<u>Práctica 1</u>: Determinación de la constante de Madelung

En Física de la Materia Condensada, la constante de Madelung da una medida del potencial eléctrico que experimenta un átomo de un sólido debido a las cargas de los iones que lo rodean. Se puede calcular como una suma de la contribución de todos los iones. En el cloruro sódico los iones de Cl^- y Na^+ se alternan en una estructura cúbica, donde podemos definir la posición de cada ión mediante tres coordenadas enteras, (*i*, *j*, *k*). Si se toma el origen de coordenadas en la posición de un ión de Na^+, habrá iones de Na^+ en las posiciones donde *i*+*j*+*k* sea par, e iones Cl^- donde dicha suma sea impar. Sea *a* la distancia entre iones. Entonces la distancia de un ión cualquiera al origen será:

$$\sqrt{(ia)^2 + (ja)^2 + (ka)^2} = a\sqrt{i^2 + j^2 + k^2}$$

El potencial creado por este ión en el origen será:

$$V(i,j,k) = \pm \frac{e}{4\pi\varepsilon_o a\sqrt{i^2 + j^2 + k^2}}$$

siendo el signo positivo o negativo dependiendo de si (*i*+*j*+*k*) es par (+) o impar (–). El potencial total experimentado por el átomo central es la suma sobre todos los iones. Consdierand un cubo de lado 2L centrado en el origen, dicho potencial será

$$V_{total} = \sum_{\substack{i,j,k=-L \\ no\ i=j=k=0}}^{L} V(i,j,k) = \frac{e}{4\pi\varepsilon_o a} M$$

En la ecuación anterior, *M* es la constante de Madelung, es decir:

$$M = \sum_{\substack{i,j,k=-L \\ no\ i=j=k=0}}^{L} \pm \frac{1}{\sqrt{i^2 + j^2 + k^2}}$$

Donde los términos con *i*+*j*+*k* par son positivos y los términos con *i*+*j*+*k* impar son negativos, y L⟶∞. En la práctica no se puede llegar a este límite, sino quedarse en valores lo más altos posible.

Escribir un programa para calcular la constante de Madelung en Python y otro en C++ y comparar el valor obtenido con el valor exacto (M = –1.74756) para varios valores de L. Comparar el tiempo de ejecución que se tarda en calcular la constante de Madelung entre el código escrito en Python y el código escrito en C++. Observar también si hay diferencias en tiempo de ejecución utilizando distintas funciones como exponenciales, bucles if, etc.

Entregar los códigos junto con un breve informe comentando los resultados. Los códigos deben proporcionar en pantalla los valores obtenidos para L=20, 50, 100 y 200, junto con el error absoluto y el error relativo (en %) cometido.

(Opcional: explorar herramientas para Python que permitan reducir el tiempo de ejecución)