

Control 2. Computación I

Jose M. Soler
Universidad Autónoma de Madrid

11 de febrero de 2019

1. La interacción entre átomos de gases nobles puede aproximarse bien por un potencial de Lennard-Jones (LJ)

$$V(r) = \epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^6 \right] \quad (1)$$

donde $\epsilon = 120\text{K}$ y $\sigma = 3.80\text{\AA}$ para argon.

- a) Escribir una función *forceLJ.m* con la siguiente interfaz (2 puntos):

```
function [fa,u] = forceLJ(ra,eps,sigma)
% Lennard-Jones forces and energy of a noble gas cluster
% uij = eps*((sigma/dij)^12 - 2*(sigma/dij)^6)
% fij = -duij/drij
%      = 12*eps*((sigma/dij)^12-(sigma/dij)^6)*rij/dij^2
% u = sum_{i<j}(uij)
% fi = sum_{j/=i}(fij)
% rij=ri-rj, dij=|rij|
% Input:
% ra(3,na) : atomic coordinates (Ang)
% eps      : LJ energy parameter (K)
% sigma    : LJ distance parameter (Ang)
% Output:
% fa(3,na) : atomic forces (K/Ang)
% u        : potential energy (K)
```

Sugerencia: partir de la función *gravity.m* de clase.

- b) En un programa *clusterMotion.m*, inicializar las posiciones de $n = 8$ átomos de argon en los vértices de un cubo de lado σ , y sus velocidades con una distribución normal aleatoria (usar `randn`) de anchura $\Delta v = \sqrt{T_0/m}$, siendo $T_0 = 50\text{K}$ la temperatura inicial y $m = 40\text{Da}$ la masa atómica del Ar. Restar la velocidad del centro de masas. (1.5 puntos)
- c) Definir una función *myforce* adecuada y llamar a la función *verlet* de clase para calcular la trayectoria durante un tiempo $t_{max} = 10\text{Ut}$, en pasos de $\Delta t = 5 \times 10^{-3}\text{Ut}$ (Ut es la unidad de tiempo definida después). (1 punto)

- d) Calcular en cada instante la energía potencial (llamando a `forceLJ`) y cinética, representando en un subplot la energía total y en otro la temperatura instantánea, definida por $T = E_c/(3n/2)$, en función del tiempo. (1 punto)
- e) Crear un video del agregado, representando en 3D cada átomo por una esfera de radio $\sigma/2$, cada 100 pasos de tiempo, manteniendo la figura durante 0.1s con `pause(0.1)`. Usar las funciones `sphere` y `surf` de Matlab (2 puntos). Alternativamente, representar cada átomo por un símbolo 'o' (1 punto).
- f) Para simular una medida experimental, sumar a las coordenadas un 'error' con una distribución normal de anchura $\Delta x = 0,01\text{\AA}$. Y escribir, en un fichero de texto `coords.dat`, el tiempo y las coordenadas atómicas cada 10 pasos, en el formato $(t, x_1, y_1, z_1, \dots, x_n, y_n, z_n)$, con una línea por tiempo). (1 punto)

Nota: las unidades empleadas son: masa: $Da = 1,66 \times 10^{-27} \text{kg}$; longitud: $\text{\AA} = 10^{-10} \text{m}$; energía: $K = k_B \times 1\text{K} = 1,38 \times 10^{-23} \text{J}$; tiempo $U_t = (Da \text{\AA}^2/K)^{1/2} = 1,097 \times 10^{-12} \text{s}$. Obsérvese que el símbolo K se usa tanto como unidad de energía como de temperatura. Estas unidades se dan como referencia, pero no necesitan usarse en el examen, pues todas las magnitudes se referirán a ellas.

- 2. Se supone ahora que desconocemos las constantes ϵ y σ de la interacción, y que debemos obtenerlas mediante el ajuste de la trayectoria obtenida en el primer problema. Para ello,

- a) Escribir una función `forceModel.m` con la siguiente interfaz (2 puntos)

```
function fa = forceModel(par,ra)
% Atomic LJ forces at nt times, as needed by nlinfit
% Input:
%   par(2)      : LJ interaction parameters:
%                 par(1) : energy parameter eps (K)
%                 par(2) : distance parameter sigma (Ang)
%   ra(3,na,nt) : atomic positions at nt times (Ang)
% Output:
%   fa(3*na*nt,1) : atomic forces at nt times,
%                 as a single column vector (K/Ang)
```

Sugerencia: llamar a `forceLJ` dentro de un bucle de tiempos y usar `reshape` para cambiar la forma final de `fa` a un vector columna.

- b) En un programa `fitForces.m`, leer los tiempos y posiciones del fichero `coords.dat`. Si no se pudo generar dicho fichero, usar el suministrado en la página web (1 punto). Alternativamente (0 puntos), cargar directamente los tiempos y posiciones con `load('coords.mat')`.
- c) Calcular las fuerzas atómicas en cada instante (salvo los instantes inicial y final) a partir de la masa del argón y de la derivada segunda de las posiciones, obtenida por diferencias finitas (tomar el intervalo de tiempo como diferencia de dos tiempos consecutivos). (1 punto)

- d) Inicializar el vector de parámetros $[\epsilon, \sigma]$ con valores `par0=[100,5]` y llamar a `nlinfit` para obtener los valores óptimos y una estimación de sus errores a partir de la matriz de covarianza, escribiendo dichos valores y errores en la línea de comandos (1.5 puntos). Nota: el segundo argumento de `nlinfit` debe ser un vector columna, por lo que deberá llamarse mediante

```
[par,~,~,cov]=nlinfit(ra,reshape(fa,3*na*nt,1),@forceModel,par0);
```

Notas:

- Los programas deben estar documentados en inglés y no deben escribir nada que no se pida.
- Las figuras deben incluir etiquetas en cada eje.
- Enviar los ficheros *forceLJ.m*, *clusterMotion.m*, *coords.dat*, *forceModel.m* y *fitForces.m* a jose.soler@uam.es antes de terminar el examen.
- Para cada función y programa, se valorará:
 1. Que tenga la interfaz que se pide, y haga lo que se pide sin errores.
 2. Que esté bien documentado.
 3. Que esté programado de forma sencilla y fácil de entender.
- La nota máxima será de 10 puntos, aunque la suma de los apartados sea mayor.