

Universidad de Alicante

FACULTAD DE CIENCIAS

BLOQUE 2

Física Computacional

Gabriel Simón López

2023/2024

Índice

1. Introducción	2
2. Práctica 1	2
3. Práctica 2	6
4. Práctica 3	9
5. Práctica 4	11
6. Práctica 5	13
7. Práctica 6	14
7.1. Número de partículas	14
7.2. Temperatura del sistema	15
7.3. Número de pasos de Montecarlo	15
7.4. Extra	15

1. Introducción

En este documento presentaré:

- Los programas de cada ejercicio.
- Un breve informe explicando qué partes hemos hecho de cada ejercicio, resaltando las partes más importantes de cada programa, mostrando los resultados más importantes como comparación de métodos, convergencia, etc. Menos en las prácticas en las que se piden animaciones, estas se encontrarán en los programas.
- Las conclusiones sacadas de los programas o extras.

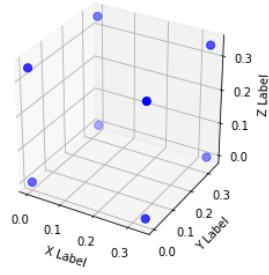
2. Práctica 1

En esta práctica crearemos un programa el cual generará o bien la celda unidad o el cristal de las siguientes redes cristalinas: cúbica, fcc, bcc y hcp. Además, para cada uno de los ejemplos anteriores, se le ha asociado un elemento de la tabla periódica el cual presente esa estructura y los representaremos en una gráfica en 3D.

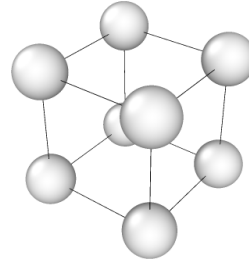
Por lo tanto, el programa nos pedirá que le indiquemos si queremos una celda unidad o un cristal luego, el tipo de red deseada y por último, el elemento. Y este nos devolverá, una representación en 3D de los valores pedidos, además de un fichero el cual contendrá el número de posiciones generadas (número de átomos) y el número de identificación de la posición x,y,z .

Este fichero obtenido lo hemos introducido en OVITO con el fin de, no solo comparar las estructuras generadas en nuestro programa en Python, sino para realizar por ejemplo cortes en la dirección de Miller (111).

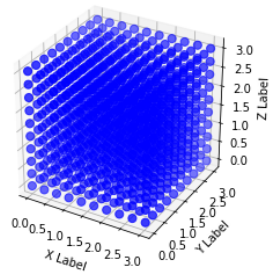
Es importante comentar que este programa también se ha realizado en la primera práctica de ordenador de Física del Estado Sólido, sirviendo así como complemento en ambas prácticas.



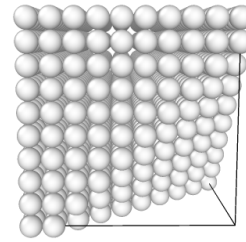
(a) *Celda unidad cúbica simple Python*



(b) *Celda unidad cúbica simple OVITO*

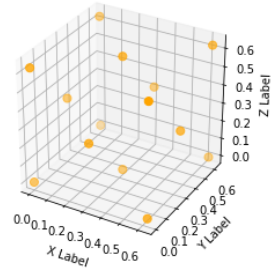


(c) *Cristal cúbica simple Python*

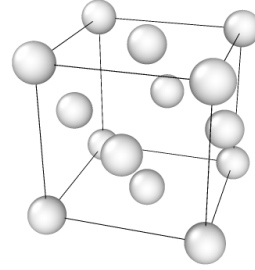


(d) *Cristal sc OVITO corte (111)*

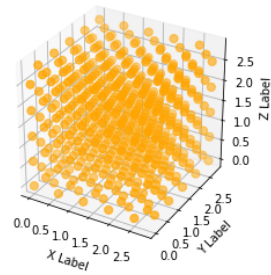
Figura 1: *Resultados Cúbica Simple Polonio*



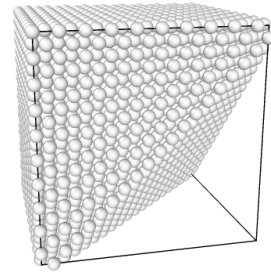
(a) Celda unidad FCC Python



(b) Celda unidad FCC OVITO

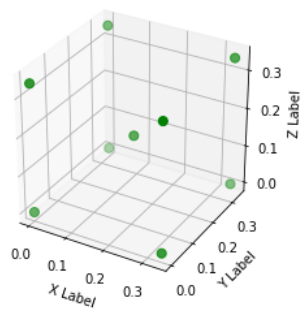


(c) Cristal FCC Python

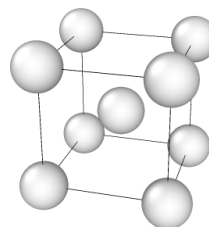


(d) Cristal FCC OVITO con corte (111)

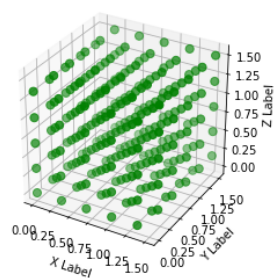
Figura 2: Resultados FCC Xenón



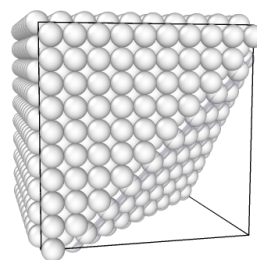
(a) *Celda unidad BCC Python*



(b) *Celda unidad BCC OVITO*

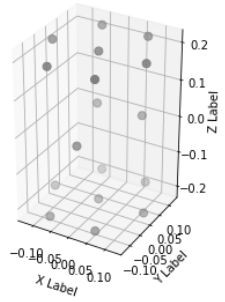


(c) *Cristal BCC Python*

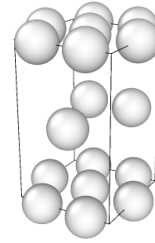


(d) *Cristal BCC OVITO con corte (111)*

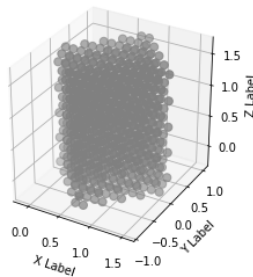
Figura 3: *Resultados BCC Vanadio*



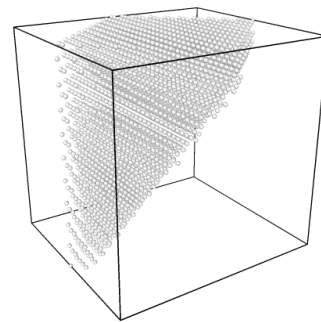
(a) Celda unidad HCP Python



(b) Celda unidad HCP OVITO



(c) Cristal HCP Python



(d) Cristal HCP OVITO con corte (111)

Figura 4: Resultados HCP Cobalto

En donde hemos obtenido para la celda unidad de la red cúbica 8 átomos en las esquinas ,para la red bcc 9 átomos en total. Para la red fcc obtenemos los 8 átomos de la red cúbica más 6 correspondientes a las caras, 14 en total, por último para la red de hcp se tienen 15 en total, los 6 en cada cara y 3 en el centro.

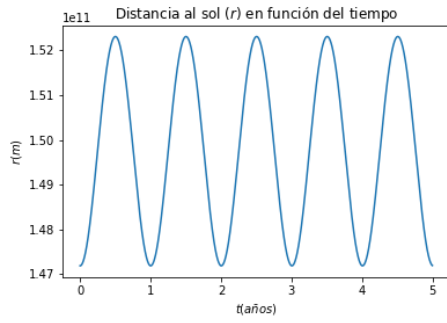
Es importante comentar como para cada elemento tendremos que escribir los parametros de red que lo definen, siendo estos distintos en cada caso.

El código consta de 4 partes muy diferenciadas, la primera son las funciones, en la segunda se encuentra todo lo relacionado con los inputs y las posibilidades en cada caso, en el tercero el código para obtener un archivo con lo pedido y por último, una parte para la representación en 3D.

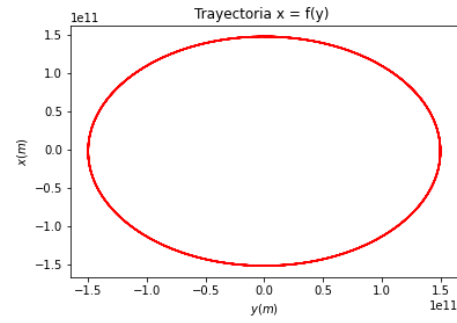
3. Práctica 2

En esta práctica hemos calculado la órbita de la Tierra alrededor del Sol mediante tres métodos distintos: Verlet, Runge-Kutta segundo orden y Runge-Kutta de cuarto orden. Para cada método presentaremos una gráfica relacionada con la distancia al Sol en función del tiempo, otra con la trayectoria en función del tiempo y por último las energías en función del tiempo.

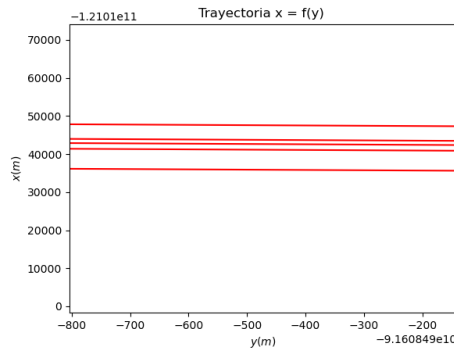
En primer lugar comenzamos con dos condiciones iniciales: la posición y la velocidad en el instante 0. Utilizaremos el algoritmo de Verlet para calcular los valores de x, y, v_x, v_y a partir de los valores anteriores en intervalos de distancia h , que será diferencial. Avanzaremos en distancias de $h/2$ y usaremos las fórmulas planteadas en clase, de las cuales obtendremos:



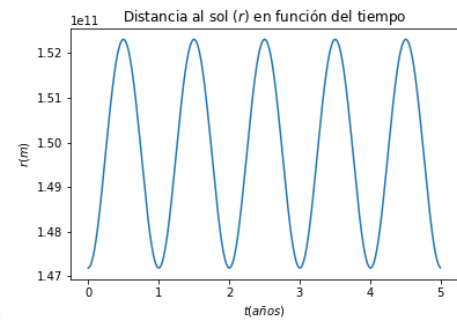
(a) Radio en función del tiempo Verlet



(b) Trayectoria en función del tiempo Verlet



(c) Trayectoria en función del tiempo Verlet
Ampliado



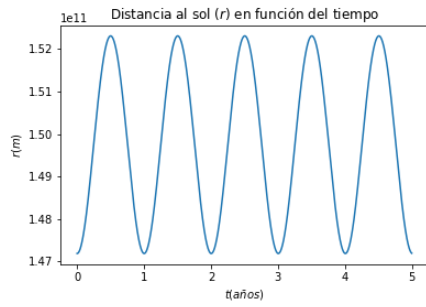
(d) Energías en función del tiempo Verlet

Figura 5: Resultados algoritmo Verlet

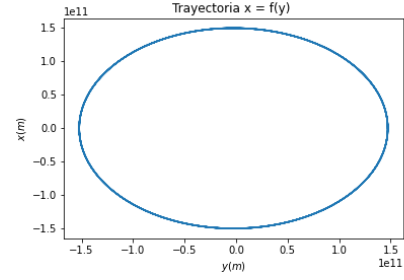
Por lo que podremos concluir:

- Tiene sentido que la distancia entre el Sol y la Tierra represente un movimiento sinusoidal. Uno podría pensar que debería ser constante al ser algo similar a un movimiento circular, pero esto se daría si la órbita fuese un círculo perfecto. En el caso real las órbitas son elípticas, y por ello aparece un movimiento periódico que define una distancia mínima cuando la Tierra pasa por el perifelio y máxima en el afelio.
- La trayectoria representa una elipse. Además, se aprecia si hacemos zoom a las órbitas cómo varía la distancia.
- Los resultados obtenidos conservan la energía total, que es lo que se esperaría para este caso. Además, vemos como ambas energías varían de forma periódica.

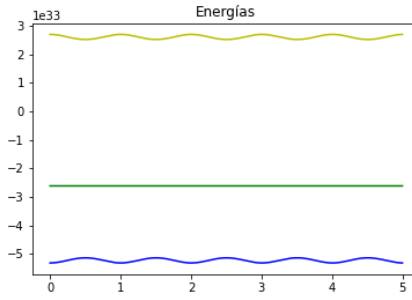
Estudiaremos los mismos casos para Runge Kutta orden 2 y Runge-Kutta orden 4 como extra. En los que obtendremos lo siguiente:



(a) Radio en función del tiempo RK2

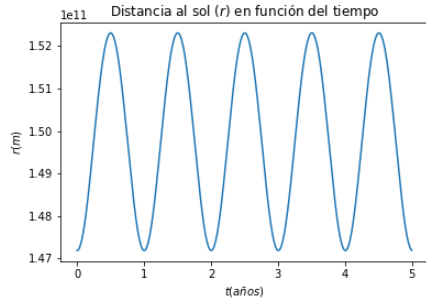


(b) Trayectoria en función del tiempo RK2

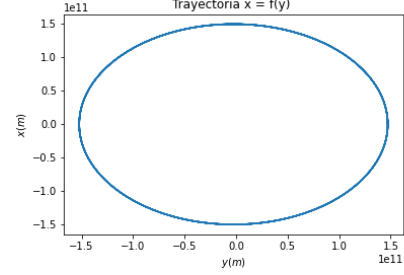


(c) Energías en función del tiempo RK2

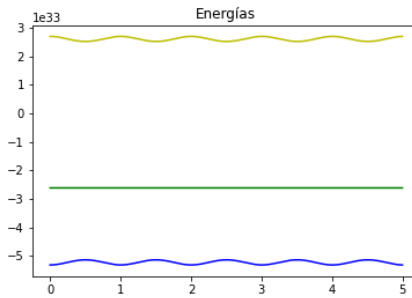
Figura 6: Resultados Runge-Kutta orden 2



(a) Radio en función del tiempo RK4



(b) Trayectoria en función del tiempo RK4



(c) Energías en función del tiempo RK4

Figura 7: Resultados Runge-Kutta orden 4

En donde podemos observar los mismos resultados que los obtenidos con el algoritmo de Verlet, sin embargo al comparar las energías observamos que en el caso de Verlet y Runge-Kutta de orden 2 no se conservan, mientras que en el Runge-Kutta 4, si.

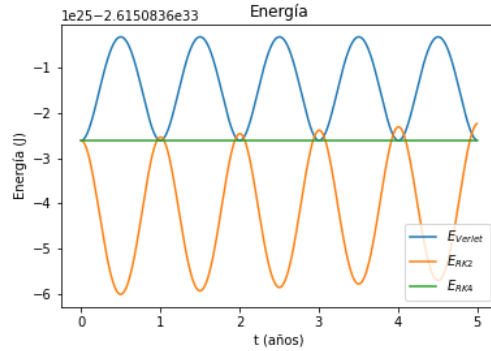


Figura 8: Comparación de energías en los distintos algoritmos

Podremos concluir que aunque el método de Runge-Kutta 4 es el más costoso, también será el más fiable.

4. Práctica 3

En esta práctica vamos a plantear un sistema que mediante números aleatorios resuelva una integral. En este caso será: $f(x) = \sin^2 \frac{1}{x(2-x)}$ en el intervalo entre 0 y 2 y como extra, la Gaussiana.

Las resolveremos de dos métodos distintos, uno en el que colocaremos aleatoriamente puntos en un rectángulo que contenga el área en la que queremos calcular la integral. En donde evaluaremos que puntos quedan dentro (aceptamos) o fuera (no aceptamos) del área encerrada y calcularemos la integral con los puntos aceptados. Y otro en el que evaluaremos la función en un número discreto de valores aleatorios de x , por lo que empleando la definición del valor esperado de una función y el Teorema del Valor Medio para integrales, obtendremos el valor de la integral.

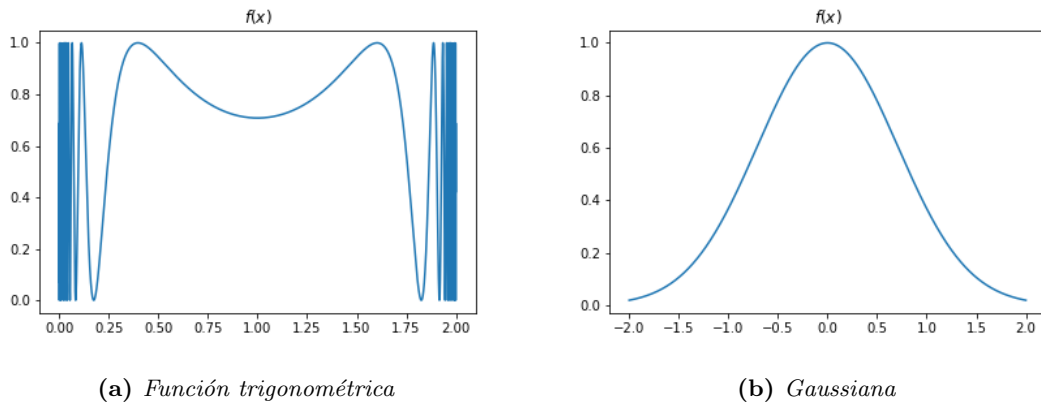
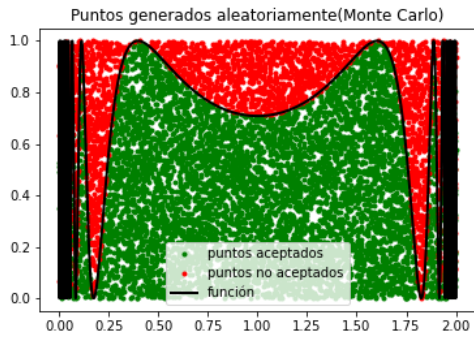
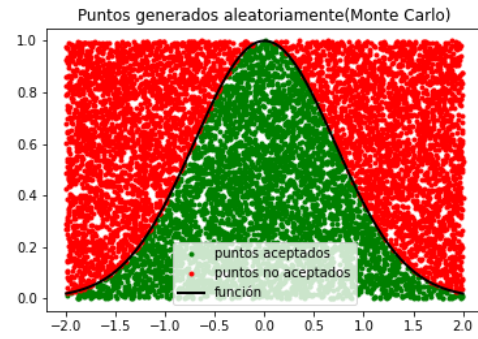


Figura 9: Funciones a integrar

Una vez graficadas las funciones veamos la misma gráfica anterior tras haber lanzado aleatoriamente 10^4 puntos. Es importante destacar que la Gaussiana se integra entre -2 y 2 en el eje x , mientras que la trigonométrica entre 0 y 2, esto será importante para el valor de la integral.



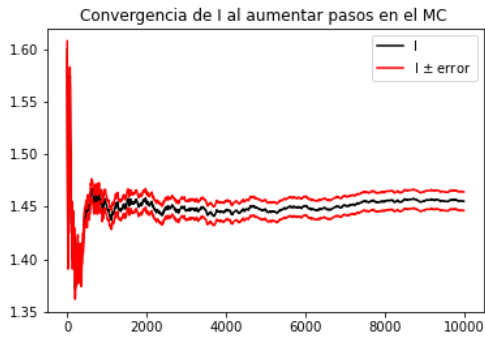
(a) *Función trigonométrica con puntos aleatorios*



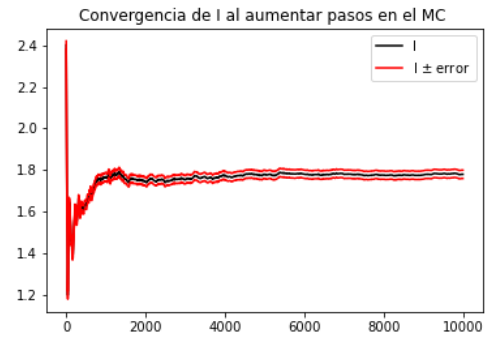
(b) *Gaussiana con puntos aleatorios*

Figura 10: *Funciones a integrar con los puntos aleatorios*

Tal y como hemos explicado antes con la cantidad de puntos aceptados podremos calcular el valor de la integral y su error, la cual observaremos en las siguientes gráficas:



(a) *Evolución valor integral trigonométrica*

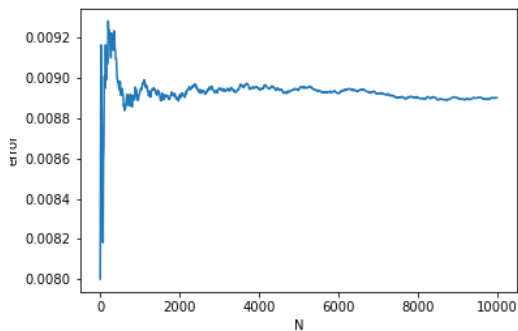


(b) *Evolución valor Gaussiana*

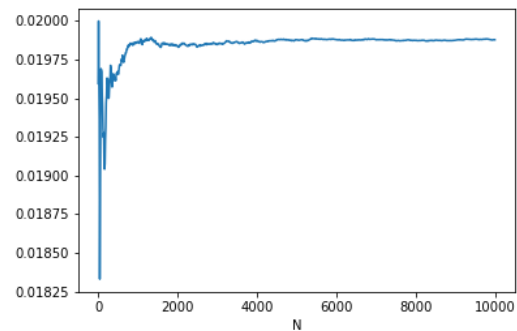
Figura 11: *Evolución valor integrales*

En donde observamos como el valor devuelto por el programa y el que oscilan las gráficas no solo coinciden sino que también se adaptan al valor real de estas.

Por último estudiaremos la evolución del error:



(a) *Evolución error integral trigonométrica*



(b) *Evolución error Gaussiana*

Figura 12: *Evolución error integrales en función de N*

En el que podemos observar se estabiliza al aumentar el número de pasos, esto puede ser debido a que con el método Montecarlo, una vez este encuentra el mínimo o el valor deseado este hará cambios pero ya oscilará en ese valor, ya que por la naturaleza de este algoritmo no aceptará esos cambios.

Como conclusiones, además de las anteriores, podemos añadir:

- Los cálculos se harán menos precisos dependiendo de lo grande que sea la altura de nuestro rectángulo (si mantenemos el numero de puntos aleatorio). Esto es porque hay un área mayor y la probabilidad de que cada punto aleatorio caiga en la región encerrada por la función es menor
- Si aumentamos el numero de pasos del Monte Carlo, el valor de la integral será mucho mas preciso (su error disminuirá) ya que tenemos un conjunto de datos mayor y nuestro método se basa en probabilidad

Por último, se ha calculado el volumen de una esfera de dimensión D , en donde hemos seguido el mismo procedimiento, solo que hemos tenido que centrar la caja en $-1,1$.

Volumen de la hiperesfera para 10000 pasos = 2.56

Figura 13: Volumen devuelto por el programa para esfera $D = 10$

5. Práctica 4

Hemos calculado la energía total de una red fcc para distintos átomos en la celda unidad, considerando el potencial de interacción de Lennard-Jones con los parámetros del cobre. En donde hemos tomado como radio de corte el valor de 3 veces σ . El potencial usado tiene la siguiente forma:

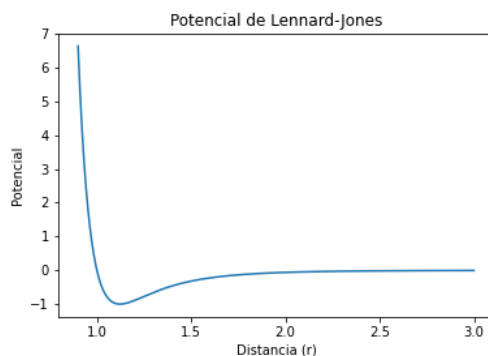


Figura 14: Potencial de Lennard-Jones

Realizaremos el cálculo para dos condiciones de contorno diferentes: superficies libres y condiciones periódicas. Para esta última hemos realizado una corrección de la distancia en la que establecemos que si el valor de esta es mayor que la mitad de la longitud de la red realizaremos una corrección en la que desplazaremos a la izquierda o a la derecha el átomo multiplicado por el parámetro de red del cobre (en este caso), para así simular una "periocidad".

El programa nos devuelve el valor de la energía total del sistema en eV, la energía por átomo y las siguientes gráficas:

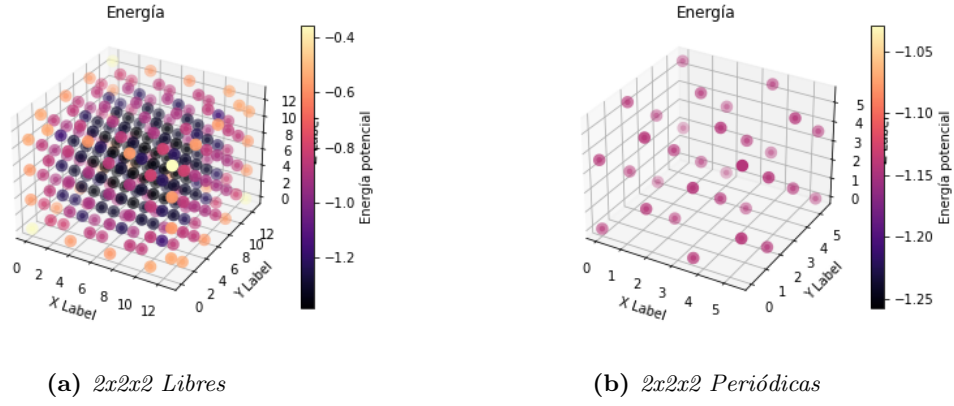


Figura 15: Resultados sistema $2 \times 2 \times 2$

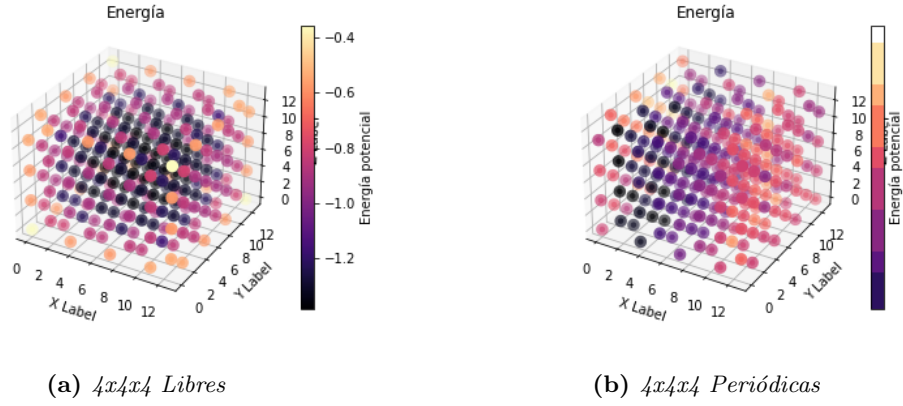


Figura 16: Resultados sistema $4 \times 4 \times 4$

Observamos que si simulamos el sistema para 2,3,4 y 5 obtenemos los resultados de la presentación de la clase teórica. Es importante destacar que a partir del sistema $5 \times 5 \times 5$ empezamos a obtener los mismos valores de energía.

Además podemos concluir que, a medida que aumentamos el tamaño de la red, aumenta la energía del sistema (el valor absoluto) para ambas condiciones de contorno. También podemos observar como la energía del sistema aumenta si consideramos condiciones periódicas que si no, esto es debido a que en los bordes dependiendo de que condiciones elijamos tendremos una mayor o menor contribución a la energía del sistema.

Por último, si realizamos este mismo estudio pero para distintos radios de corte obtenemos lo siguiente:

Radio de corte	Energía del sistema cc libres (eV)	Energía del sistema cc periódicas (eV)	Energía por átomo cc libres (eV)	Energía por átomo cc periódicas (eV)
$2,5\sigma$	-94.875	-144.048	-0.878	-1.334
3σ	-95.885	-146.217	-0.888	-1.354
$3,5\sigma$	-96.311	-146.301	-0.892	-1.355

Figura 17: Tabla $3 \times 3 \times 3$ para distintos radios de corte

En donde observamos como conforme aumenta el radio de corte aumenta la energía del sistema.

6. Práctica 5

Esta práctica se trata de una continuación de la anterior, en la que añadiremos al script realizado el cálculo de las fuerzas, y el gradiente de interacción. Mantendremos los mismos parámetros y el mismo potencial de interacción. Por lo tanto, calcularemos la fuerza sobre cada átomo del sistema tanto con condiciones de contorno libres como periódicas tomando siempre un radio de corte de 3σ .

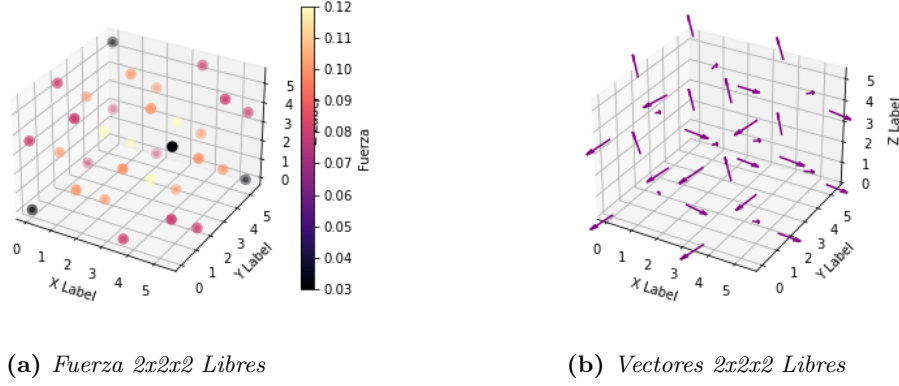


Figura 18: Resultados sistema 2x2x2 Libres

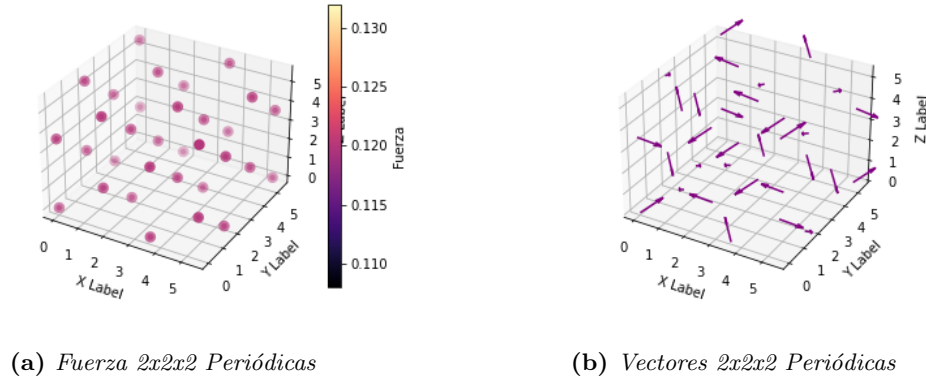


Figura 19: Resultados sistema 2x2x2 Periódicas

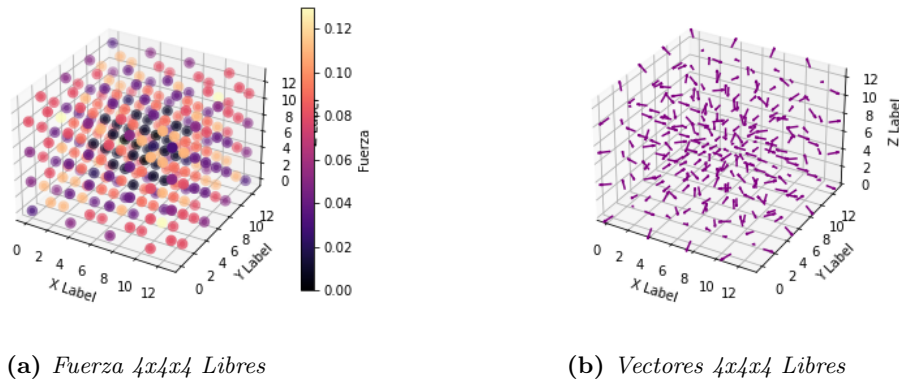
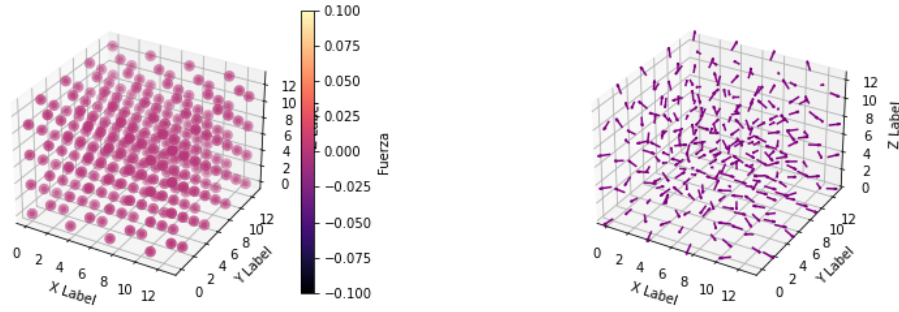


Figura 20: Resultados sistema 4x4x4 Libres



(a) Fuerza $4x4x4$ Periódicas

(b) Vectores $4x4x4$ Periódicas

Figura 21: Resultados sistema $4x4x4$ Periódicas

Es importante comentar que para representar el vector fuerza se ha empleado una flecha para representar la dirección y un mapa de color para representar el módulo.

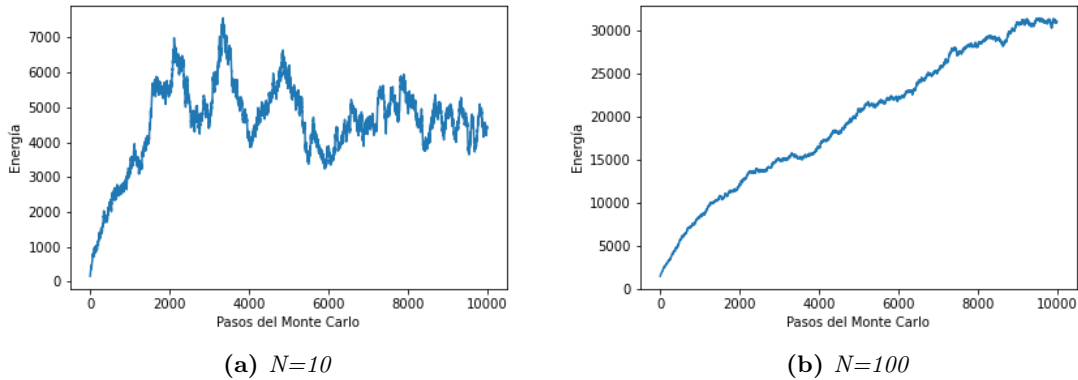
7. Práctica 6

En esta práctica implementaremos el método montecarlo para un sistema de un gas ideal de masa, m , en una caja cúbica de longitud, L , con energía que depende de los números cuánticos, n_x, n_y y n_z .

Dando por hecho el conocimiento del funcionamiento del método Montecarlo presento los siguientes gráficos.

7.1. Número de partículas

En este apartado mantenemos constantes todos los parámetros y variaremos la N . Sea $\text{pas} = 10000$ y $KT = 300$ obtenemos:



(a) $N=10$

(b) $N=100$

Figura 22: Resultados variando N

Como mantenemos constantes todos los parámetros menos el número de partículas del sistema, puede darse el caso que para un mayor número de partículas menos probable será encontrar el equilibrio.

Con las siguientes gráficas podremos concluir que necesitaremos una mayor cantidad de pasos Montecarlo a mayor tamaño del sistema, ya sea en número de partículas o el tamaño de la caja, o mayor sea la temperatura.

7.2. Temperatura del sistema

En este apartado mantenemos constantes todos los parámetros y variaremos la KT . Sea $N = 100$ y $\text{pas} = 10000$ obtenemos:

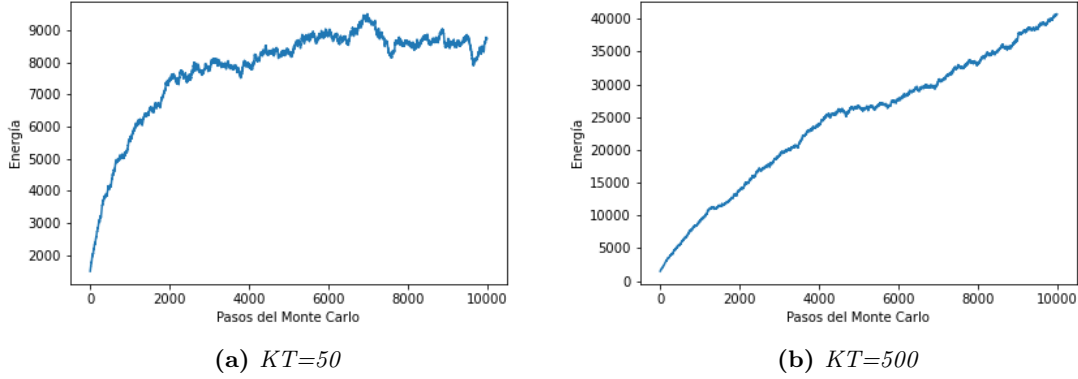


Figura 23: Resultados variando KT

Observamos la misma conclusión que en apartados anteriores.

7.3. Número de pasos de Montecarlo

En este apartado mantenemos constantes todos los parámetros y variaremos la variable, pas . Sea $N = 100$ y $KT = 300$ obtenemos:

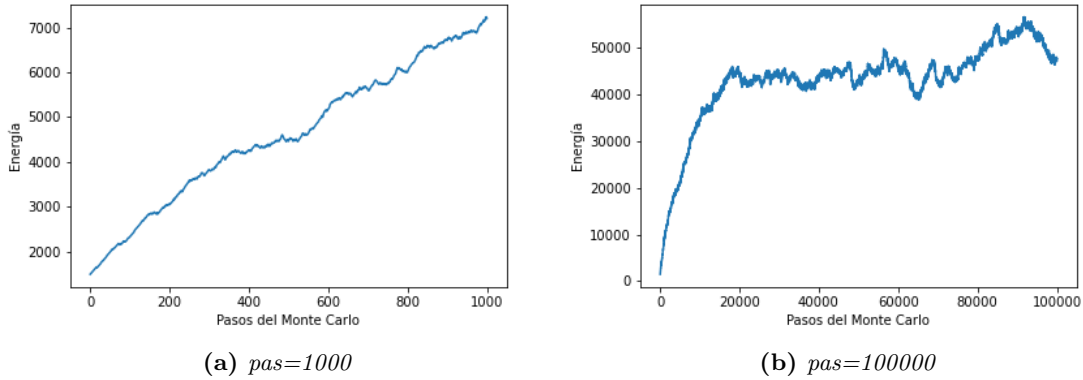


Figura 24: Resultados variando pas

Observamos la misma conclusión que en apartados anteriores.

7.4. Extra

Como extra utilizaremos el método montecarlo para conseguir que el Cu (cuya estructura es la fcc) se organice en una fcc, en donde supuestamente tendrá su mínimo de energía.

Es decir, crearemos un programa el cual nos pregunte el número de partículas que queremos soltar, el lado del cubo de la simulación, la temperatura a la que queremos trabajar y el numero de pasos de Montecarlo deseados.

Utilizaremos el potencial de L-J usado en las practicas anteriores y necesitaremos definir el parametro de red del Cu.

De esta manera, cabe esperar que, dado que la estructura de menor energía para el Cu es la fcc, obtengamos efectivamente una fcc.

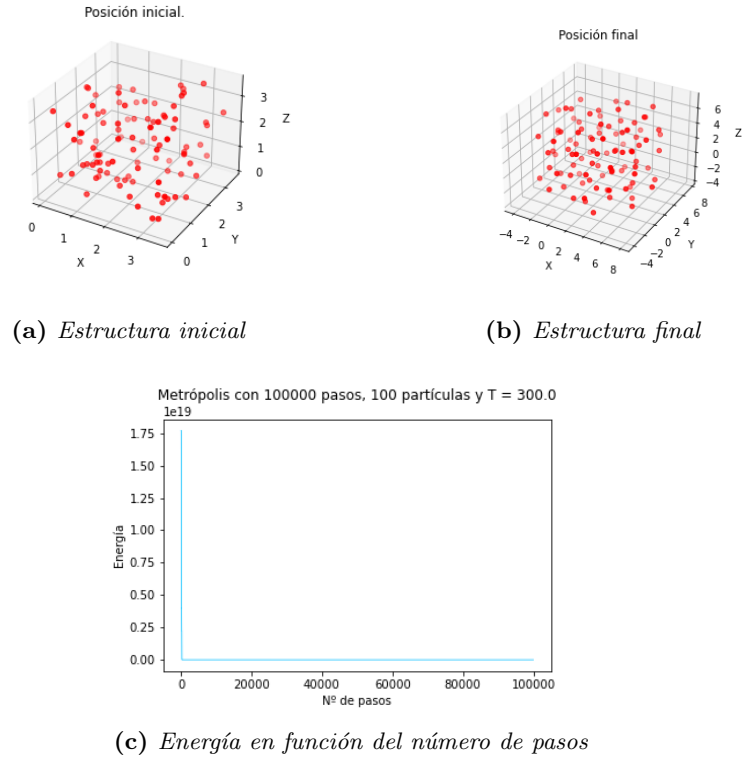


Figura 25: Resultados Montecarlo estructura Cu

En donde observamos como a una gran cantidad de pasos, los átomos que inicialmente se encontraban aleatoriamente repartidos por la celda de simulación, se han colocado en una estructura fcc, la cual sabemos que es la estructura de menor energía para este elemento. Siendo la energía inicial $E_0 = 1,77e + 19$ y la final $E_f = -0,797$.