

Dinámica Molecular regida por el paso temporal

Grupo 7
Duffau, Teófilo Manuel
Lynch, Ezequiel



Fundamentos



Fundamentos

2 sistemas simulados:

- Oscilador Puntual Amortiguado: Al tener una solución analítica, se puede utilizar para comparar el error de distintos esquemas de integración.
- Gas de Lennard-Jones: Simulación compleja de partículas de un gas en un recipiente cerrado cuyo resultado analítico sería muy costoso de calcular.



Implementación



Oscilador Amortiguado

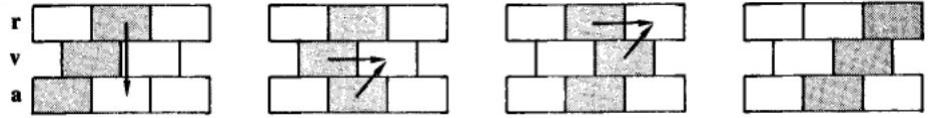
Ya cuenta con solución analítica:

$$y = Ae^{-\frac{b}{2m}t} \cos(\omega t + \phi) \quad \omega = \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2}$$

Esquemas de integración implementados:

- **Verlet Leap-Frog** usando cálculo de velocidades intermedias.
- **Beeman (Variante predictor-corrector)** ya que la fuerza depende de la velocidad.
- **Gear Predictor-Corrector** de orden 5.

Verlet Leap-Frog



$$\mathbf{v}_i(t + \frac{\Delta t}{2}) = \mathbf{v}_i(t - \frac{\Delta t}{2}) + \frac{\Delta t}{m_i} \mathbf{f}_i(t),$$

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t + \frac{\Delta t}{2}).$$

$$\mathbf{v}_i(t) = \frac{\mathbf{v}_i(t - \frac{\Delta t}{2}) + \mathbf{v}_i(t + \frac{\Delta t}{2})}{2}.$$



Beeman (Variante predictor-corrector)

A predictor-corrector variant is useful when the forces are velocity-dependent:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{2}{3}a(t)\Delta t^2 - \frac{1}{6}a(t - \Delta t)\Delta t^2 + O(\Delta t^4).$$

The velocities at time $t = t + \Delta t$ are then calculated from the positions.

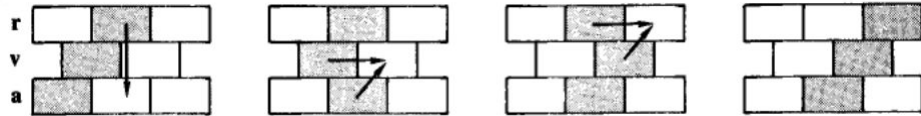
$$v(t + \Delta t)_{(\text{predicted})} = v(t) + \frac{3}{2}a(t)\Delta t - \frac{1}{2}a(t - \Delta t)\Delta t + O(\Delta t^3)$$

The accelerations at time $t = t + \Delta t$ are then calculated from the positions and predicted velocities.

$$v(t + \Delta t)_{(\text{corrected})} = v(t) + \frac{1}{3}a(t + \Delta t)\Delta t + \frac{5}{6}a(t)\Delta t - \frac{1}{6}a(t - \Delta t)\Delta t + O(\Delta t^3)$$

Lennard Jones Gas

Para simular el gas de Lennard Jones, se utilizó la versión de leapfrog de Verlet ya que en este se calcula la velocidad, dato necesario para los resultados.



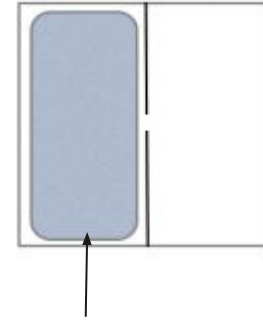
Para el cálculo de las fuerzas de potencial y para las fuerzas ejercidas sobre cada partícula se utilizó el cell index method para calcular los vecinos de cada partícula.

Lennard Jones Gas

La simulación cuenta con los siguientes pasos:

1. Se generan las partículas con una distribución uniforme en el recinto izquierdo y separadas entre sí para que no ocurrieran explosiones instantáneas por proximidad y con distribución de velocidad uniforme.
2. Se calculan los vecinos de cada partícula y se guardan en ella.
3. Se calculan las fuerzas que se ejercen sobre cada partícula y el potencial de LJ en cada partícula
4. Con la fuerza calculada se actualizan las posiciones y las velocidades de las partículas.
5. Si se tiene que guardar el estado, solo se guardan 30 estados por segundo real, se guarda y si no se retorna al paso 2.
6. La condición de corte ocurre cuando por primera vez hay mayor cantidad de partículas en el recinto derecho que en el izquierdo.

Estado Inicial





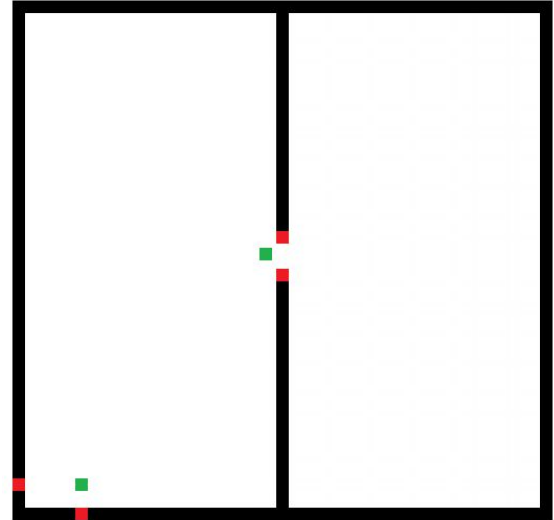
Implementación

Se cuenta con la clase `WeightedParticle` que es sobre la cual se calculan las posiciones, las velocidades, las fuerzas, etc., con la clase `LennardJonesGrid` que es la encargada de mantener el cell index method y calcular los vecinos y la clase `LennardJonesSimulator` que se encarga de realizar la simulación



Implementación

Para las paredes, se implementó que si una partícula se encuentra a una distancia de ellas menor al corte de potencial se hace de cuenta que la pared más cercana (horizontal y verticalmente), o las puntas del tabique central si se encuentra en ese espacio, es una partícula como cualquier otra y le aplica su fuerza correspondiente.





Implementación

Para el cálculo de la energía potencial se utilizó la siguiente ecuación por cada vecina i:

$$V_i = \epsilon \left[\left(\frac{r_m}{r} \right)^{12} - 2 \left(\frac{r_m}{r} \right)^6 \right]$$

Para el cálculo de fuerzas ejercidas sobre una partícula, la sumatoria de cada vecina suya i

$$F_i = \frac{12\epsilon}{r_m} \left[\left(\frac{r_m}{r} \right)^{13} - \left(\frac{r_m}{r} \right)^7 \right]$$

donde r es la distancia entre partículas en ambos casos.



Resultados

Oscilaciones amortiguadas



De la teoria:

Parámetros

$$m = 70 \text{ kg}; \quad k = 10^4 \text{ N/m};$$

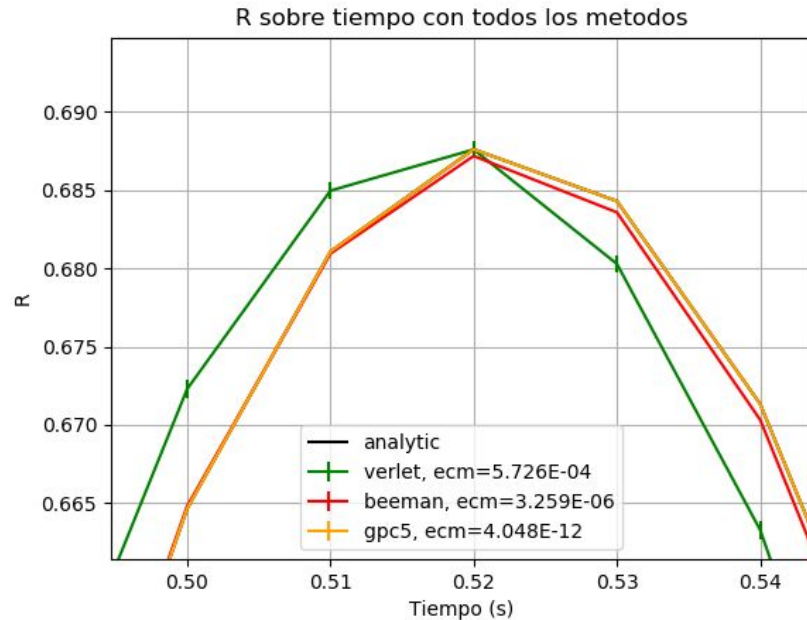
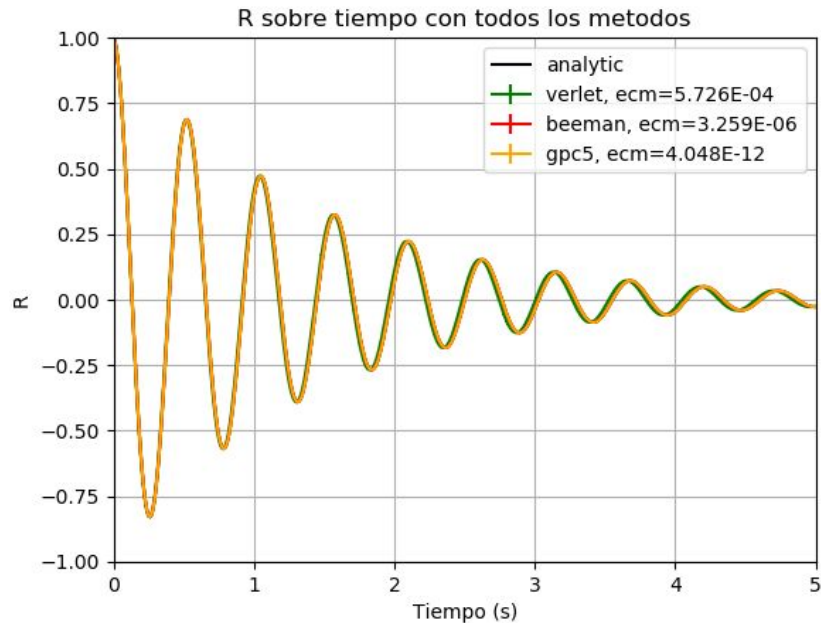
$$b = 100 \text{ kg/s}; \quad t_f = 5 \text{ s}$$

Condiciones Iniciales

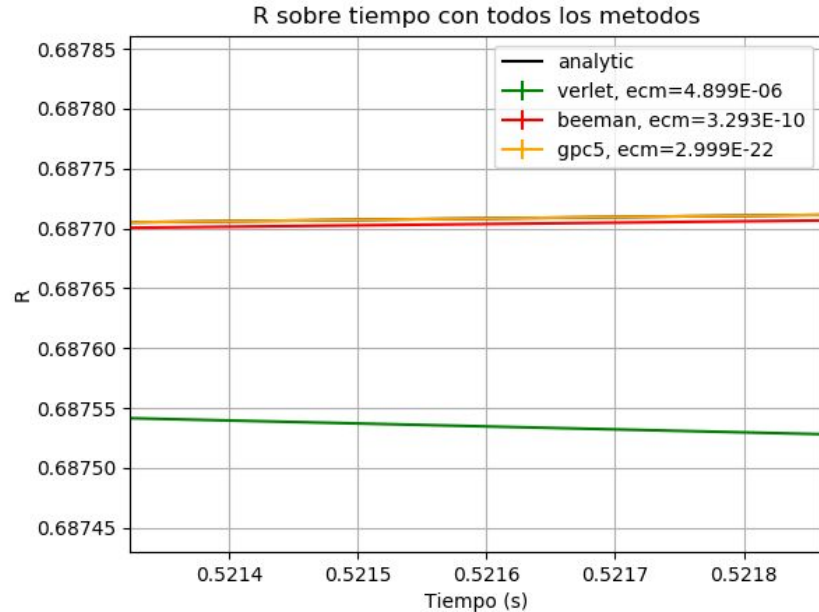
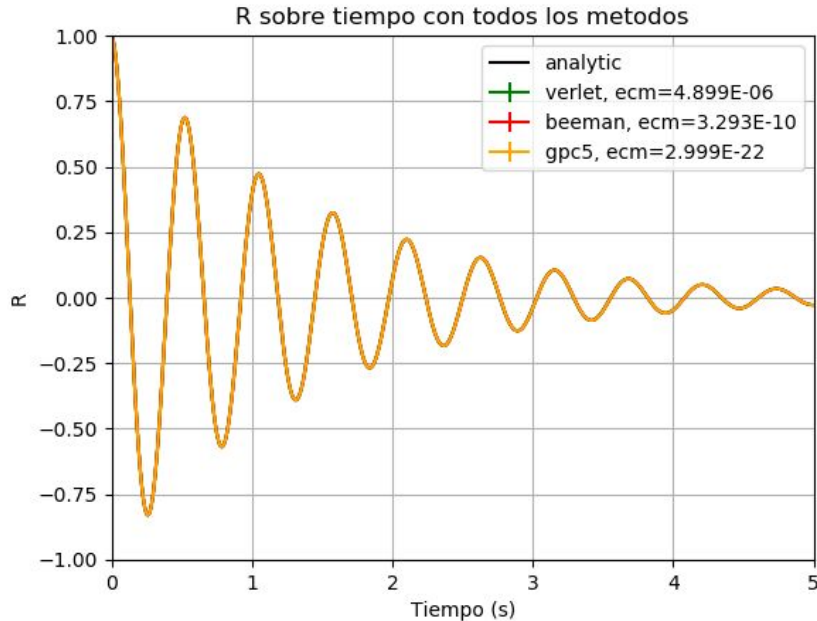
$$r(t=0) = 1 \text{ m};$$

$$v(t=0) = -A b/(2m) \text{ m/s};$$

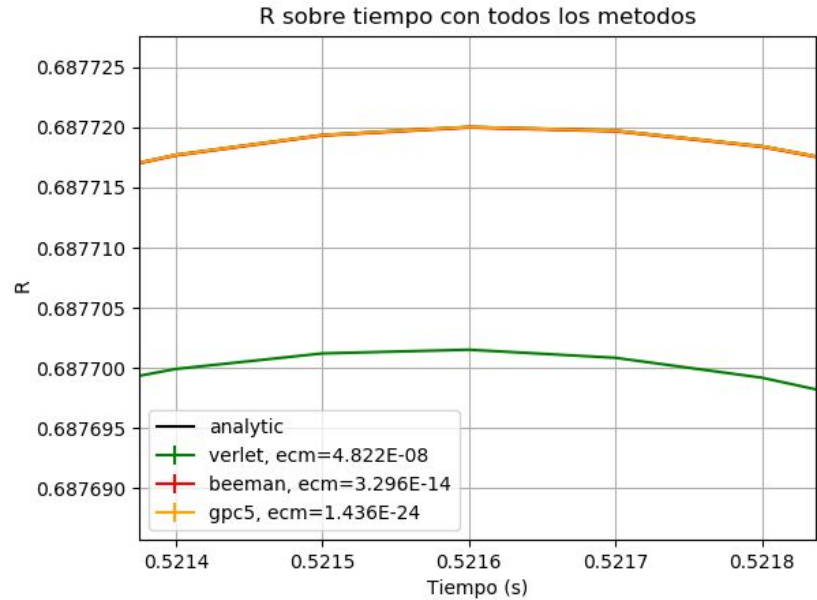
Oscilaciones amortiguadas con $dt=0.01s$



Oscilaciones amortiguadas con $dt=0.001s$

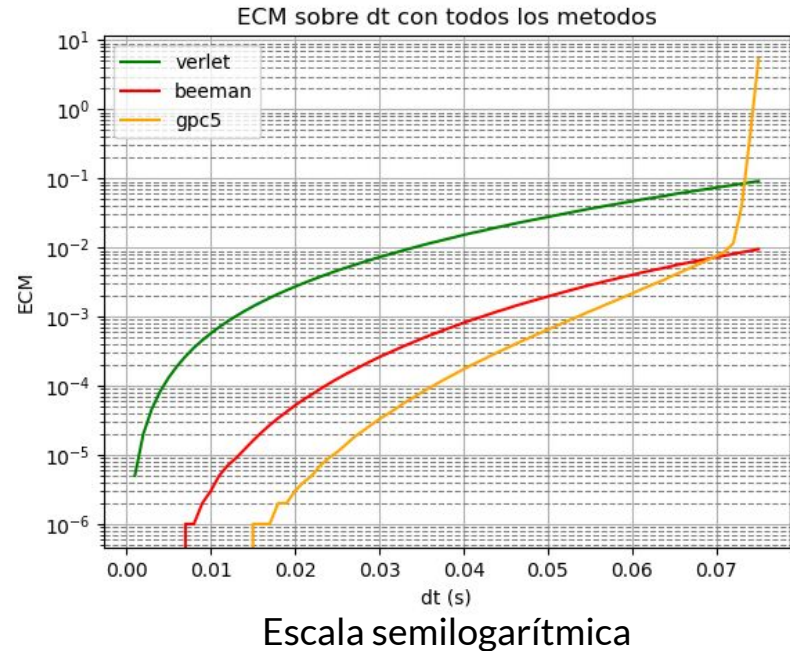
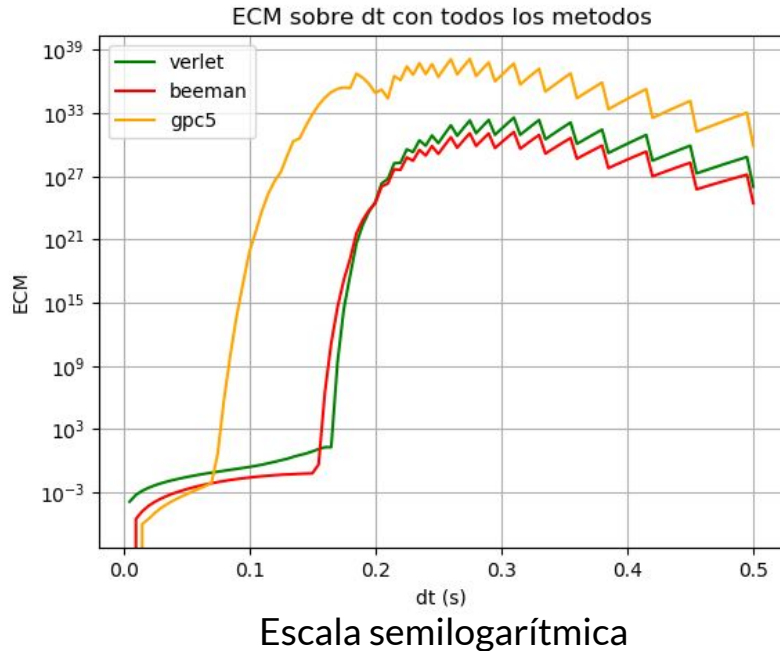


Oscilaciones amortiguadas con $dt=0.0001s$



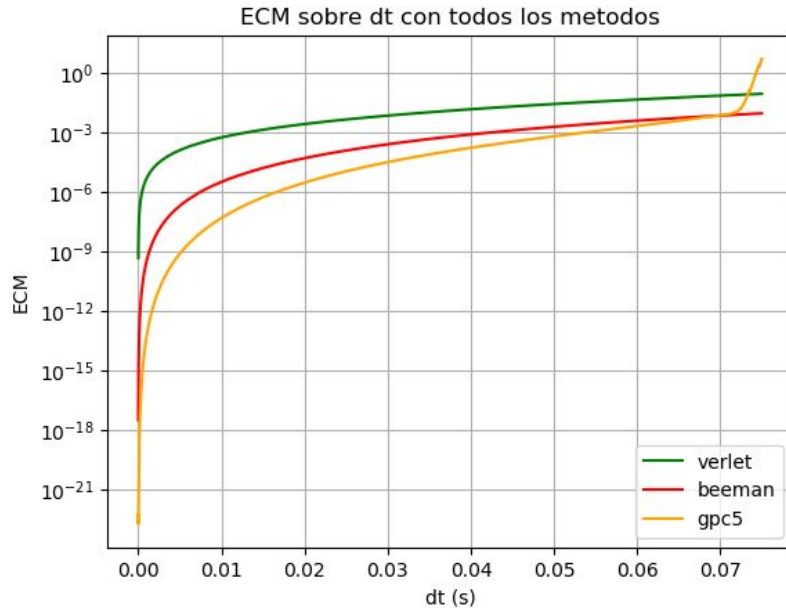
Error cuadrático medio en oscilaciones

Aumento dt en 0.001

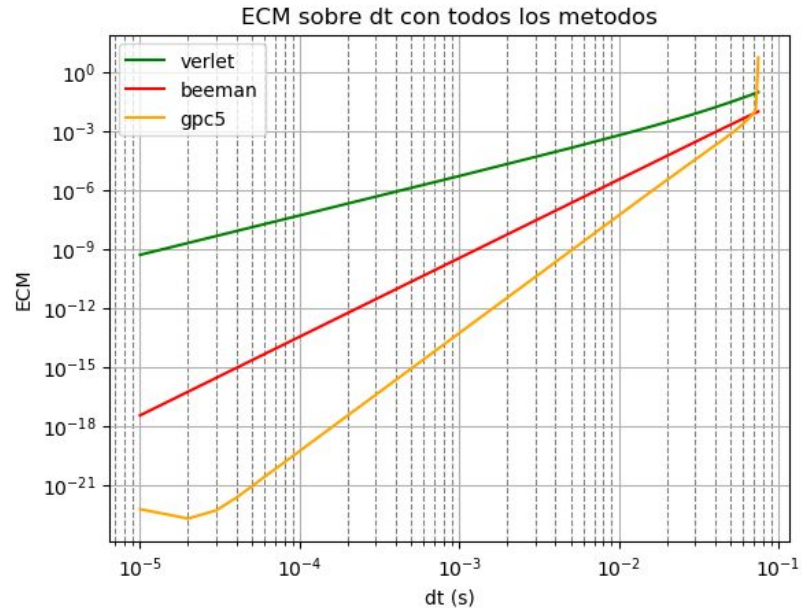


Error cuadrático medio en oscilaciones

Aumento dt en 0.00001

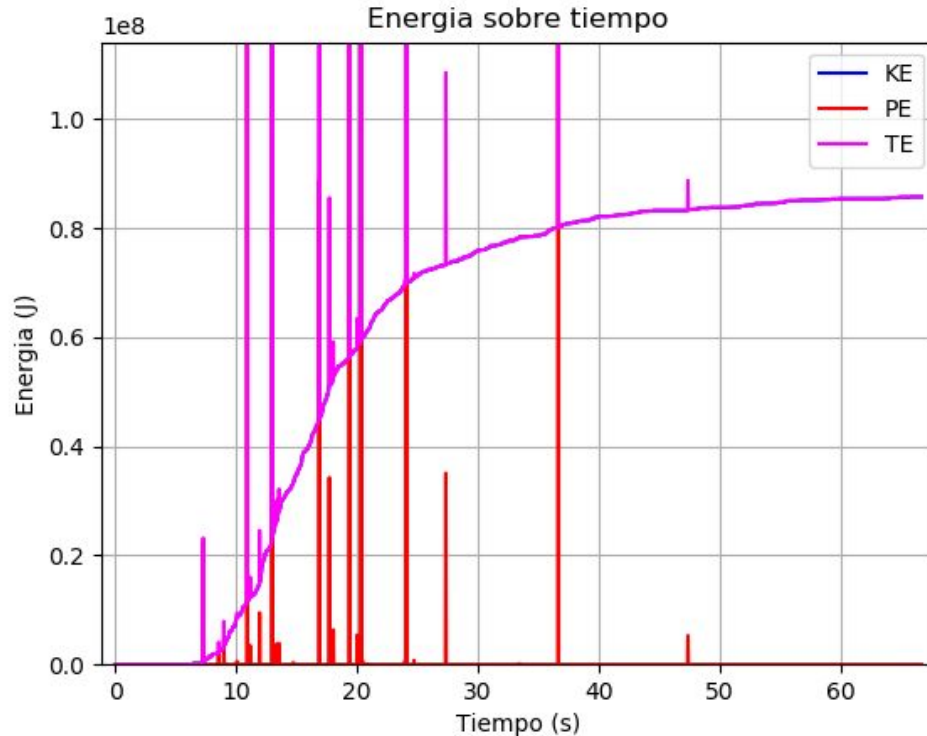


Escala semilogarítmica



Escala logarítmica

Pruebas de tiempo delta (0.005 s)

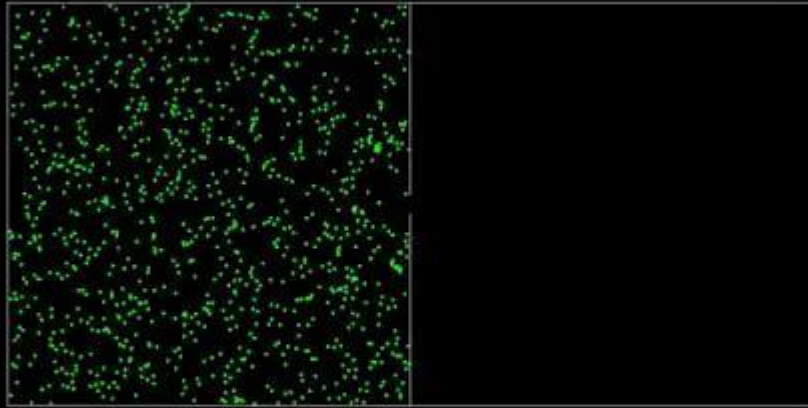


$N = 1000$
 $r_m = 1$
 $\varepsilon = 2$
 $m = 0.1$
 $|v_i| = 10$

Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a 2000/30 segundos ≈ 66 segundos

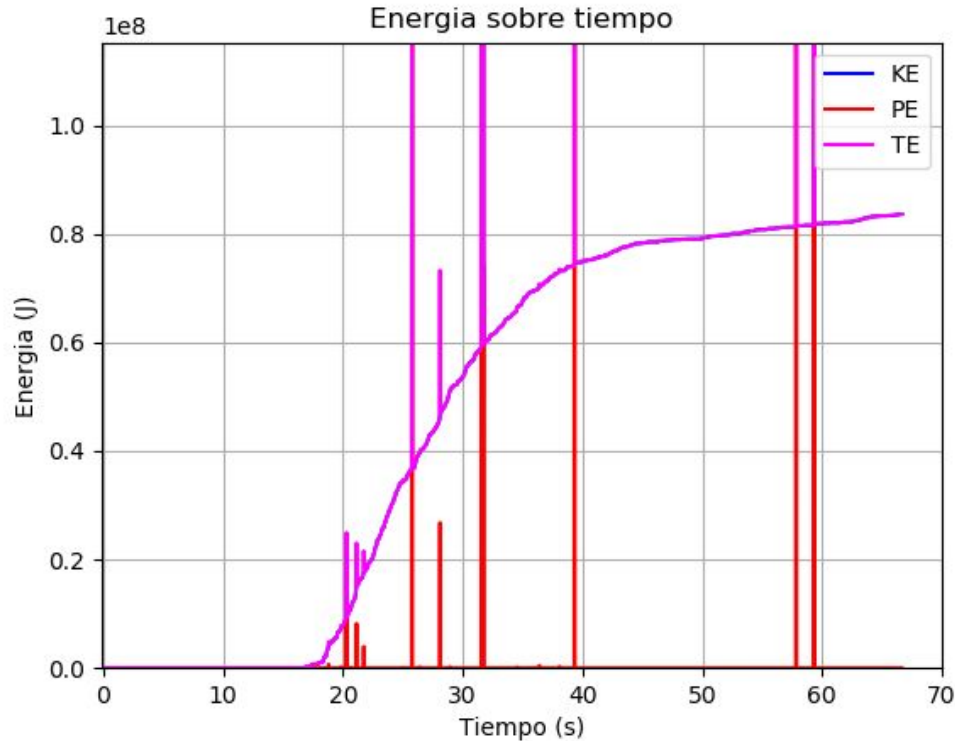
Pruebas de tiempo delta (0.005 s) - Explosiones



Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a $2000/30$ segundos ≈ 66 segundos

Pruebas de tiempo delta (0.004 s)



$N = 1000$
 $r_m = 1$
 $\varepsilon = 2$
 $m = 0.1$
 $|v_i| = 10$

Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a $2000/30$ segundos ≈ 66 segundos

Pruebas de tiempo delta (0.004 s) - Explosiones

Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a $2000/30$ segundos ≈ 66 segundos

Pruebas de tiempo delta (0.003 s)

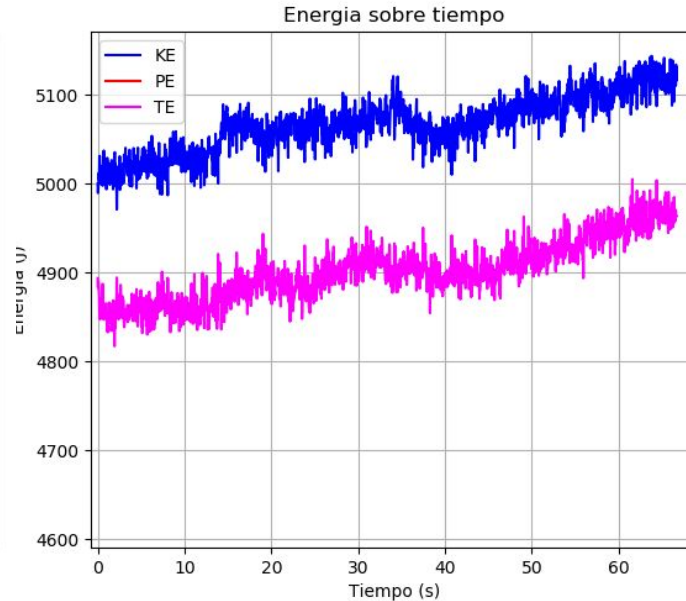
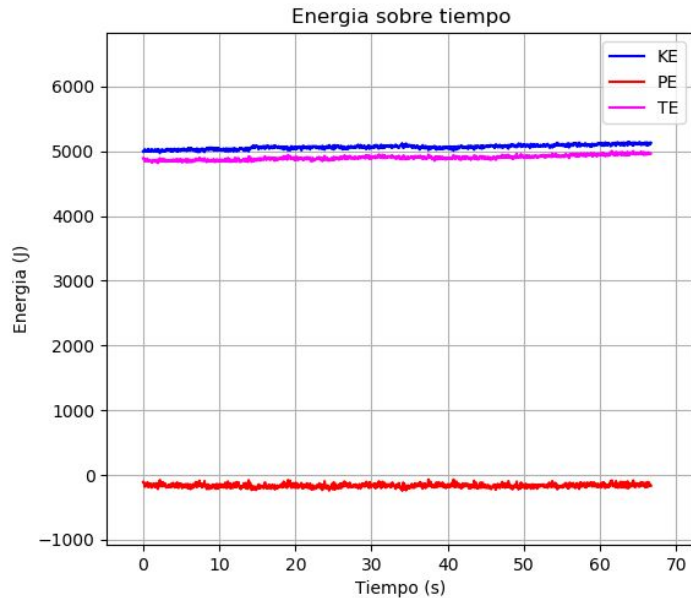
$N = 1000$

$r_m = 1$

$\varepsilon = 2$

$m = 0.1$

$|v_i| = 10$



Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a 2000/30 segundos ≈ 66 segundos

Pruebas de tiempo delta (0.002 s)

$$N = 1000$$

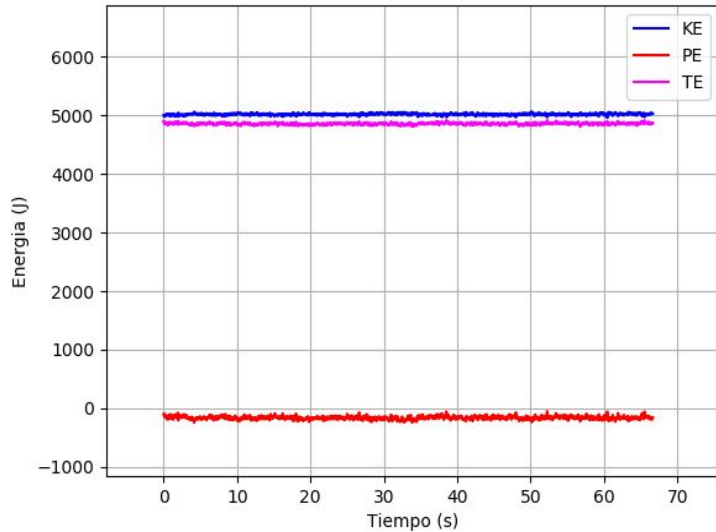
$$r_m = 1$$

$$\varepsilon = 2$$

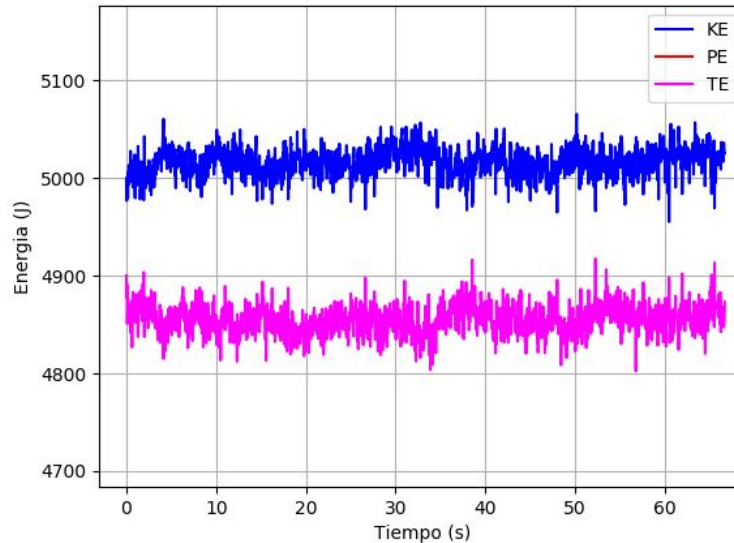
$$m = 0.1$$

$$|v_i| = 10$$

Energía sobre tiempo



Energía sobre tiempo



Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a 2000/30 segundos \approx 66 segundos

Pruebas de tiempo delta (0.001 s)

$$N = 1000$$

$$r_m = 1$$

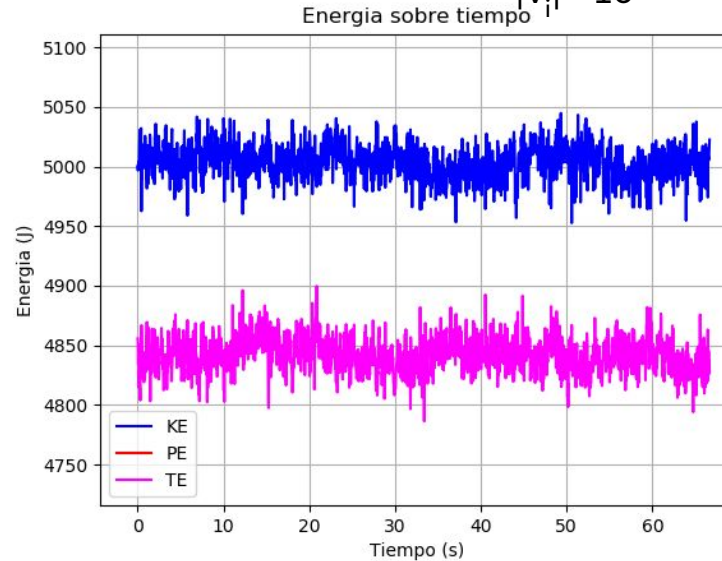
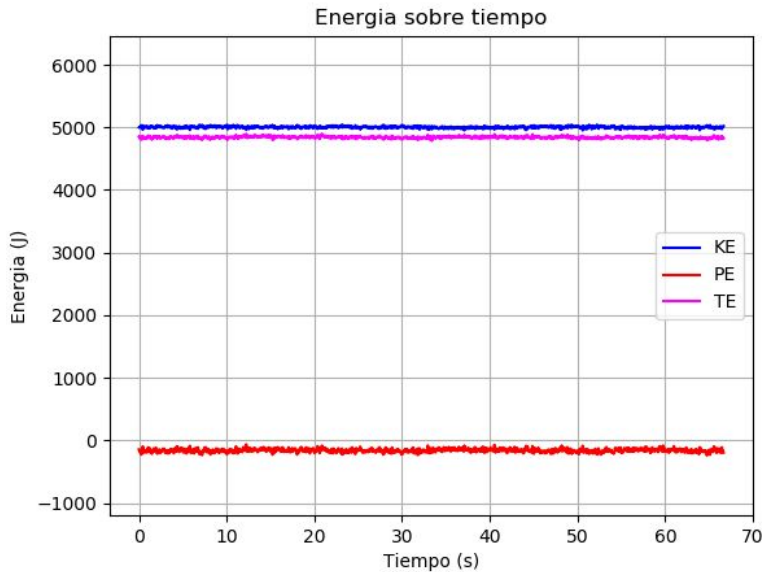
$$\varepsilon = 2$$

$$m = 0.1$$

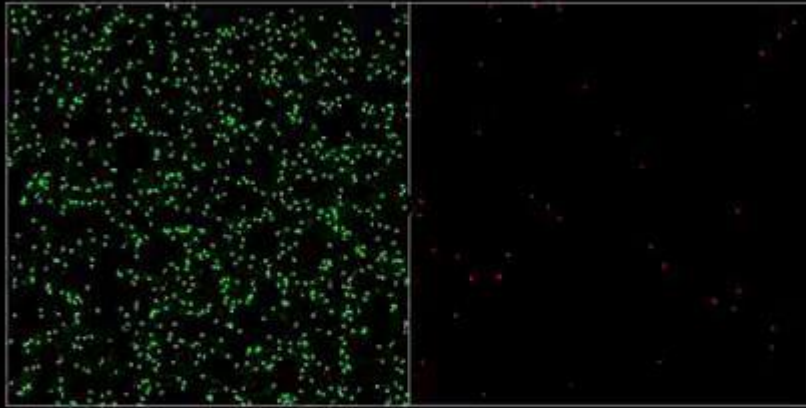
$$|v_i| = 10$$

Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a 2000/30 segundos ≈ 66 segundos



Pruebas de tiempo delta (0.001 s)



Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a $2000/30$ segundos ≈ 66 segundos

Pruebas de tiempo delta (0.001 s - 0.005 s)



Los tiempos 0.005s y 0.004s no son utilizables ya que al ser muy grandes para el sistema, generan explosiones como las vistas en las animaciones.

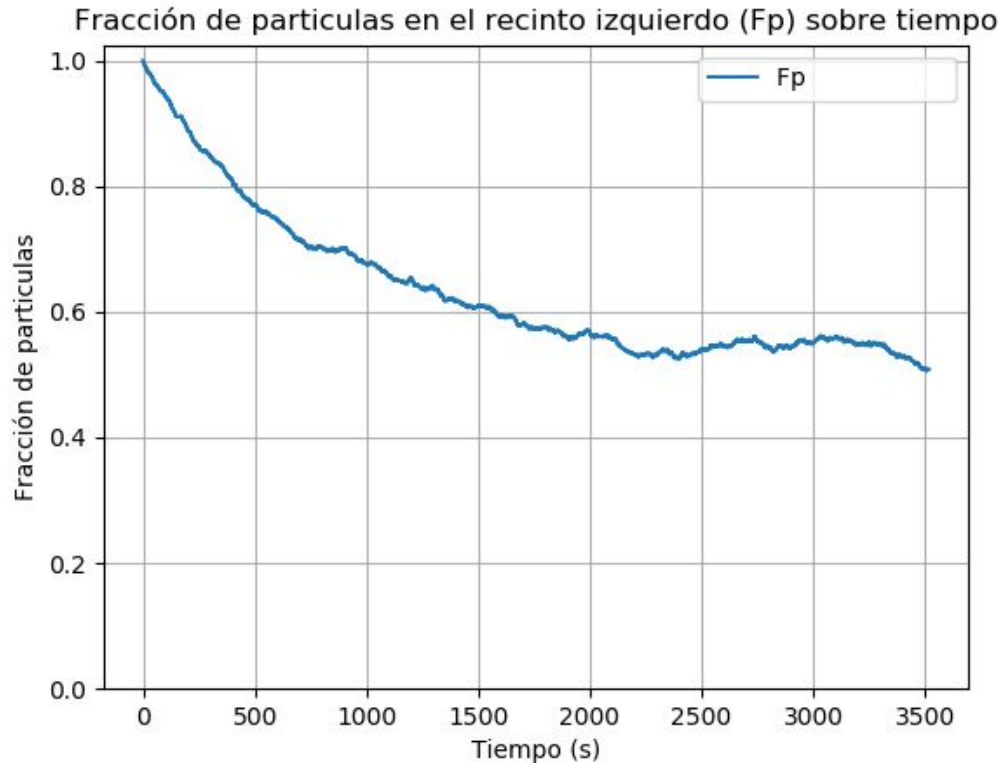
El tiempo 0.003s demuestra un aumento de la energía cinética, cosa que no debería ocurrir, por lo que aún es muy grande para utilizar.

Se decidió utilizar como tiempo delta 0.001s ya que por seguridad tomamos el segundo tiempo cuya energía total se mantiene estable comparada con el anterior. No tendría sentido seguir disminuyendo el delta tiempo si ya con 0.001s no hay problemas.

Para todas las simulaciones se utilizó el integrador de leapfrog verlet.

Para la elección del tiempo delta se corrieron pruebas de 2000 frames (guardando 30 por segundo real) que equivaldrían a $2000/30$ segundos ≈ 66 segundos

Fracción de partículas en el recinto izquierdo



$N = 1000$

$r_m = 1$

$\varepsilon = 2$

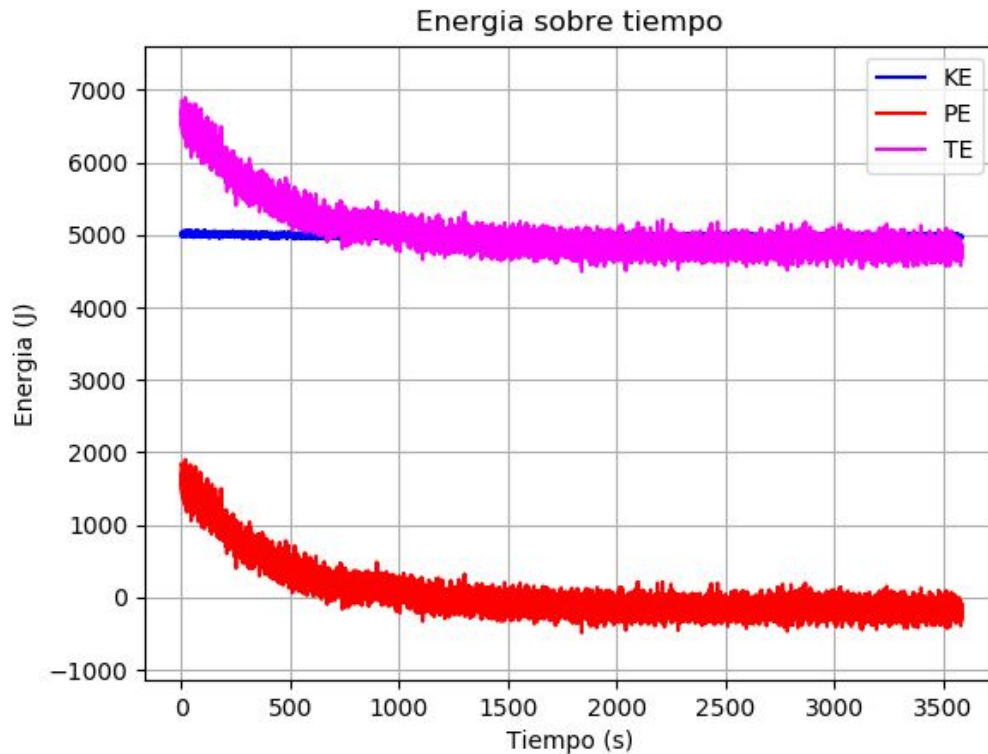
$m = 0.1$

$|v_i| = 10$

$dT = 0.001s$

Condición de corte: $Fp < 0.5$

Energía Total



$N = 1000$

$r_m = 1$

$\varepsilon = 2$

$m = 0.1$

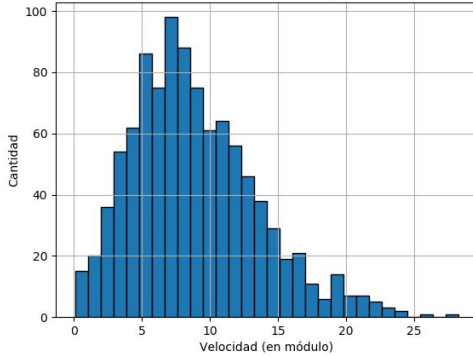
$|v_i| = 10$

$dT = 0.001s$

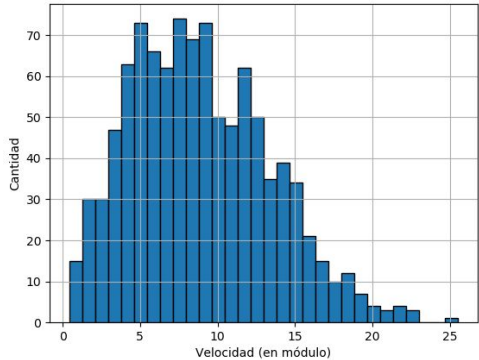
Condición de corte: $F_p < 0.5$

Distribución de velocidades

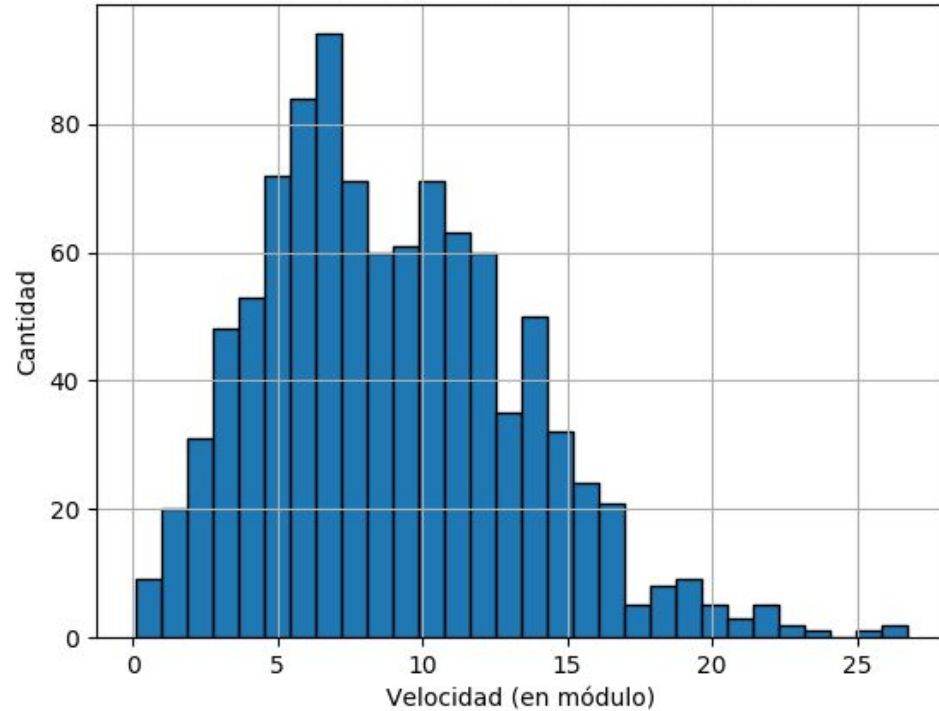
Distribución de Velocidades al primer tercio



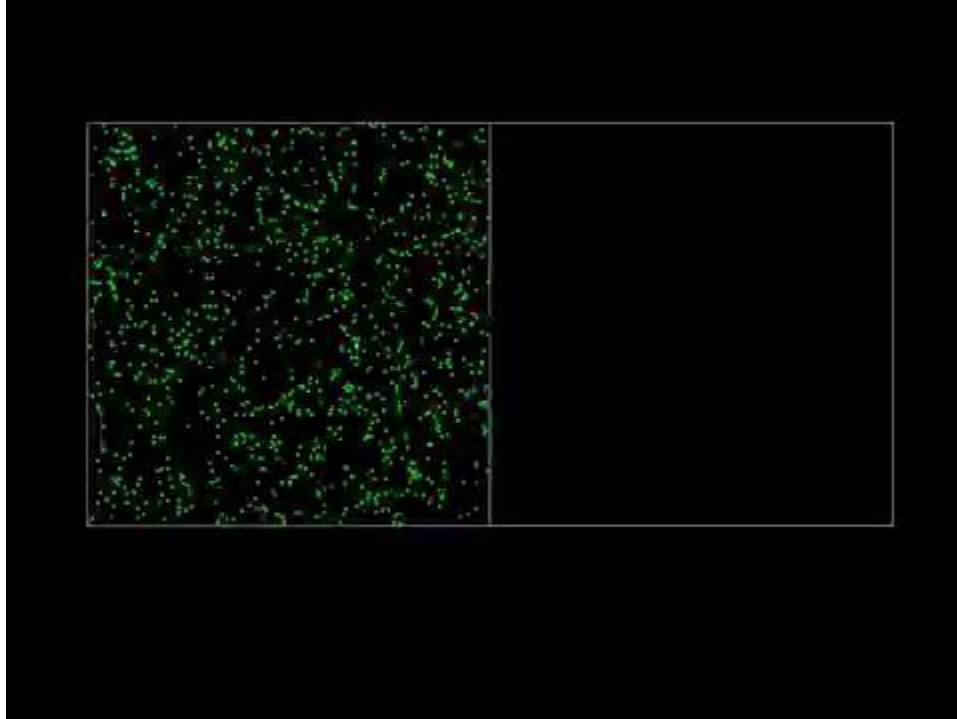
Distribución de Velocidades al segundo tercio



Distribución de Velocidades al final



Animación completa





Conclusiones



Conclusiones

- GPC5 es el mas exacto, pero tambien es el mas lento.
- LeapFrog el más rápido, pero el que tiene mayor error.
- A menor tiempo delta, se observa menos errores de cálculos.
- La energía potencial a largo plazo disminuye hasta quedar estable.
- Con las condiciones iniciales dadas, un sistema de 1000 partículas se demora alrededor de una hora de tiempo real en equilibrarse en el recipiente.
- Las distribuciones de velocidad se equilibran antes del primer tercio de la simulación.