



**UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS**

# Simulación Dinámica Molecular en C++ para fluido con potencial de Lennard-Jones y Simulación de solución de la ecuación de Langiven para motor molecular de dos estados

**Gómez S. Angie V.  
Tovar R. Leonardo**

*Física computacional 2, Programa Académico de Física,  
Universidad Distrital Francisco José de Caldas*

1 de noviembre de 2025

## Potencial de Lennard-Jones:

### 1. Objetivo:

- analizar la dinámica molecular del gas, validando la conservación de la energía y comparando las propiedades estadísticas con la distribución de Maxwell-Boltzmann.

### 2. Deducción de la Fuerza Interatómica de Lennard-Jones

En mecánica clásica, la fuerza conservativa  $F(r)$  que actúa entre dos partículas se deriva del gradiente negativo de su energía potencial  $V(r)$ . Dado que el potencial de Lennard-Jones es isotrópico (solo depende de la distancia radial  $r$  y no de la orientación angular), podemos tratar el problema en una dimensión:

$$F(r) = -\nabla V(r) = -\frac{d}{dr}V(r) \quad (1)$$

Esta relación implica que el sistema siempre buscará minimizar su energía potencial; una pendiente negativa en el potencial genera una fuerza positiva (repulsiva) y viceversa.

### 2.1. Derivación Matemática

Partimos de la ecuación del potencial de Lennard-Jones definida anteriormente:

$$V(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] = 4\epsilon (\sigma^{12} r^{-12} - \sigma^6 r^{-6}) \quad (2)$$

Para encontrar la fuerza, calculamos la derivada respecto a  $r$  aplicando la regla de la potencia  $\frac{d}{dr}(r^n) = nr^{n-1}$ :

$$\frac{dV}{dr} = 4\epsilon \left[ \frac{d}{dr}(\sigma^{12} r^{-12}) - \frac{d}{dr}(\sigma^6 r^{-6}) \right] \quad (3)$$

$$= 4\epsilon [\sigma^{12}(-12r^{-13}) - \sigma^6(-6r^{-7})] \quad (4)$$

$$= 4\epsilon \left[ -12 \frac{\sigma^{12}}{r^{13}} + 6 \frac{\sigma^6}{r^7} \right] \quad (5)$$

Ahora, aplicamos la definición de fuerza ( $F = -dV/dr$ ), lo que invierte los signos de los términos:

$$F(r) = - \left( 4\epsilon \left[ -12 \frac{\sigma^{12}}{r^{13}} + 6 \frac{\sigma^6}{r^7} \right] \right) \quad (6)$$

Simplificando la expresión, obtenemos la ecuación general de la fuerza de Lennard-Jones:

$$F(r) = 24\epsilon \left[ 2 \frac{\sigma^{12}}{r^{13}} - \frac{\sigma^6}{r^7} \right] \quad (7)$$

O, factorizando para obtener una forma más compacta y común en literatura computacional:

$$F(r) = \frac{24\epsilon}{r} \left[ 2 \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] \quad (8)$$

## 2.2. Interpretación Física de la Fuerza

La ecuación (7) nos revela el comportamiento dinámico de las partículas en tres regímenes distintos:

- **Régimen Repulsivo (Principio de exclusión de Pauli) ( $F > 0$ ):** Cuando  $r < 2^{1/6}\sigma$ , el término  $r^{-13}$  domina. La fuerza es positiva, lo que indica que las partículas se empujan mutuamente para separarse. La pendiente es extremadamente pronunciada, modelando la "dureza" de los átomos.
- **Régimen Atractivo Fuerzas de London (Tipo de fuerza de Van der Waals) ( $F < 0$ ):** Cuando  $r > 2^{1/6}\sigma$ , el término  $r^{-7}$  domina. La fuerza es negativa, indicando una atracción que intenta unir las partículas. Esta fuerza decae rápidamente a cero a medida que aumenta la distancia.
- **Equilibrio ( $F = 0$ ):** En el punto  $r_{min} = 2^{1/6}\sigma$ , los términos atractivos y repulsivos se cancelan exactamente. En este punto, la aceleración relativa de las partículas es nula.

## 3. Algoritmo de Verlet de Velocidad

A diferencia del método de Euler básico, el algoritmo de Verlet de Velocidad permite calcular posiciones  $\vec{r}$ , velocidades  $\vec{v}$  y aceleraciones  $\vec{a}$  con un error de orden local  $\mathcal{O}(\Delta t^4)$  para la posición y  $\mathcal{O}(\Delta t^2)$  para la velocidad.

### 3.1. Esquema del Algoritmo

El avance del sistema desde un tiempo  $t$  a un tiempo  $t + \Delta t$  se realiza en dos etapas principales para asegurar que la velocidad y la posición estén sincronizadas. El procedimiento discretizado es el siguiente:

#### 3.1.1. Paso 1: Media actualización de velocidad y posición completa

Primero, avanzamos las velocidades medio paso de tiempo y utilizamos esa velocidad intermedia para calcular la nueva posición completa.

$$\vec{v}\left(t + \frac{\Delta t}{2}\right) = \vec{v}(t) + \frac{1}{2}\vec{a}(t)\Delta t \quad (9)$$

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}\left(t + \frac{\Delta t}{2}\right)\Delta t \quad (10)$$

#### 3.1.2. Paso 2: Cálculo de la Fuerza

Una vez que las partículas se han movido a sus nuevas posiciones  $\vec{r}(t + \Delta t)$ , el campo de fuerzas del sistema cambia. Es necesario recalcular las fuerzas (y por ende las aceleraciones) usando el potencial de interacción (por ejemplo, Lennard-Jones):

$$\vec{a}(t + \Delta t) = \frac{\vec{F}(\vec{r}(t + \Delta t))}{m} = -\frac{1}{m}\nabla V(\vec{r}(t + \Delta t)) \quad (11)$$

Este paso es computacionalmente el más costoso de la simulación.

#### 3.1.3. Paso 3: Segunda media actualización de velocidad

Finalmente, utilizamos la nueva aceleración para completar el paso de velocidad:

$$\vec{v}(t + \Delta t) = \vec{v}\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2}\vec{a}(t + \Delta t)\Delta t \quad (12)$$

### 3.2. Forma Compacta

Si combinamos los pasos, las ecuaciones maestras que definen la actualización en un ciclo completo de Dinámica Molecular son:

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)\Delta t^2 \quad (13)$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{\Delta t}{2}[\vec{a}(t) + \vec{a}(t + \Delta t)] \quad (14)$$

### 3.3. Ventajas del Método

La principal ventaja de dividir la actualización de la velocidad en dos mitades ( $\Delta t/2$ ) es que permite calcular la velocidad en el mismo instante de tiempo  $t + \Delta t$  que la posición. Esto es crucial para el cálculo preciso de propiedades termodinámicas instantáneas, como la energía cinética ( $E_k = \frac{1}{2}mv^2$ ) y la temperatura instantánea del sistema, sin necesidad de interpolaciones.

## 4. Implementacion en el codigo:

---

```

1  // -----
2  //  Implementacion en En bola.cpp
3  //  -----
4  // Paso medio de velocidad (Velocity-Verlet)
5  void Bola::vel_half_step(const Vec2& acc, double dt) {
6      vel.x += 0.5 * acc.x * dt;
7      vel.y += 0.5 * acc.y * dt;
8  }
9
10 // Paso completo de posición
11 void Bola::pos_full_step(double dt) {
12     pos.x += vel.x * dt;
13     pos.y += vel.y * dt;
14 }
15
16 // Completar la velocidad (segunda mitad del paso)
17 void Bola::vel_full_step(const Vec2& acc, double dt) {
18     vel.x += 0.5 * acc.x * dt;
19     vel.y += 0.5 * acc.y * dt;
20 }
21 // -----
22 //  Implementacion en En caja.cpp
23 //  -----
24 // FASE 1: Paso completo de Posición y medio de Velocidad
25 for (auto& b : bolas) {
26     b.pos.x += b.vel.x * dt + 0.5 * b.acc_old.x * dt * dt;
27     b.pos.y += b.vel.y * dt + 0.5 * b.acc_old.y * dt * dt;
28
29     b.vel.x += 0.5 * b.acc_old.x * dt;
30     b.vel.y += 0.5 * b.acc_old.y * dt;
31 }

```

---

## 5. Resultados simulación Potencial Lennard-Jones

### 5.1. Prueba 1

Condiciones iniciales:  $N = 100$ ,  $\epsilon = 1$ ,  $\sigma = 0,1$   
 $v = 0,01$

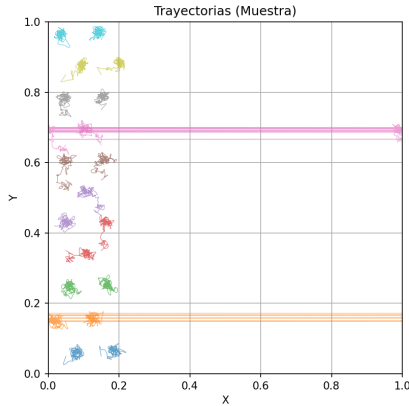


Figura 1: Trayectorias Prueba 1 correspondientes a las 20 primeras partículas

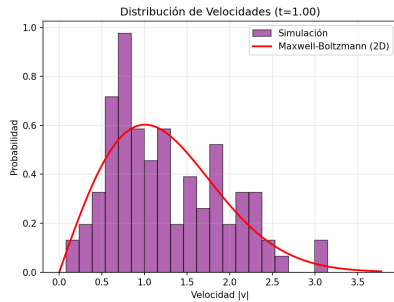


Figura 2: Comparación distribución de velocidades de la simulación y la teórica

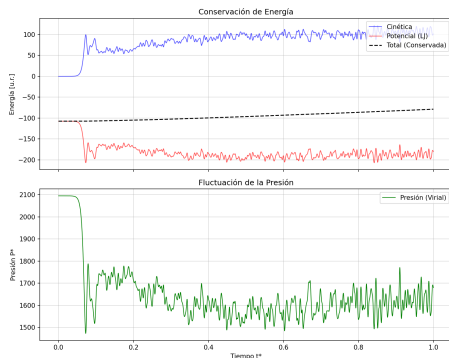


Figura 3: Gráficas energía y presión de Virial

### 5.2. Prueba 2

Condiciones iniciales:  $N = 500$ ,  $\epsilon = 1$ ,  $\sigma = 0,1$   
 $v = 0,04$

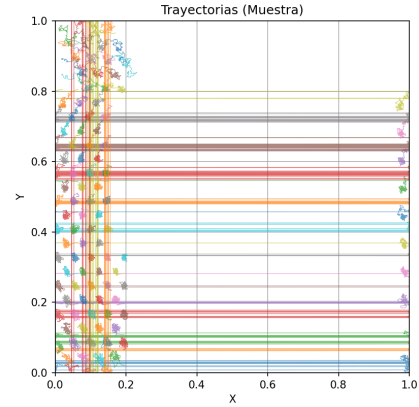


Figura 4: Trayectorias Prueba 2 correspondientes a las 20 primeras partículas

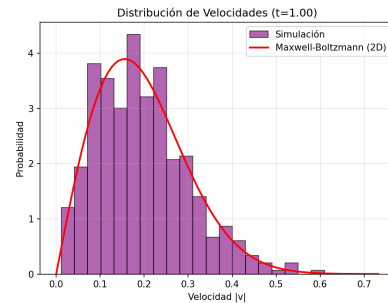


Figura 5: Comparación distribución de velocidades de la simulación y la teórica

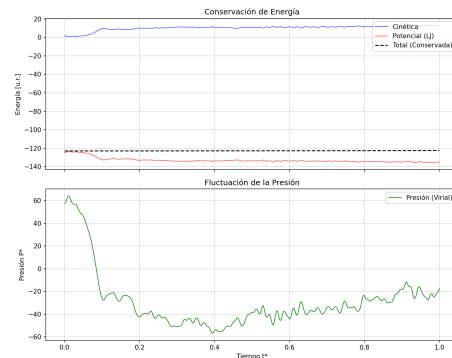


Figura 6: Gráficas energía y presión de Virial

## Simulación Motor Molecular DM dos estados:

### 6. Objetivo:

- Desarrollar e implementar un modelo computacional de un motor molecular de dos estados que simule el movimiento direccional de una partícula sometida a potenciales armónicos conmutados estocásticamente, considerando efectos térmicos y de fricción a través de la ecuación de Langevin.

## 7. Dinámica de Partículas Brownianas en Potenciales Conmutados

### 7.1. Ecuaciones de Movimiento

El sistema físico considerado consiste en una partícula browniana de masa  $m$  sometida a un potencial externo que conmuta entre dos estados. La dinámica está gobernada por la **ecuación de Langevin**:

$$m\ddot{x}(t) = -\gamma\dot{x}(t) - \nabla U(x, t) + \sqrt{2\gamma k_B T} \xi(t) \quad (15)$$

Esta ecuación modela el movimiento de la partícula bajo fuerzas inerciales ( $m\ddot{x}$ ), disipativas viscosas ( $-\gamma\dot{x}$ ), y externas, incluyendo el ruido térmico.

Donde los términos representan:

- $x(t)$ : Es la posición de la partícula en el tiempo  $t$ .
- $\gamma$ : Es el coeficiente de fricción del medio (viscosidad).
- $U(x, t)$ : Es el potencial externo dependiente del tiempo.
- $k_B T$ : Es la energía térmica del baño, donde  $k_B$  es la constante de Boltzmann y  $T$  la temperatura.
- $\xi(t)$ : Es un ruido gaussiano blanco que satisface las siguientes propiedades estadísticas del teorema de fluctuación-disipación:

$$\langle \xi(t) \rangle = 0, \quad \langle \xi(t) \xi(t') \rangle = \delta(t - t') \quad (16)$$

### 7.2. Potenciales Armónicos Conmutados

El potencial  $U(x, t)$  conmuta periódicamente entre dos estados de tipo armónico (parabólico) para modelar, por ejemplo, el mecanismo de trinquete (ratchet) de un motor molecular.

Los dos estados se definen mediante el potencial  $U(x, t)$  como:

$$U(x, t) = \begin{cases} U_0(x) = \frac{1}{2}k_0x^2 & \text{para el estado 0} \\ U_1(x) = \frac{1}{2}k_1(x - l)^2 & \text{para el estado 1} \end{cases} \quad (17)$$

Donde  $k_0$  y  $k_1$  son las constantes elásticas (rigidez) para cada estado, y  $l$  es la distancia de desplazamiento entre los mínimos de los dos potenciales.

### 7.3. Ciclo de Conmutación Química Determinista

Para modelar la dinámica de un motor molecular, cuyo ciclo de movimiento está típicamente impulsado por la hidrólisis de ATP (trifosfato de adenosina), se introduce un ciclo de conmutación química que acopla la dinámica del potencial a la variable de estado químico  $s(t)$ . Esta variable determina cuál de los potenciales ( $U_0$  o  $U_1$ ) rige el movimiento en un momento dado, simulando las etapas de unión y desunión del motor.

El potencial total  $U(x, t)$  queda entonces definido por la variable de estado:

$$U(x, t) = U_{s(t)}(x) \quad (18)$$

El cambio entre los estados  $s(t) \in \{0, 1\}$  sigue un ciclo periódico y **determinista** con un periodo total  $T_{\text{ciclo}}$ . La regla de conmutación es:

$$s(t) = \begin{cases} 0 & \text{si } t \bmod T_{\text{ciclo}} < T_{\text{off}} \\ 1 & \text{si } t \bmod T_{\text{ciclo}} \geq T_{\text{off}} \end{cases} \quad (19)$$

Donde las variables de tiempo son:

- $T_{\text{ciclo}} = T_{\text{off}} + T_{\text{on}}$ : El período total del ciclo de conmutación. La frecuencia del ciclo es  $\nu = 1/T_{\text{ciclo}}$ .
- $T_{\text{off}}$ : La duración del Estado 0 ( $s = 0$ ), que corresponde al potencial  $U_0(x)$  (unión débil o desunión), donde el motor puede difundir libremente.

- $T_{\text{on}}$ : La duración del **\*\*Estado 1\*\*** ( $s = 1$ ), que corresponde al potencial  $U_1(x)$  (unión fuerte), donde el motor está anclado y se prepara para el siguiente paso.
- $t \bmod T_{\text{ciclo}}$ : Es el resto de la división del tiempo  $t$  entre el período total  $T_{\text{ciclo}}$ , lo que garantiza la periodicidad.

Este ciclo determinista simula la frecuencia de hidrólisis de ATP y es el mecanismo fundamental que rompe la simetría térmica, permitiendo el movimiento direccional de la partícula browniana (el motor molecular).

## 8. Integración de las Ecuaciones de Langevin: Algoritmo BAOAB (Velocity Verlet Estocástico)

Para simular la dinámica del motor molecular bajo ruido térmico y fricción, se implementa una versión estocástica del algoritmo de Velocity Verlet conocida como **esquema BAOAB**. Este método integra correctamente la ecuación de Langevin manteniendo la relación fluctuación-disipación y con un error local de orden  $\mathcal{O}(\Delta t^2)$ .

### 8.1. Esquema BAOAB para Ecuaciones de Langevin

El esquema BAOAB divide cada paso temporal  $\Delta t$  en cinco sub-pasos que manejan separadamente la parte determinista (fuerza) y estocástica (fricción + ruido). Para nuestro sistema unidimensional con masa  $m = 1$ , el algoritmo discretizado es:

#### 8.1.1. Paso B: Medio paso de fuerza (determinista)

Primero, avanzamos las velocidades medio paso usando la fuerza determinista del potencial:

$$v \leftarrow v + \frac{F(x)}{2m} \Delta t \quad (20)$$

donde  $F(x) = -\nabla U(x, t)$  es la fuerza instantánea.

#### 8.1.2. Paso A: Fricción y ruido (estocástico)

Aplicamos la fricción y el ruido térmico, que en el régimen sobreamortiguado se combinan como:

$$c_1 = e^{-\gamma \Delta t} \quad (\text{factor de amortiguamiento}) \quad (21)$$

$$\sigma = \sqrt{k_B T (1 - c_1^2)} \quad (\text{amplitud del ruido}) \quad (22)$$

$$v \leftarrow c_1 v + \sigma R \quad (23)$$

con  $R \sim \mathcal{N}(0, 1)$  siendo un número aleatorio gaussiano.

#### 8.1.3. Paso O: Paso completo de posición (cinemático)

Actualizamos la posición usando la velocidad actualizada:

$$x \leftarrow x + v \Delta t \quad (24)$$

#### 8.1.4. Paso A: Fricción y ruido (estocástico, repetido)

Aplicamos nuevamente fricción y ruido con el mismo factor  $c_1$  y nuevo ruido  $R'$ :

$$v \leftarrow c_1 v + \sigma R' \quad (25)$$

#### 8.1.5. Paso B: Medio paso de fuerza (determinista, final)

Completamos el paso temporal con la otra mitad de la fuerza, ahora evaluada en la nueva posición:

$$v \leftarrow v + \frac{F(x)}{2m} \Delta t \quad (26)$$

## 8.2. Propiedades del Algoritmo BAOAB

### 8.2.1. Conservación de la Relación Fluctuación-Disipación

El esquema BAOAB garantiza que se mantenga correctamente:

$$\langle v^2 \rangle = \frac{k_B T}{m} \quad (\text{teorema de equipartición}) \quad (27)$$

incluso para pasos temporales  $\Delta t$  relativamente grandes.

### 8.2.2. Estabilidad con Potenciales Rígidos

Para nuestro motor con  $k_1 = 10,0 \gg k_0 = 0,1$ , el algoritmo es estable porque:

- Los términos estocásticos se aplican en el "medio" del paso
- Las fuerzas grandes no desestabilizan la integración
- La fricción  $\gamma = 10,0$  amortigua oscilaciones numéricas

### 8.2.3. Exactitud para Sistemas Armónicos

Para un potencial armónico  $U(x) = \frac{1}{2}kx^2$ , el algoritmo BAOAB reproduce exactamente la distribución de equilibrio de Boltzmann:

$$P(x) \propto \exp\left(-\frac{kx^2}{2k_B T}\right) \quad (28)$$

## 8.3. Ventajas sobre Otros Esquemas

### ▪ Algoritmo de Euler:

- Ventaja: Es el más **simple** y rápido de implementar.
- Desventaja: Es numéricamente **inestable** y no conserva la energía a largo plazo. Requiere pasos de tiempo extremadamente pequeños ( $\Delta t$ ) para ser preciso.

### ▪ Position Verlet (Verlet de Posición):

- Ventaja: Es **simpéctico** (conserva la energía total), ideal para sistemas hamiltonianos.
- Desventaja: En su forma estándar, no está diseñado para manejar fuerzas disipativas ( $\gamma$ ) ni el ruido térmico ( $\xi(t)$ ) de la Ecuación de Langevin.

### ▪ Leapfrog (Salto de Rana):

- Ventaja: Es un algoritmo **sincronizado** (como Velocity Verlet) y reversible, con buen rendimiento.
- Desventaja: Difícil de adaptar para incluir la fricción y el ruido térmico de manera precisa y estable, propiedades esenciales en este modelo browniano.

### ▪ BAOAB (Biot-Adjoint Operator Based):

- Ventaja: Es **simpéctico** y **mantiene la relación fluctuación-disipación** (conserva la temperatura y el equilibrio).
- Desventaja: Es **óptimo para la Ecuación de Langevin** y sistemas termostáticos, superando las deficiencias de los otros métodos.

## 9. Implementación en el código:

---

```
1 // -----
2 //  Implementacion en En integrador.cpp
3 // -----
4 // Implementación de Stochastic Velocity Verlet (tipo BAOAB)
5 // Este método integra la Ecuación de Langevin correctamente.
6 void StochasticVelocityVerletIntegrator::step(MotorModel& motor, double dt) {
7     Particle& p = motor.getParticle();
8     double m = p.m;
9     double gamma = motor.getGamma();
10    double kBT = motor.getKBT();
11
12    // --- Coeficientes del paso de Fricción/Ruido (A) ---
13    // c1: Factor de amortiguamiento de la fricción (Friction factor)
14    double c1 = std::exp(-gamma * dt);
15    // sigma: Amplitud del ruido Gaussiano (Noise amplitude)
16    // Se asegura la relación Fluctuation-Dissipation: sigma = sqrt(kBT * (1-c1^2))
17    double sigma = std::sqrt(kBT * (1.0 - c1 * c1));
18
19    // --- 1. Paso B (Force Half-Step) ---
20    // Aplicar la mitad de la fuerza determinista (potencial)
21    double F_old = motor.force(0.0);
22    p.v += F_old / m * (0.5 * dt);
23
24    // --- 2. Paso A (Friction & Noise) ---
25    // Aplicar fricción y ruido (Movement, Friction, Noise)
26    double R_t = motor.generateGaussianNoise(); // Ruido Gaussiano
27    p.v = c1 * p.v + sigma * R_t / std::sqrt(m);
28
29    // --- 3. Paso D (Position Full-Step) ---
30    // Actualizar la posición
31    p.x += p.v * dt;
32
33    // --- 4. Paso B (Force Half-Step) ---
34    // Recalcular la fuerza con la nueva posición y aplicar la otra mitad
35    // Nota: El potencial ya fue conmutado por updateChemicalState() en Simulator.run()
36    double F_new = motor.force(0.0);
37    p.v += F_new / m * (0.5 * dt);
38 }
```

---



## 10. Resultados simulación

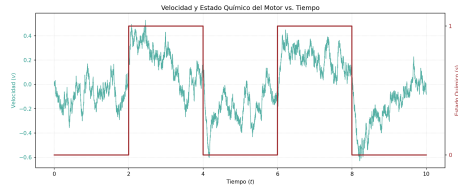


Figura 7: Velocidad y estado químico

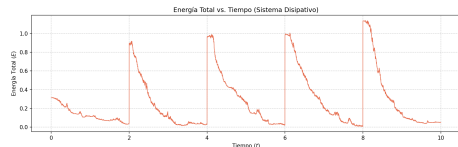


Figura 8: Energía total

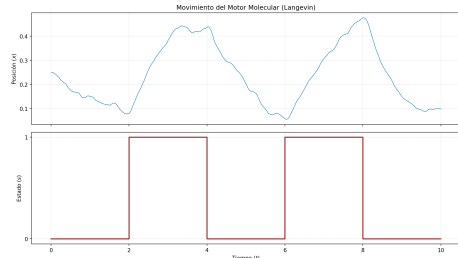


Figura 9: Movimiento del motor molecular

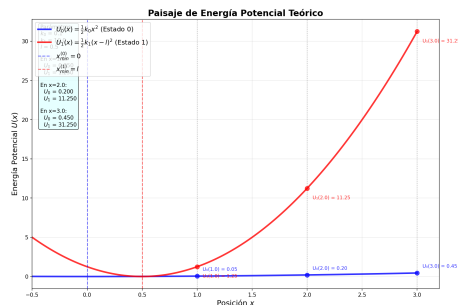


Figura 10: Potencial teórico

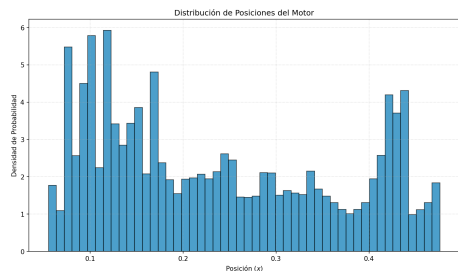


Figura 11: Distribución de posiciones

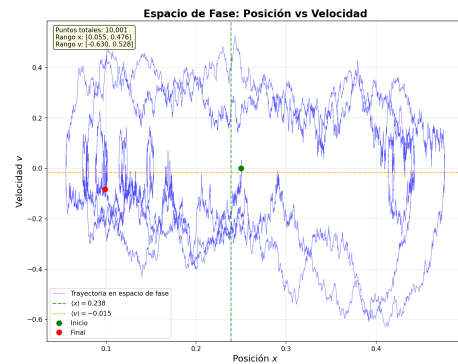


Figura 12: Espacio de fase

## Referencias

- [1] Sztrajman, A. (2014). *Simulación MD de un fluido de Lennard-Jones*. Informe de Proyecto no Publicado, University College London.
- [2] Chaturvedi, H. (2015). Lennard-Jones gas Molecular Dynamics Simulation. *Harshwardhan Chaturvedi*. [https://youtu.be/MGTzTvG\\_OZw](https://youtu.be/MGTzTvG_OZw)
- [3] Berglund, N. (2022). Under pressure: Particles with Lennard-Jones interaction in a piston. *Nils Berglund*. <https://youtu.be/P2Vw9CWD8YM>
- [4] Haile, J. M. (1992). *Molecular Dynamics Simulation: Elementary Methods*. John Wiley & Sons, Inc., New York.
- [5] Stickler, B. A., Schachinger, E. (2016). *Basic Concepts in Computational Physics* (2<sup>nd</sup> ed.). Springer International Publishing, Switzerland.
- [6] Mukherjee, R., Soppina, P., Patel, N. M., Soppina, V., & Rane, K. (2022). Effect of Binding-Affinity and ATPase Activity on the Velocities of Kinesins Using Ratchet Models. *Cell Biochemistry and Biophysics*, 80(31–38). <https://doi.org/10.1007/s12013-021-01057-x>
- [7] Aboelkassem, Y., McCabe, K. J., Huber, G. A., Regnier, M., McCammon, J. A., & McCulloch, A. D. (2019). A Stochastic Multiscale Model of Cardiac Thin Filament Activation Using Brownian-Langevin Dynamics. *Biophysical Journal*, 117(1–18). <https://doi.org/10.1016/j.bpj.2019.08.003>