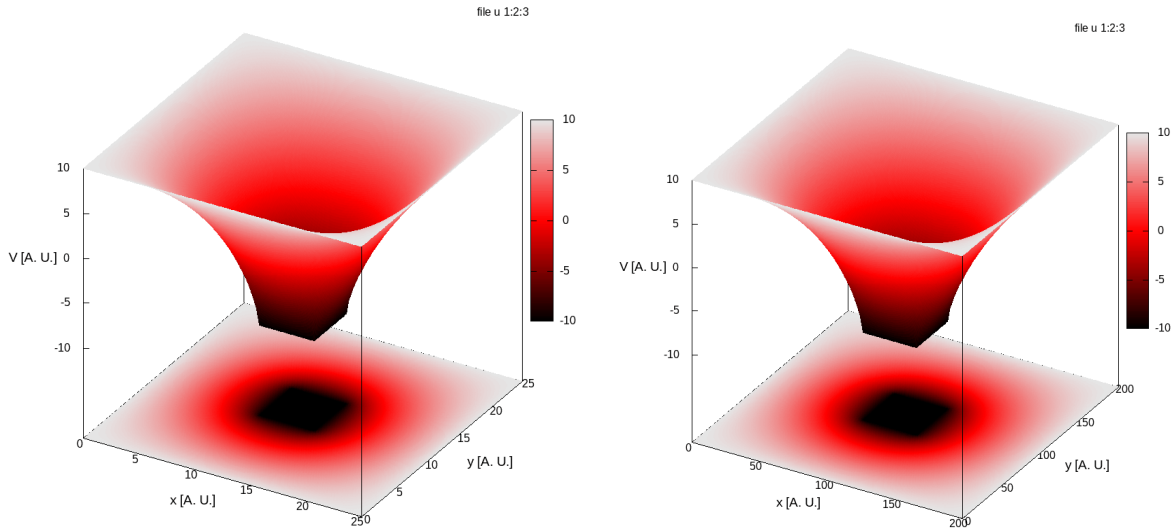


Figura 4: Logaritmo natural del potencial para cada punto del espacio en presencia de 6 cargas de mismo módulo y signo alternado. Se denotan las líneas equipotenciales (fig. izq.) y las líneas del campo eléctrico (fig. der.).

Todo el desarrollo se encuentra en los archivos **P2/P2A.f90** y **P2/plot2A.gnu**.

- (b) Dentro del mismo código, inmediatamente después de llamar a la subrutina **guardar**, se llama a la subrutina **Q**. Para cada valor de  $L_1$  se imprime en pantalla el valor de la carga en ambos conductores y el valor estimado de la capacitancia. Los resultados se resumen en una tabla a continuación.

$L_1$	$Q_-$	$Q_+$	$C$
25	-6.589671	6.301485	0.322278
50	-6.589671	6.301485	0.322278
100	-6.589671	6.301485	0.322278
200	-6.589671	6.301485	0.322278

Cuadro 1: Cantidades para distintos valores de  $L_1$ .  $Q_+$  representa la carga resultante positiva y  $Q_-$  la negativa.  $C$  es la capacitancia.

Notamos que los valores dan, con muchísima precisión, exactamente iguales para todo  $L_1$ . Esto no es sorpresa, debido a que en esencia es exactamente la misma situación mapeada de manera distinta. Si bien el parámetro  $h$ , que depende directamente de  $L_1$ , es distinto en todos los casos, este ve anulado su efecto cuando integramos la densidad de carga.

Se observa que los módulos de  $Q_+$  y  $Q_-$  no son iguales. Esto se puede deber al modelamiento de las esquinas en los conductores, que son puntos conflictivos en un cálculo discreto. Por ejemplo, para el conductor interno, existen dos puntos que satisfacen ser el más cercano, y la decisión fue tomada de manera arbitraria en este trabajo. Por otro lado, en el conductor externo, los únicos puntos vecinos son conductores, y el punto más cercano se encuentra a una distancia mayor que en las otras partes del conductor.

Dado que la igualdad en módulo de estas cargas era algo que suponíamos en el enunciado, esta diferencia puede ocasionar una propagación de error en el cálculo de la capacitancia. Esta última fue calculada como el promedio simple entre los valores absolutos de ambas cargas.

- (c) Se modificó ligeramente el código anterior, a fin de mantener un  $L_1$  constante, mantener también un  $V_2 = 5$  constante e iterar para distintos valores de  $V_1$ , tal que  $\Delta V = V_1 - V_2$  adopte los valores pedidos. El código final está en **P2/P2C.f90**.

De igual manera que en el inciso anterior se escribieron los valores de las cargas y capacitancia para los distintos  $\Delta V$ . Los resultados se muestran en la siguiente tabla.

$\Delta V$	$Q_-$	$Q_+$	$C$
1	-0.332924	0.313557	0.323240
2	-0.662224	0.628712	0.322734
5	-1.650126	1.574177	0.322430
10	-3.296627	3.149952	0.322329
15	-4.943128	4.725728	0.322295
20	-6.589671	6.301485	0.322278
25	-8.236131	7.877278	0.322268

Cuadro 2: Cantidades para distintos valores de  $\Delta V$ .  $Q_+$  representa la carga resultante positiva y  $Q_-$  la negativa.  $C$  es la capacitancia.

Se observa qué, efectivamente, la capacitancia se mantiene constante ante una variación en la diferencia de potencial. Observamos que esto se cumple por lo menos hasta el cuarto decimal.

En el inciso anterior era comentada la diferencia entre los valores absolutos de las cargas en los dos conductores. En esta oportunidad vemos qué, además de existir sistemáticamente esta diferencia, el error es mayor cuando trabajamos con mayores diferencias de voltaje. Esto se puede deber a una propagación en el cálculo inicial, como era predicho en la discusión anterior.

### Pregunta 3

Se tiene la ecuación para la distribución de la temperatura:

$$\frac{\partial u}{\partial t} = D(x) \frac{\partial^2 u}{\partial x^2} - \frac{4}{L} \frac{\partial u}{\partial x} \quad (6)$$

con condiciones de borde  $u(0, t) = u(L, t) = 0$ ,  $D(x) = ae^{-4x/L}$  y condición inicial conocida

$$u(x, 0) = Ce^{-(x-L/2)^2/\sigma^2} \quad (7)$$

A partir de esto, podemos escribir un modelo discreto para afrontar el problema computacionalmente. Consideramos una grilla de elementos discretos  $(i, j)$ , los nodos  $i$  representarán la discretización en la posición, mientras que los nodos  $j$  en el tiempo. La distancia entre dos nodos consecutivos la llamaremos  $\Delta x$  en el caso de distancia, y  $\Delta t$  en el caso del tiempo. Comenzamos por discretizar la Ecuación 6. Sea  $u_{i,j}$  el valor de  $u$  en un punto  $(i, j)$  cualquiera, y  $D_i$  el valor de  $D(x)$  en el nodo  $i$ , discretizamos las derivadas:

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = D_i \left[ \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} \right] - \frac{4D_i}{L} \left[ \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \right] \quad (8)$$

Notamos qué, por como está formulada la ecuación, temporalmente ningún término en  $u$  depende de  $j - 1$ . Nos aprovecharemos de esto para construir, a partir de las condiciones iniciales en  $t = 0$ , un método iterativo que tome elementos en  $j$  y a partir de ellos calcule el elemento en  $j + 1$ . Con este propósito, despejamos  $u_{i,j+1}$  en la ecuación.

$$u_{i,j+1} = u_{i,j} + D_i \left[ \frac{\Delta t}{\Delta x^2} (u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{\Delta t}{\Delta x} \frac{4}{L} (u_{i+1,j} - u_{i,j}) \right] \quad (9)$$

Esta será la ecuación que resolverá el programa escrito en **P3/P3ALT\_B.f90**. El método está basado en el libro *Computational Physics* de Konstantinos Anagnostopoulos. En este libro se hace énfasis en el parámetro  $\Delta t/\Delta x^2$ , llamado parámetro de Courant. Este parámetro determina la escala de evolución temporal en el algoritmo, y, para que exista una evolución estable, se debe cumplir la condición dada en la Ecuación 10.

$$\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2} \quad (10)$$

Dicho esto, y teniendo la ecuación discretizada, continuamos con el ejercicio.

- (a) El programa comenzará preguntando al usuario el valor de las constantes  $L, a, C$  que aparecen en las ecuaciones, y también valores para  $N_x, N_t, t_f$ , que corresponden al número de nodos en la discretización espacial y temporal, y al tiempo final de la simulación, respectivamente.

Con estos parámetros ya se puede empezar el desarrollo del problema. Se entrega además el valor de las condiciones iniciales, y se localiza la variable  $u$ , que quedará como un array de tamaño  $N_x$  y guardará el valor de la temperatura en cada posición. Notamos que esta última variable no almacena el tiempo, dado que no es necesario para resolver la ecuación y no es pedido en este inciso. Esto es una ventaja computacional al utilizar mucha menos memoria.

Se define el valor de  $u$  en los extremos y se realiza una primera iteración en un intervalo  $i \in [2, N_x - 1]$ . Para cada  $i$  en esta iteración escribiremos en  $u$  el valor correspondiente a la condición inicial según la Ecuación 7. El valor de  $x$  en esa función se tomará como  $x_i = (i - 1)\Delta x$ .

Una vez completado el primer ciclo se comienza a iterar sobre  $j$ . La idea es que para cada  $j$  hagamos un ciclo similar al anterior en  $i$ , pero esta vez el valor de la temperatura  $u$  no estará dada por la condición inicial si no que dependerá del nodo temporal anterior como indica la Ecuación 9. Por supuesto, lo que genera este algoritmo es que reescribe  $u$  para cada iteración temporal, por lo que al finalizar el ciclo sólo tendremos  $u$  en el tiempo final.

El código en una última línea se preocupa de escribir esta relación  $u(x, t_f)$  en un archivo.

- **(b)** Simplemente se utilizó el código descrito anteriormente asignando los valores pedidos por el enunciado. Dado que tenemos tres tiempos finales a evaluar, preferí almacenarlos en un array e iterar sobre ellos después de definir las condiciones iniciales (son las mismas en todos los casos).

Los resultados obtenidos para la temperatura  $u$  en el tiempo final como función de la posición en la varilla son:

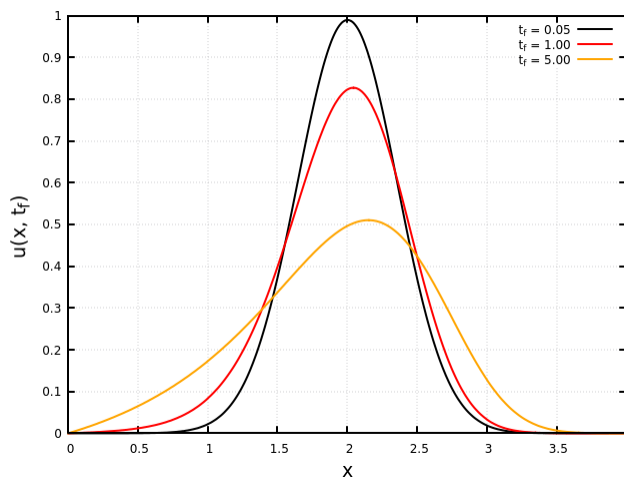


Figura 5: Temperatura  $u(x, t_f)$  para distintos  $t_f$ .

El resultado es muy acorde con el fenómeno físico que hay detrás. Al realizar los cálculos con  $t_f = 0,05$  tenemos la situación más cercana a las condiciones iniciales, que son, precisamente, una Gaussiana centrada en  $L/2 = 2$  de amplitud  $C = 1$ . Notamos que a tiempos finales posteriores disminuye la amplitud máxima de la curva, sugiriendo que la temperatura más alta es menor, a la vez que la curva en sí se vuelve más ancha, sugiriendo que está más distribuida a lo largo de la varilla. Notamos que el comportamiento natural es la tendencia al equilibrio,



y la curva a un tiempo mayor es la que más se acerca a una distribución uniforme de temperatura.

Más allá de este análisis, observamos que las curvas también son asimétricas, y esto tiene que ver directamente con la expresión  $D(x)$ . Bajo los parámetros utilizados esta expresión queda  $D = 0,2e^{-x}$ , que es una curva cuyo máximo en este intervalo está en  $x = 0$ , y luego es monótona decreciente a mayores  $x$ . Esto se refleja en que los sectores más cercanos a  $x = 0$  tienen una mayor ponderación en la temperatura resultante.

- (c) Según lo discutido en el inciso anterior, a distintos tiempos finales notamos que la temperatura tendía a un equilibrio, distribuyéndose a lo largo de toda la varilla y ponderado por un factor asimétrico.

Este mismo comportamiento sería esperable al graficar una evolución temporal. En este inciso se pide graficar la evolución para dos puntos, en un intervalo temporal que abarca de cero a veinte. Estos puntos se encuentran uno en el primer cuartil de la varilla ( $x_1$ ) y otro en la mitad de la varilla ( $x_2$ ). El máximo de temperatura en el momento inicial se encuentra en la mitad de la varilla, por lo que el comportamiento esperado para la temperatura en ese punto es que sólo decaiga inicialmente. Al contrario, el punto que se encuentra en el primer cuartil inicialmente se encuentra a temperatura casi cero, por lo que debería crecer en un principio, hasta alcanzar un determinado máximo, para luego volver a decaer buscando el equilibrio global.

En el código, se tomó el cuartil y la mitad de  $N_x$ , y se estudió la evolución temporal de estos puntos guardando su estado de temperatura para cada punto temporal en la grilla. El código es una adaptación simple del inciso anterior, y se encuentra en **P3/P3ALT\_C.f90**.

Los resultados obtenidos se muestran en la siguiente figura:

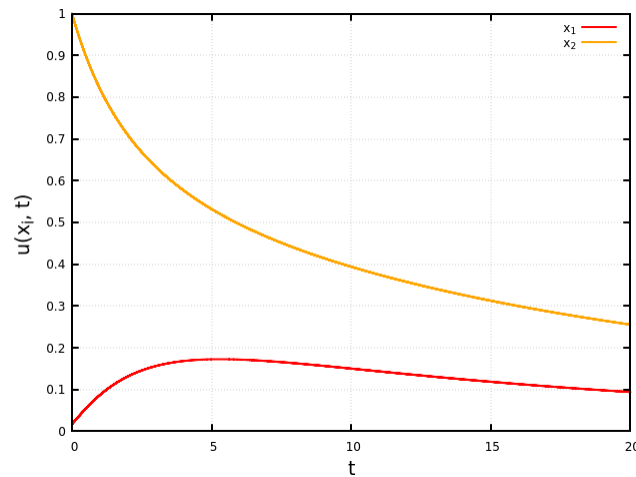


Figura 6: Evolución temporal de la temperatura:  $u(x_1, t)$  y  $u(x_2, t)$ .

El comportamiento encontrado coincide con lo predicho en la discusión previa.

## Pregunta 4

Comenzamos por escribir la Ec. de Schrödinger estacionaria en una dimensión:

$$-\frac{\hbar^2}{2\mu} \frac{d^2}{dx^2} u(x) + V(x)u(x) = Eu(x) \quad (11)$$

donde  $\mu = \frac{3M_H M_N}{3M_H + M_N}$  es la masa reducida, y  $V(x)$  es el potencial generado por la pared de hidrógeno, cuya forma es

$$V(x) = V_0 \tan^2\left(\frac{x}{r_0}\right) - V_1 \sin^2\left(\frac{x}{r_0}\right) \quad (12)$$

con  $V_0 = 45,4521eV$ ,  $V_1 = 52,9406eV$ ,  $r_0 = 1,38488\text{\AA}$ .

Al discretizar la segunda derivada obtenemos la siguiente expresión:

$$-\frac{\hbar^2}{2\mu} \left[ \frac{u_{k+1} - 2u_k + u_{k-1}}{h^2} \right] + V_k u_k = E u_k \quad (13)$$

donde el subíndice  $k$  hace alusión a la función evaluada en el elemento  $k$  de la diagonal principal, y análogamente con  $k-1$  para elementos tridiagonales. El parámetro  $h$  indica la distancia entre nodos en una discretización del intervalo  $x \in [-r_0/2, r_0/2]$ .

De esto podemos expresar individualmente los elementos de la diagonal principal ( $d_k$ ) y los elementos de las diagonales adyacentes ( $e_k$ ):

$$d_k = \frac{\hbar^2}{2\mu} \frac{2}{h^2} + V_k \quad (14)$$

$$e_k = \frac{\hbar^2}{2\mu} \frac{1}{h^2} \quad (15)$$

Para resolver el problema se utilizó el código **osciladorCuantico.f90** visto en clases, y la respectiva subrutina **diagotri.f90** para diagonalizar la matriz inicialmente tridiagonal. Por supuesto, el código se adaptó al problema actual, modificando el potencial, y esta vez utilizando las unidades necesarias para obtener los autovalores en  $eV$ . El código completo se encuentra en **P4/P4.f90**, y los valores de las constantes se indican al inicio del código en forma de parámetros.

- **(a)** Para evaluar la robustez de los autovalores que arroja el programa se iteró sobre distinto número de puntos en la malla, tal y como se realizaba en el programa visto en clases. Los autovalores se imprimen en pantalla, las autofunciones se almacenan en *data\_oscilador* y el potencial se almacena en *potencial*.

Los autovalores obtenidos son:

$N$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$
50	-0.206512	-0.206401	-0.069246	-0.062948	0.023201	0.070299	0.143738
100	-0.206252	-0.206142	-0.068404	-0.061911	0.024552	0.073163	0.148330
200	-0.206187	-0.206078	-0.068196	-0.061653	0.024888	0.073877	0.149481
400	-0.206171	-0.206062	-0.068144	-0.061588	0.024972	0.074057	0.149772
800	-0.206167	-0.206058	-0.068131	-0.061572	0.024993	0.074103	0.149847

Cuadro 3: Primeros 7 autovalores obtenidos.

Notamos que hay sólo dos autovalores entre 0 y 0.1, el intervalo pedido.

■ (b) y (c)

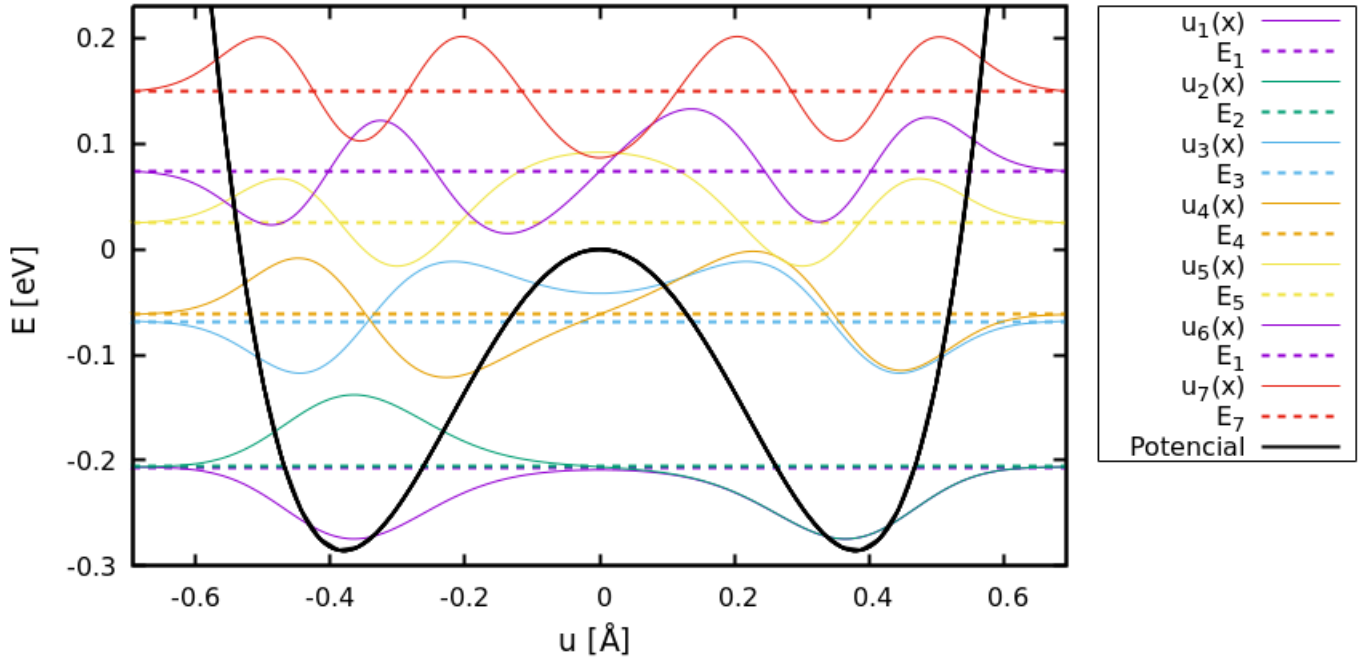


Figura 7: Potencial, Autovalores  $E_n$  y Autofunciones  $u_n$ .

En la Figura 8 se muestran los siete primeros autovalores sobre el potencial, y sus respectivas autofunciones, extraídas como las 7 primeras columnas del archivo *data\_oscilador*.

Notamos que el Nitrógeno, en su estado fundamental de energía  $E_1$ , no tiene la energía suficiente como para traspasar el potencial generado por la pared de hidrógeno. Por ello, para que pueda realizar una inversión, debe existir efecto túnel.

Calculamos la integral del cuadrado de la función de onda del nitrógeno en estado fundamental en la región donde la energía es menor que la barrera dentro del potencial, de esta manera estamos calculando la probabilidad de que exista efecto túnel, y por lo tanto, de que el nitrógeno realice una inversión.

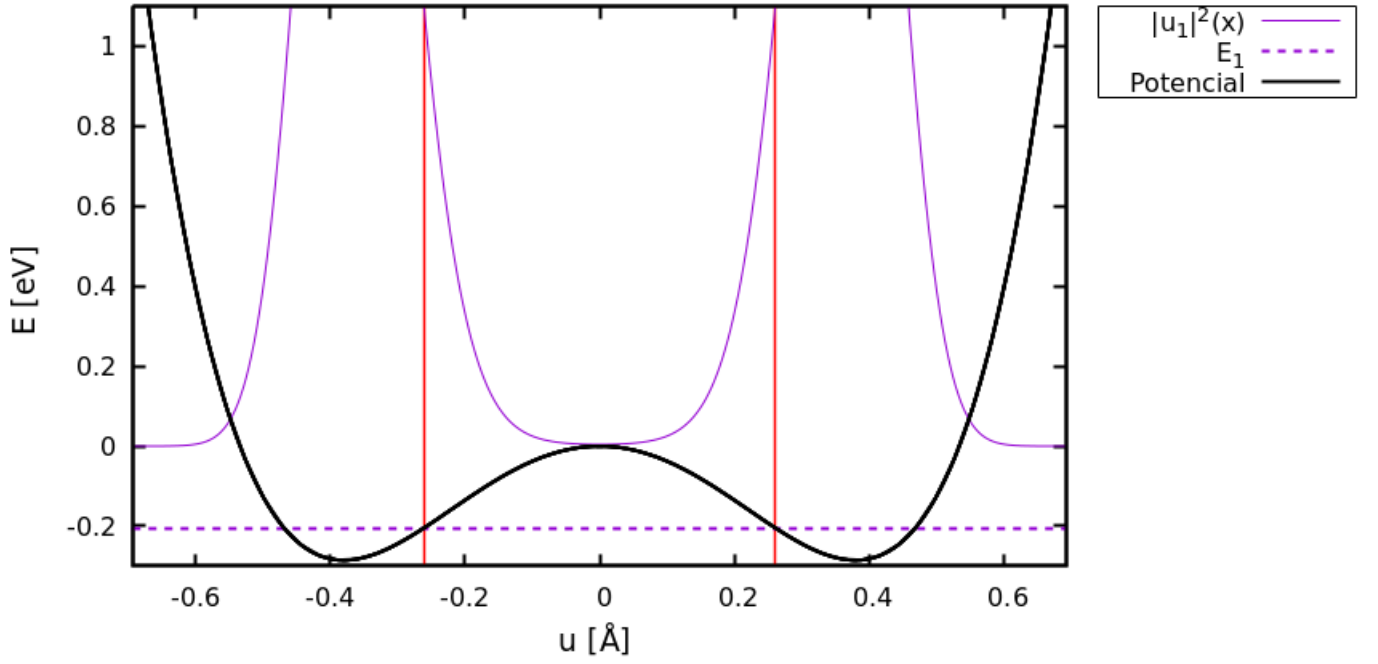


Figura 8: Potencial y gráfica referencial de la función de onda al cuadrado. Se encierra en líneas verticales la región de integración.

El cálculo de la integral se realizó tomando los puntos en el intervalo entre las líneas rojas, sumando todos los valores y multiplicando cada uno por el parámetro  $h$ , entendido como la distancia entre los puntos discretizados en la malla de  $x$ . Esto está escrito en el programa del archivo **P4/integral.f90**.

El resultado, arrojando la probabilidad de tunelamiento e inversión del nitrógeno, está dado por la probabilidad  $P$ :

$$P = 0,112515 \quad (16)$$