Flujos Granulares 3D:
Caída de una bala de cañón

Alumnos
Tomas Dorado, 56594
Tomas Dallas, 56436

Grupo 6

Introducción

Introducción

- Se estudió el comportamiento de flujos granulares tridimensionales, en particular: la caída de una bala de cañón sobre un lecho granular.
- Se realizaron distintas simulaciones parametrizando las corridas para analizar las ventajas y desventajas del modelo propuesto.

Sistema real - Fundamentos

- El sistema propuesto consiste de un cajón cuadrado lleno de partículas de distintos tamaños y centrada en el mismo, impacta una bala de cañón sobre las partículas.
- Las partículas colisionan entre sí en las 3 dimensiones definidas, como así también con las paredes que las contienen y la misma bala.
- Cuando una partícula colisiona con otra partícula o una pared se ejerce una fuerza en sentido normal y otra en sentido tangencial al trayecto de la misma que, en consecuencia, modifica la energía cinética de ella.

Sistema real - Ecuaciones

Fuerzas de las colisiones

$$\overline{Fn_{ij}} = -k_n \xi - \gamma v_{ij}^{\hat{n}} \qquad (1)$$

$$\overline{Ft_{ij}} = -k_t \xi \qquad (2)$$

Fuerza de gravedad

$$\overline{Fg_i} = m_i \overline{g}$$

Total de fuerzas

$$\overline{F_i} = \overline{Fg_i} + \sum_j \overline{F_{ij}} \qquad (5)$$

$$v_{ij}^{\hat{n}} = \overline{v_{ij}} \cdot \widehat{r_{ij}} \qquad (3)$$

$$\overline{F_{ij}} = \overline{Fn_{ij}} \cdot \widehat{r_{ij}} + \overline{Ft_{ij}} \cdot \widehat{v_{ij}} \qquad (4)$$

siendo: $v_{ij}^{\hat{t}} = \overline{v_{ij}} - v_{ij}^{\hat{n}} \cdot \widehat{r_{ij}}$ el cálculo del versor $\widehat{v_{ij}} = \frac{v_{ij}^{\hat{t}}}{\left|v_{ij}^{\hat{t}}\right|}$, la velocidad relativa $\overline{v_{ij}} = \overline{v_i} - \overline{v_j}$, el versor normal $\widehat{r_{ij}} = \frac{\overline{r_{ij}}}{\left|\overline{r_{ij}}\right|}$ y por último $\overline{r_{ij}} = \overline{r_i} - \overline{r_j}$

Aceleración dada por la segunda ley de Newton

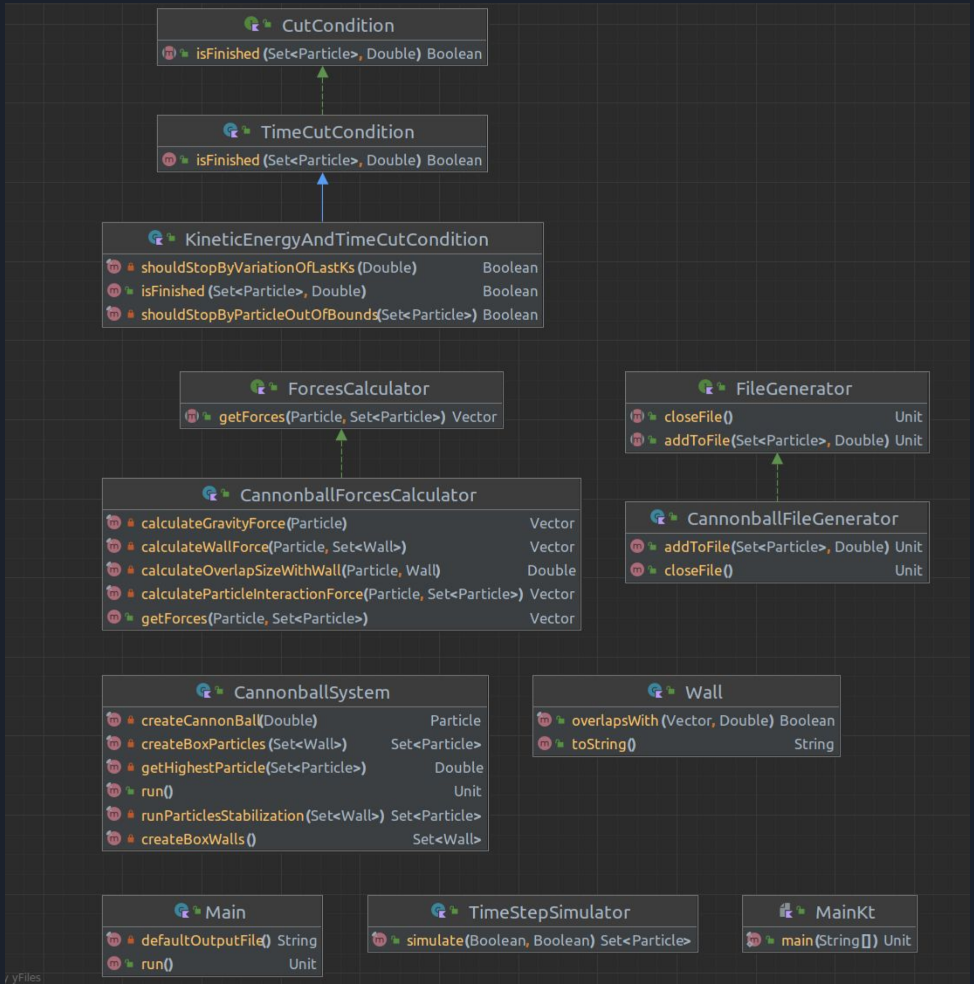
$$\overline{a_i} = \frac{\overline{F_i}}{m_i} \qquad (6)$$

Implementación

Implementación

- Se implementó el simulador en Kotlin, el post-procesamiento en Python y las animaciones en Ovito.
- Cada simulación genera un archivo de salida con formato .xyz con los datos de las partículas en cada instante de tiempo.
- La implementación en Kotlin se implementó separada en dos capas: engine y system.
- En engine se encuentra toda la implementación del motor de simulaciones dirigidas por el paso temporal.
- En system se encuentra toda la implementación para el sistema particular de este trabajo.
- Se utilizó el método de integración Beeman con la variación predictor-corrector que es mejor para sistemas con partículas que dependen de la velocidad.
- Para el post procesamiento se implementó un parser del archivo en .xyz para luego visualizarlo mediante matplotlib.

Implementación - UML



Implementación - Pseudocódigo

Ejecución de simulación

```
fun simulate() : Set<Particle> {
    fileGenerator.addToFile(particles, time)
    while (!cutCondition.isFinished(particles, time)) {
        val newParticles = particles.forEach(particle -> applyBeemanIntegrator(particle))
        time += timeDelta
        if (time >= timeToSave) {
            fileGenerator.addToFile(particles, time)
        }
    }
}
```

Cálculo de fuerzas para cada partícula

```
override fun getForces(particle: Particle, neighbours: Set<Particle>): Vector {
   val interactionForce = calculateParticleInteractionForce(particle , neighbours)
   val wallForce = calculateWallForce(particle , walls)
   val gravityForce = calculateGravityForce(particle)

   return interactionForce + wallForce + gravityForce
}
```

Simulaciones

Simulaciones

- Las partículas se generan de forma aleatoria con un diámetro distribuido uniformemente entre [$p_{LowDiam}$, $p_{UpperDiam}$]. Se simula la caída de las partículas sin la bala para permitir estabilizarse el sistema antes de agregar la caída de la bala.
- Una vez que el sistema se estabilizó, se agrega la bala al mismo con las partículas en la posición posterior a la estabilización. Todas las partículas comienzan con la posición y velocidad que tenían al momento de estabilizarse.
- La bala se dispara a una altura h definida con el punto P=(ladoDelCajon/2, ladoDelCajon/2, puntoDeLaParticulaConMayorAltura + 0.1). Es decir, centrada en **x** e y respecto del cajón y a una altura igual al punto en z de la partícula que se encuentra a una mayor altitud más 0.1 mts.
- La condición de corte de simulación tiene 2 parámetros: un tiempo final y/o variación de energía cinética haciendo un promedio en las últimos 1000 dT y comparando con la energía cinética actual.

Simulaciones - Parámetros

d_t = 5E-5

$n_{particles}$ =2000

mass= 0.085 kg

$p_{LowDiam}$= 0.015 m

$p_{UpperDiam}$= 0.03 m

p_{Kn}= 2E6 N/m

p_{Kt} = 2*p_{Kn}

p_{Gamma} = 100

$wall_{Kn}$= p_{Kn}

$wall_{Kt}$ = 2 * $wall_{Kn}$

d_{t2} = 0.001

$ball_{mass}$ = 17.5 kg

$ball_{Kn}$= p_{Kn}

$ball_{Kt}$= 2*$ball_{Kn}$

$ball_{Gamma}$ = 50

$ball_{Angle}$ = 90 grados

$ball_{Velocity}$ = 20 m/s

$ball_{diameter}$ = 0.175 m

gravity = 9.81 m/s^2

box_{width} = 0.4 m

Simulaciones - Observable

- Se definen como observables las siguientes métricas:
  - Energía cinética del sistema hasta la estabilización
  - Velocidad de la bala en el tiempo hasta la estabilización
  - Tiempo de estabilización vs gamma
  - Tiempo de estabilización vs $n_{Particle}$
  - Tiempo de estabilización vs ángulo de caída de la bala
  - Tiempo de estabilización vs diámetros de las partículas
- Para cada simulación se realizaron 5 repeticiones (debido al largo tiempo de ejecución de cada una, aproximadamente entre 2 a 3 hs cada 5 repeticiones)
- Para cada tiempo de estabilización vs parámetro se grafican el promedio con su desvío en barras.
- Se define energía cinética de la partícula p_j cómo:

$$K = \frac{m_{p_j} \cdot v_{p_j}^2}{2}$$

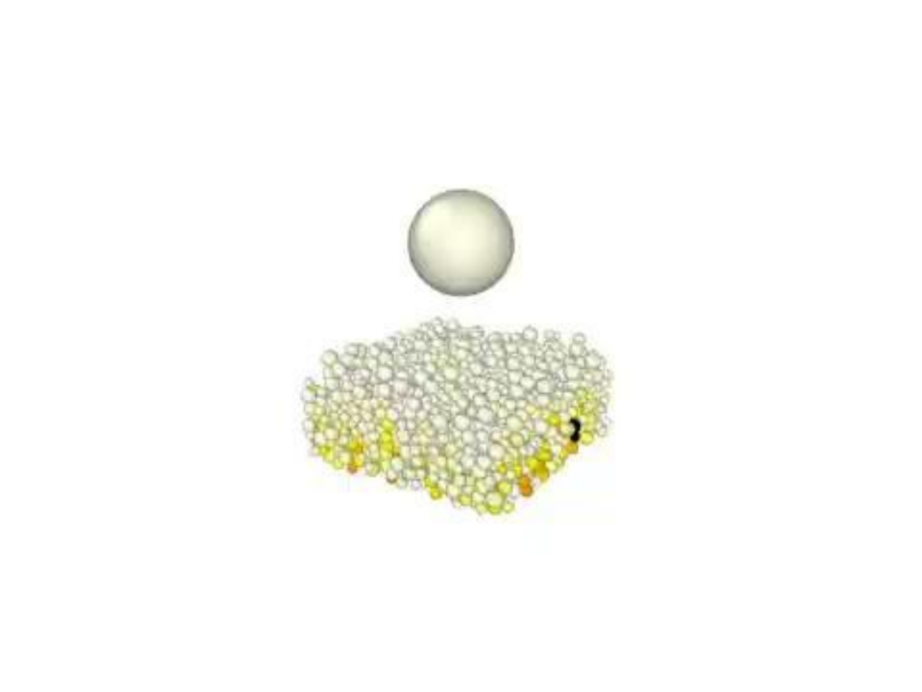
- La energía cinética del sistema es la sumatoria de energías.
- Para todas las simulaciones el color que tienen las partículas es la presión que se ejerce sobre ella.

Resultados

Resultados - Animaciones

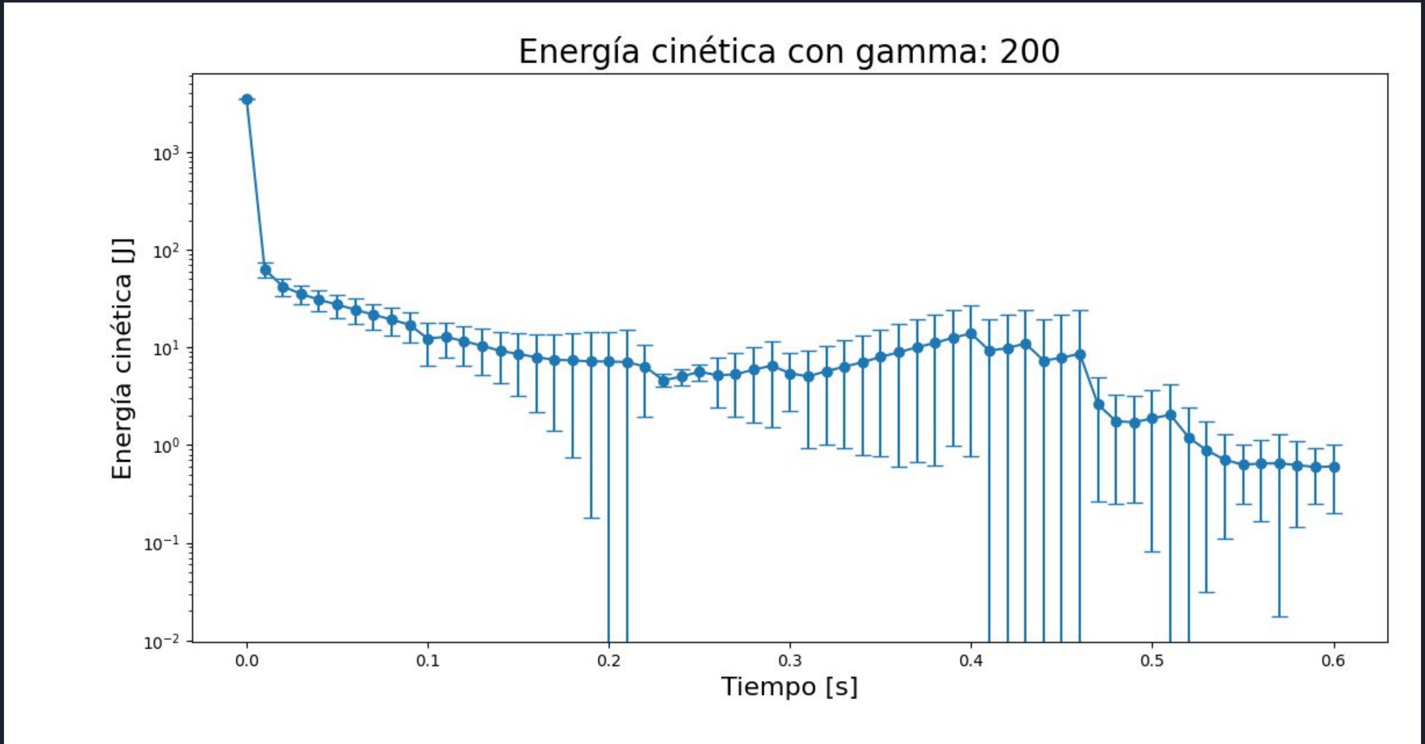
Variación de p_{gamma} = [100, 125, 150, 175, 200, 225, 250, 275, 300]

p_{gamma} = 200



Resultados - Observable

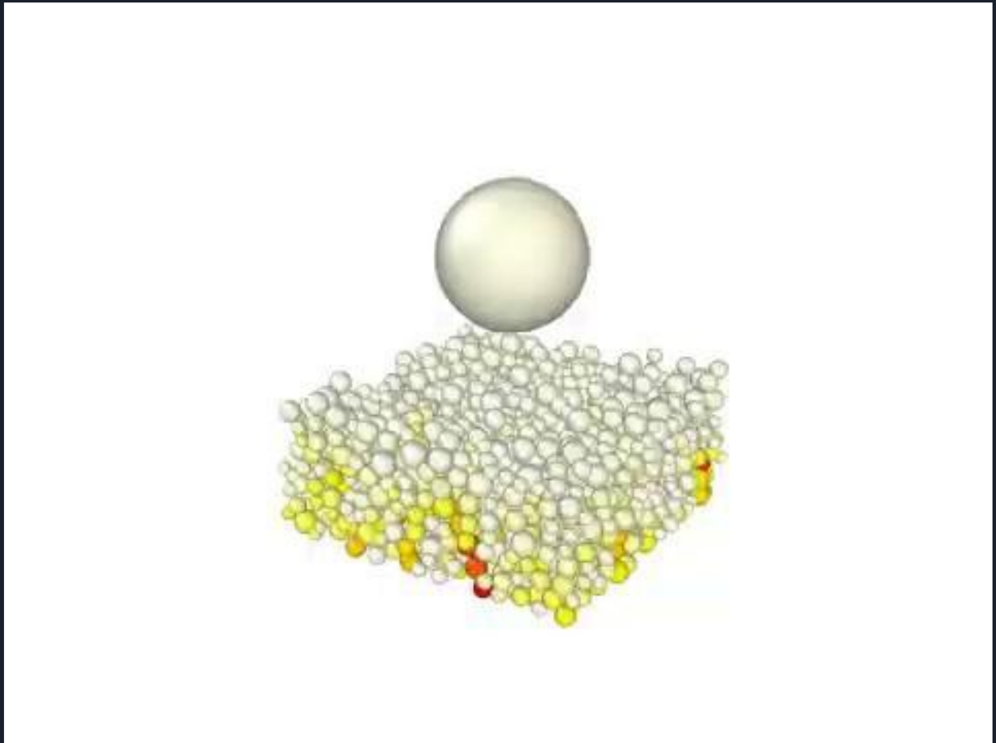
Variación de p_{gamma} = [100, 125, 150, 175, 200, 225, 250, 275, 300]



Resultados - Animaciones

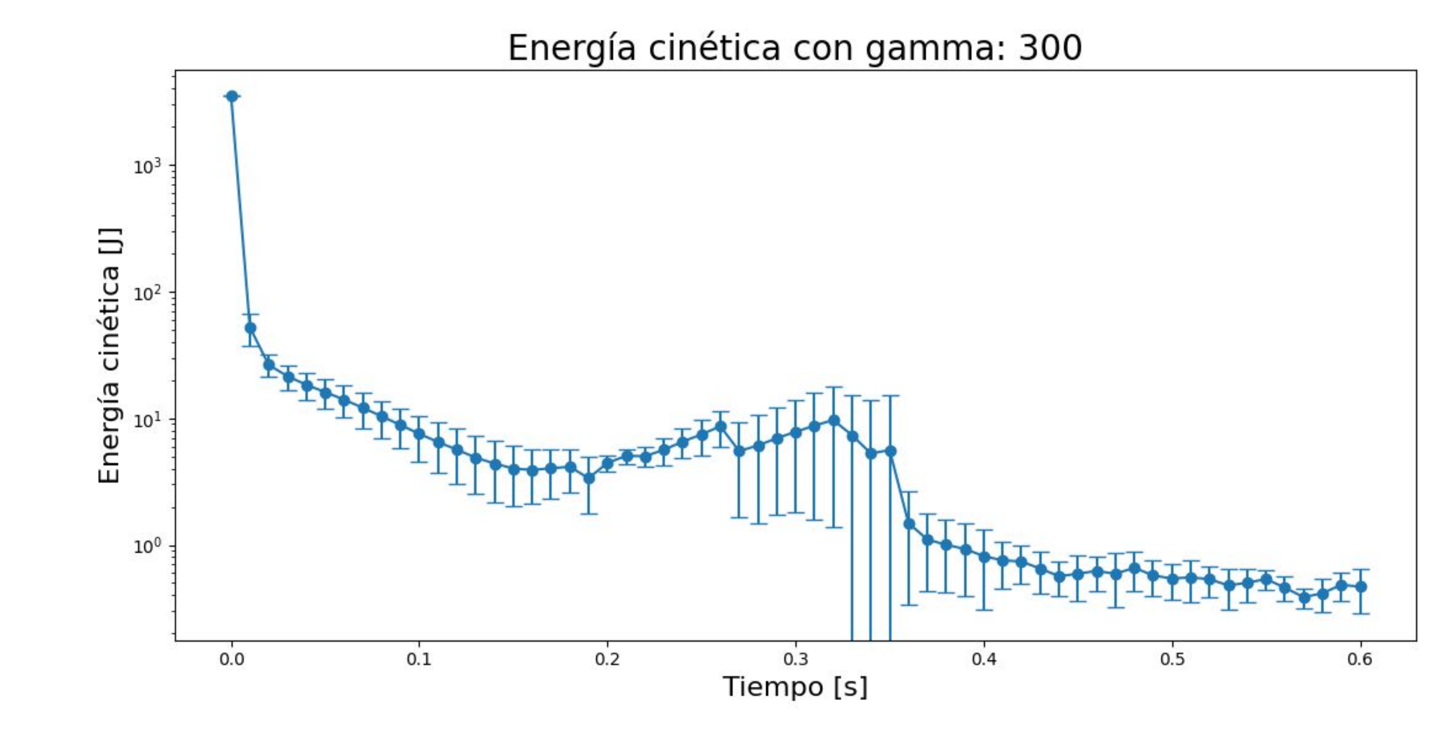
Variación de p_{gamma} = [100, 125, 150, 175, 200, 225, 250, 275, 300]

p_{gamma} = 300



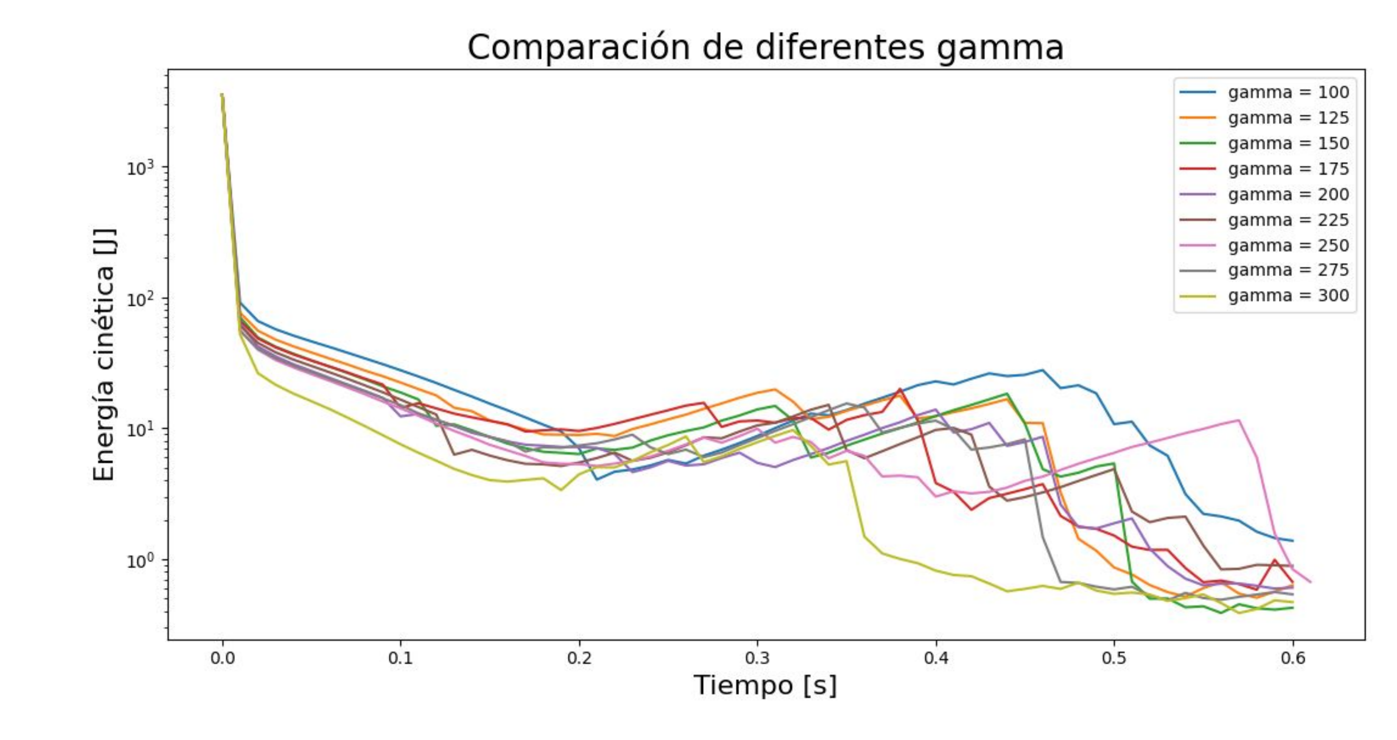
Resultados - Observable

Variación de p_{gamma} = [100, 125, 150, 175, 200, 225, 250, 275, 300]



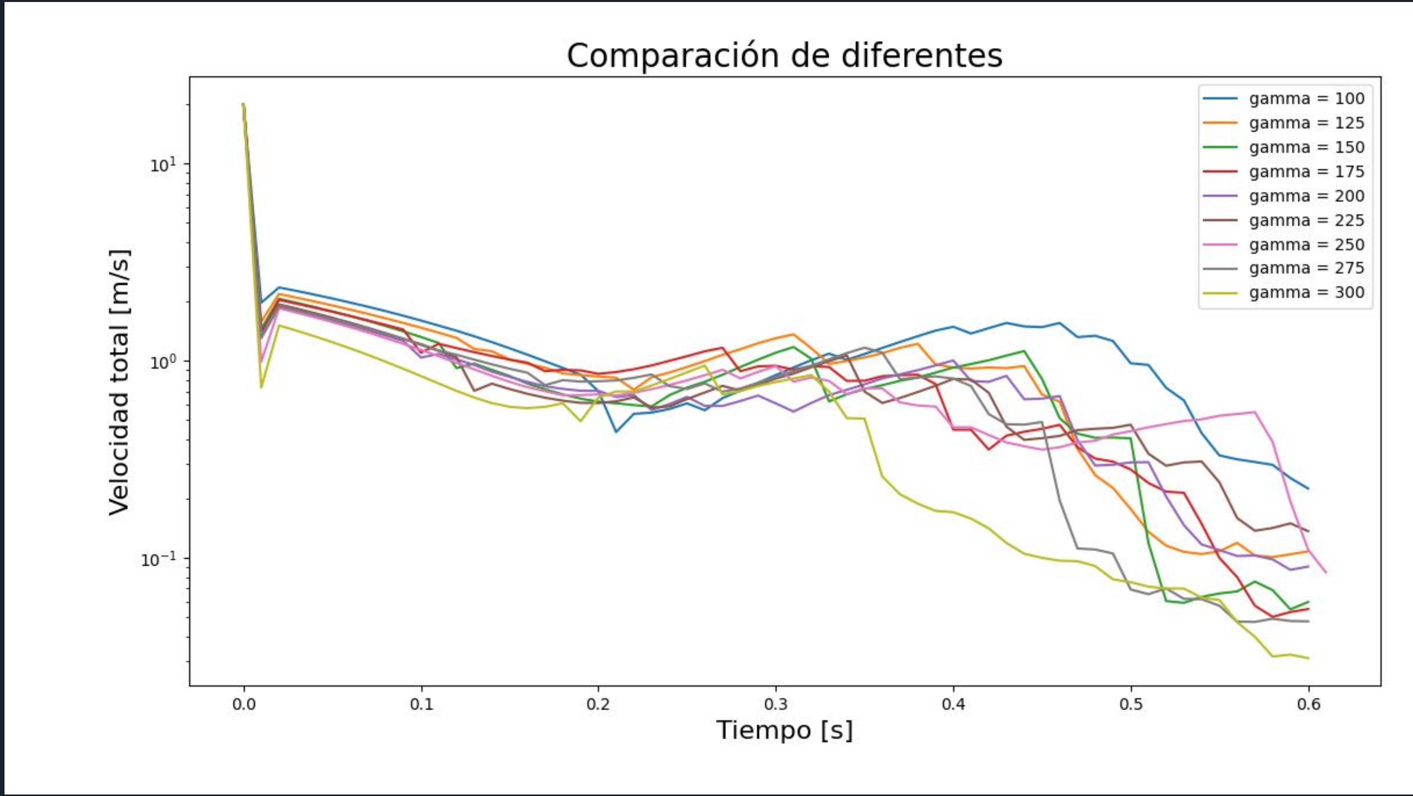
Resultados - Observable

Variación de p_{gamma} = [100, 125, 150, 175, 200, 225, 250, 275, 300]



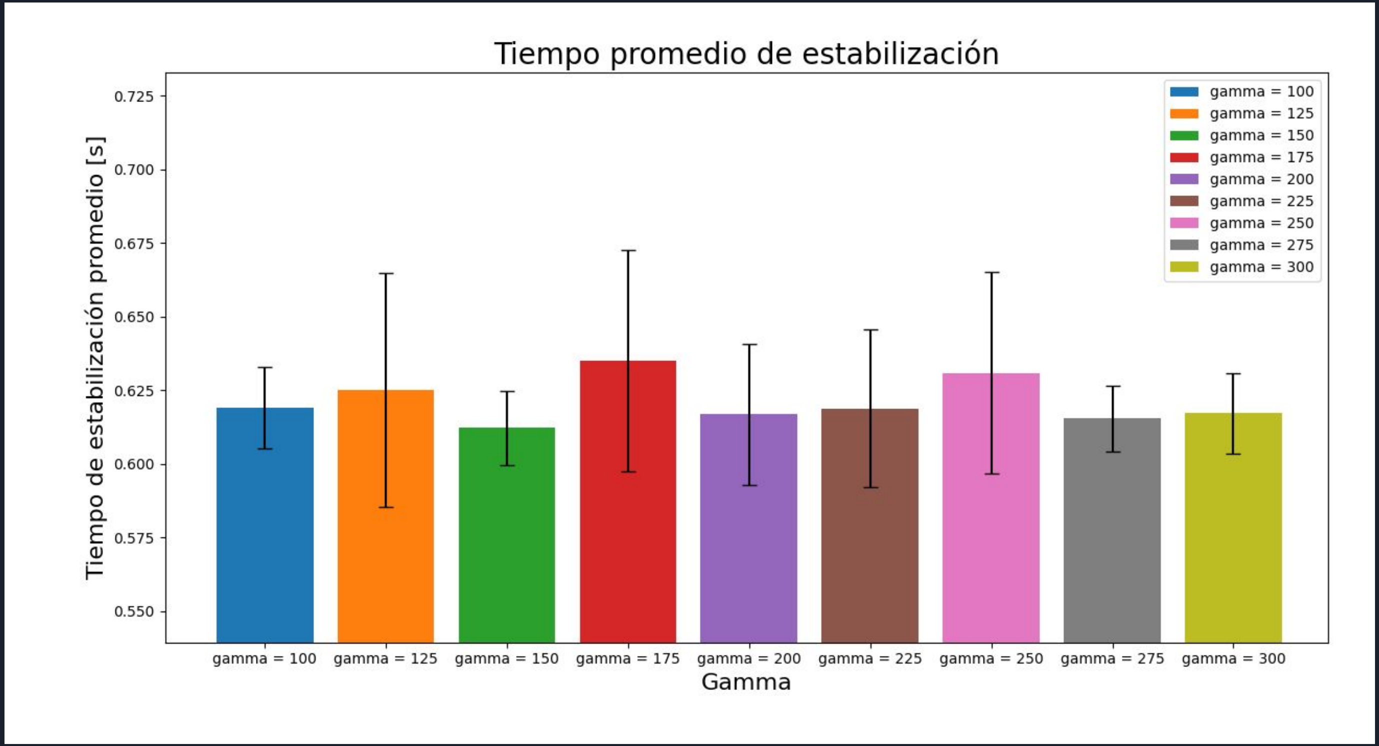
Resultados - Observable

Variación de p_{gamma} = [100, 125, 150, 175, 200, 225, 250, 275, 300]



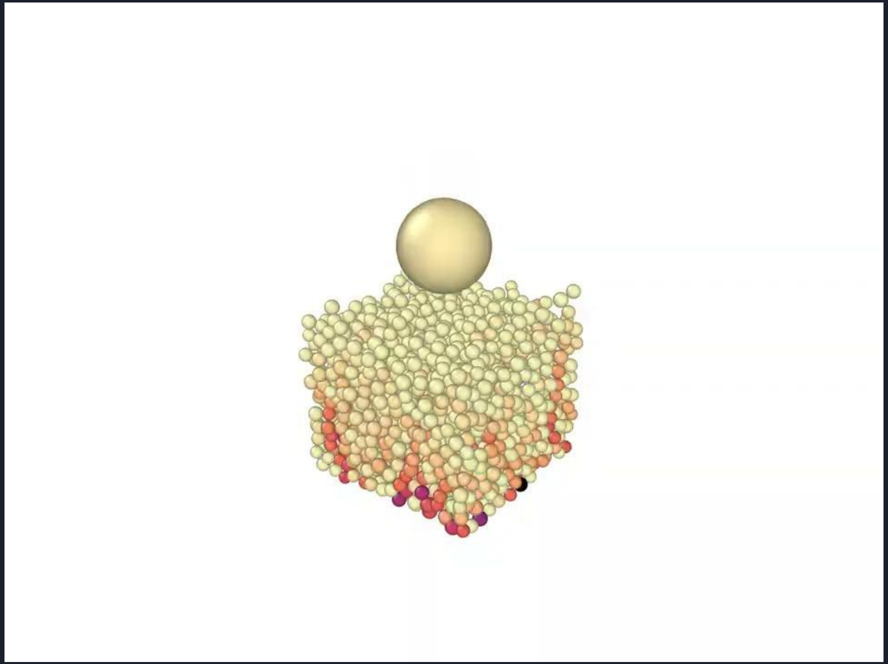
Resultados - Observable

Variación de p_{gamma} = [100, 125, 150, 175, 200, 225, 250, 275, 300]



Resultados - Animaciones

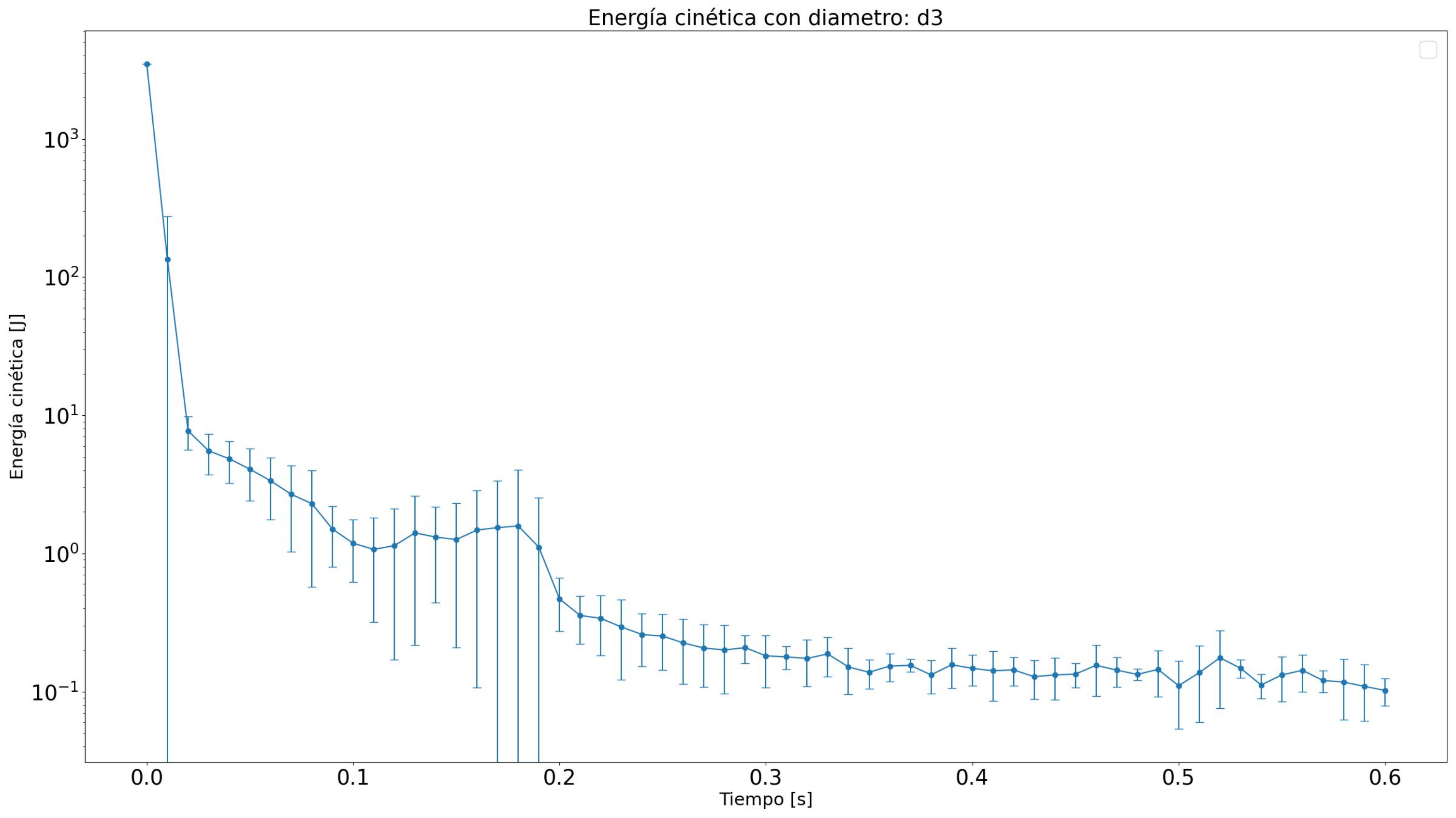
Variación de diámetros, d2=[0.02, 0.025], d3=[0.025, 0.03] en mts



d3 = [0.025, 0.03] metros

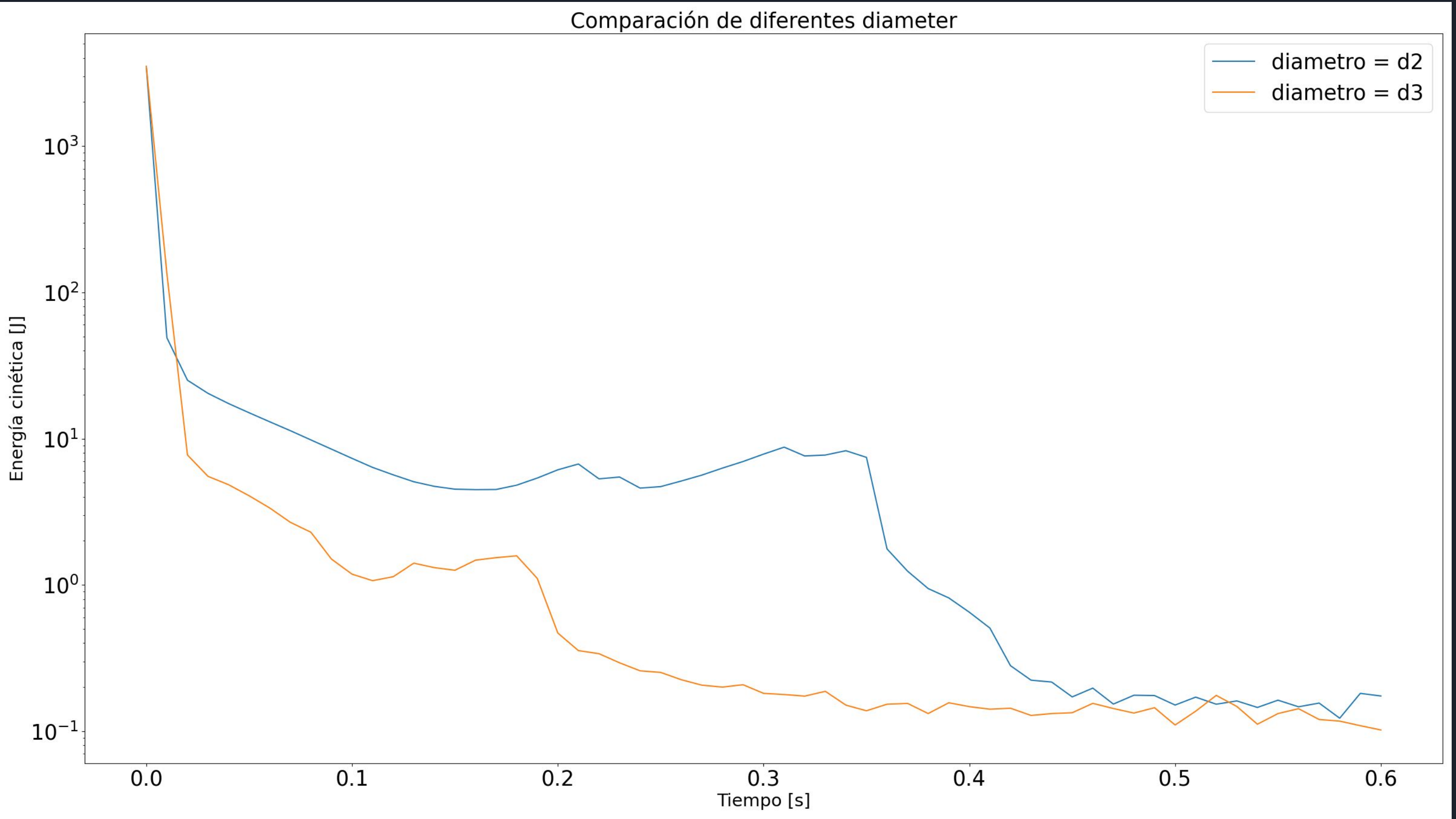
Resultados - Observable

Variación de diámetros, d2=[0.02, 0.025], d3=[0.025, 0.03] en mts



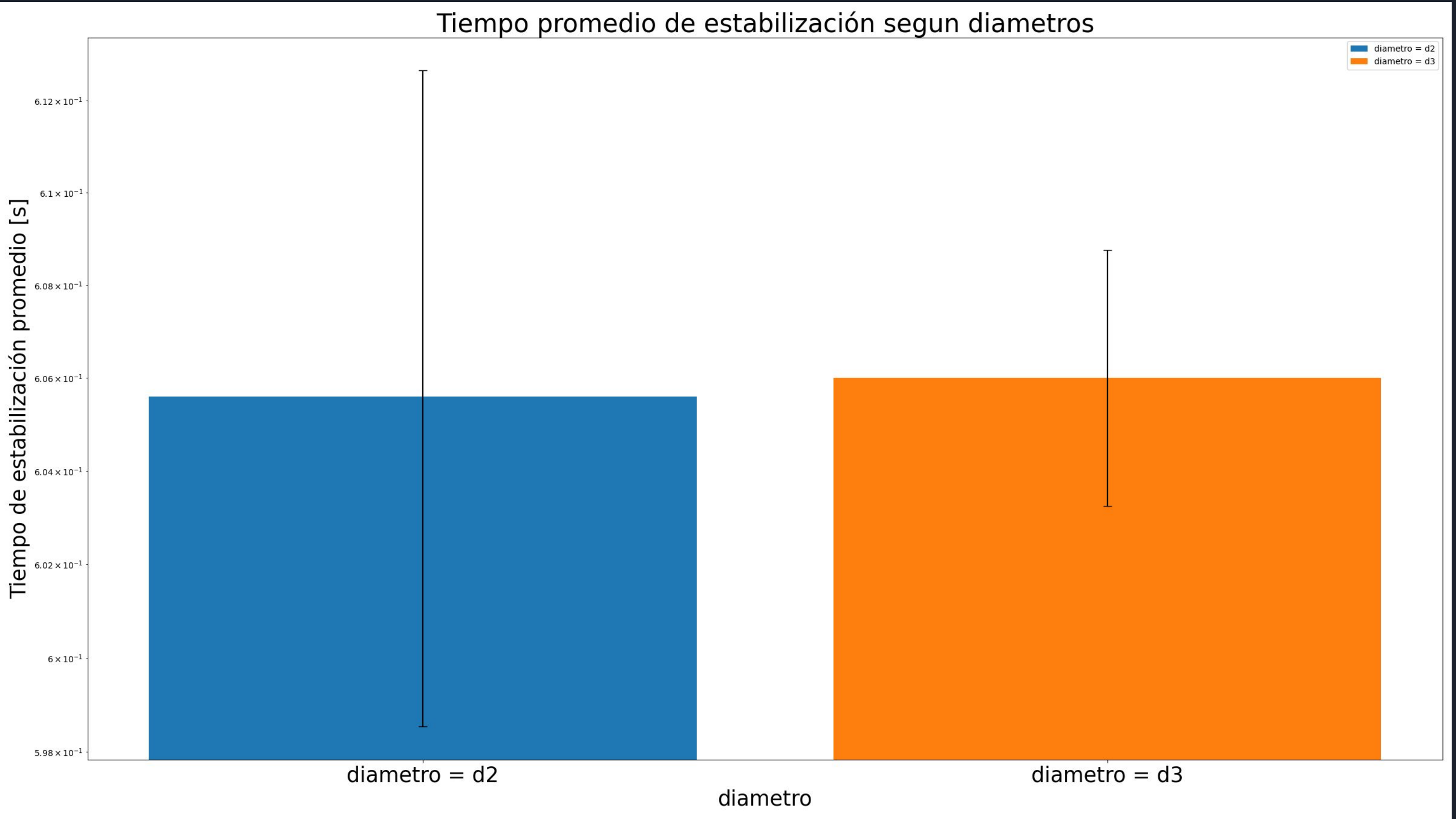
Resultados - Observable

Variación de diámetros, d2=[0.02, 0.025], d3=[0.025, 0.03] en mts



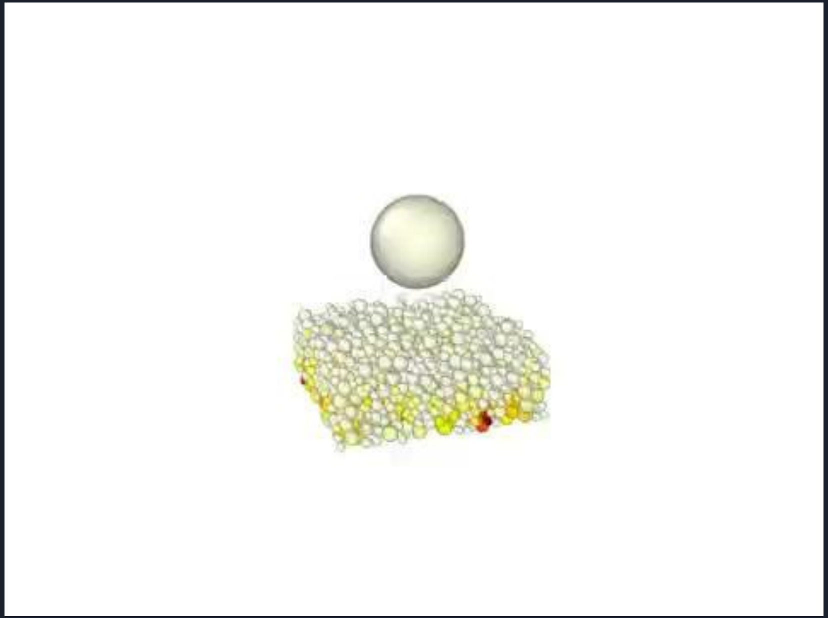
Resultados - Observable

Variación de diámetros, d2=[0.02, 0.025], d3=[0.025, 0.03] en mts



Resultados - Animaciones

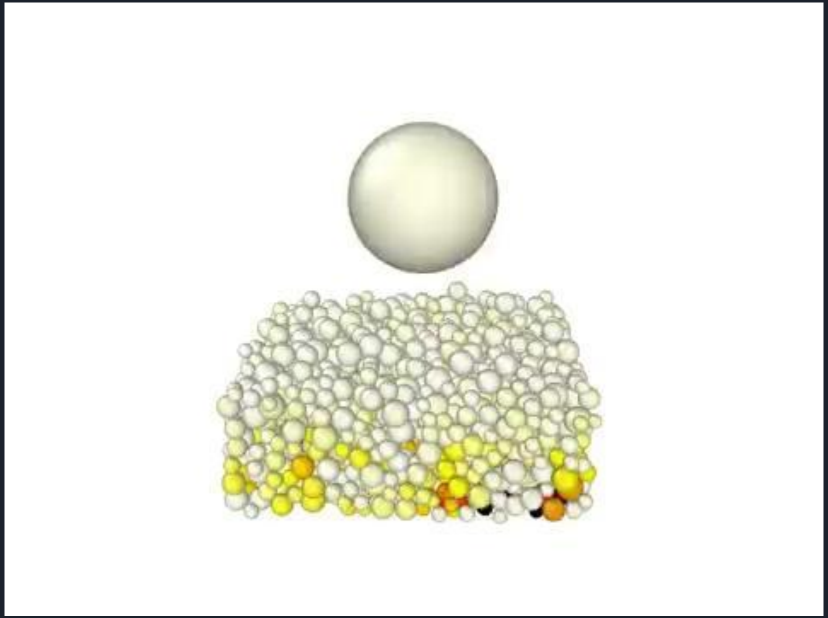
Variación de $ball_{angle}$ = [75, 80, 85] en grados



$ball_{angle}$ = 80 grados

Resultados - Animaciones

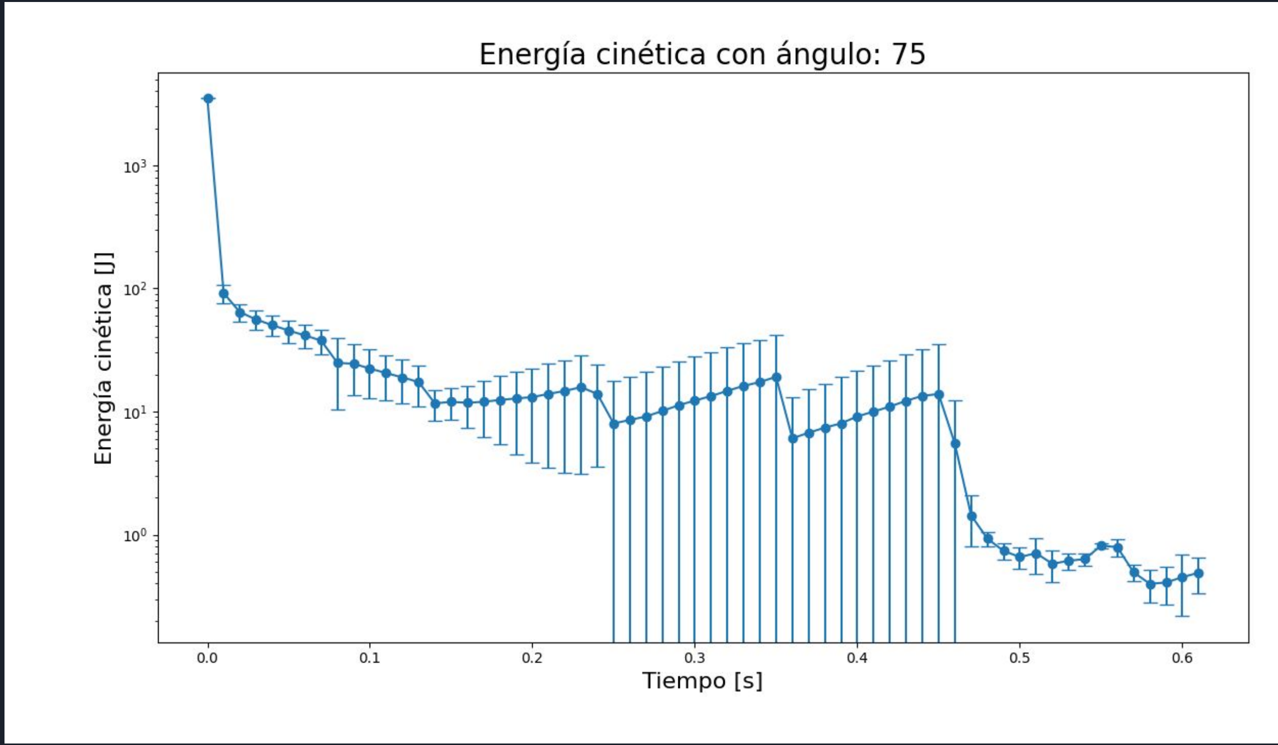
Variación de $ball_{angle}$ = [75, 80, 85] en grados



$ball_{angle}$ = 75 grados

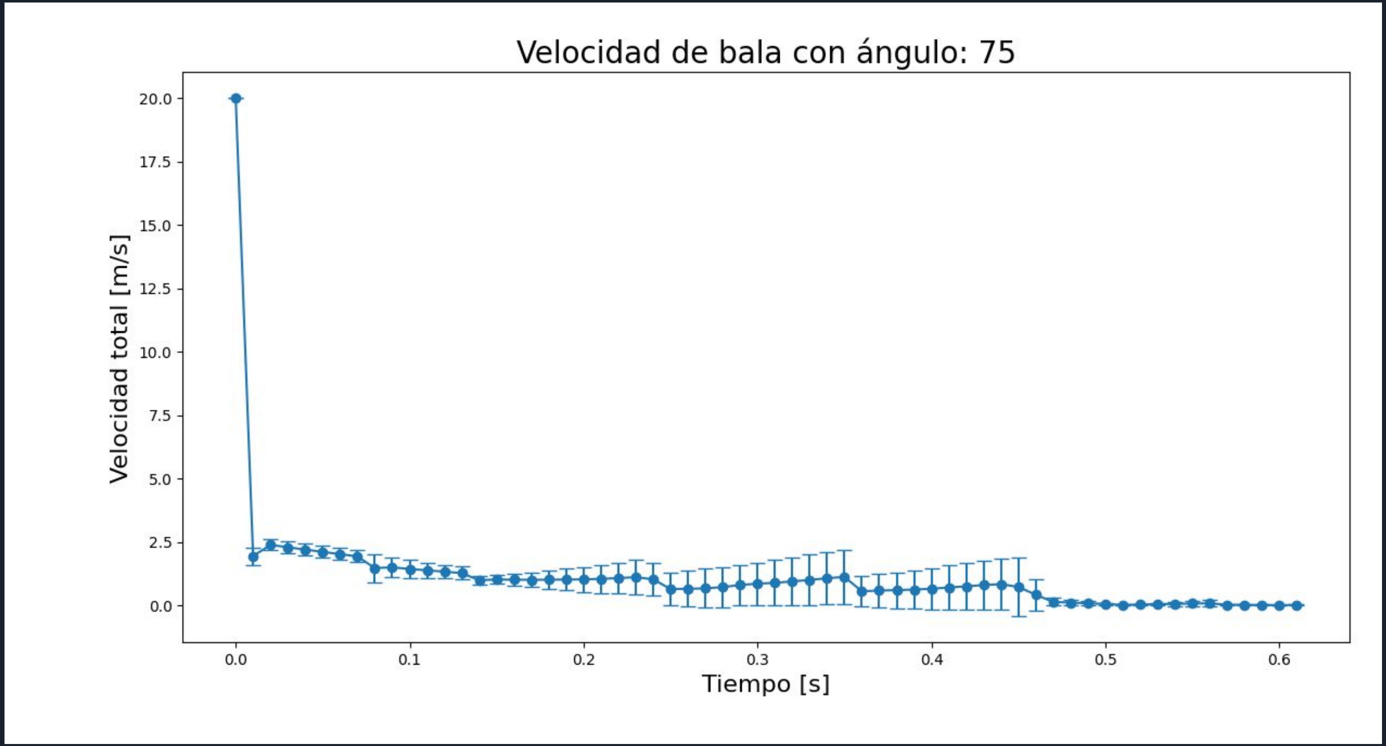
Resultados - Observable

Variación de $ball_{angle}$ = 75 grados



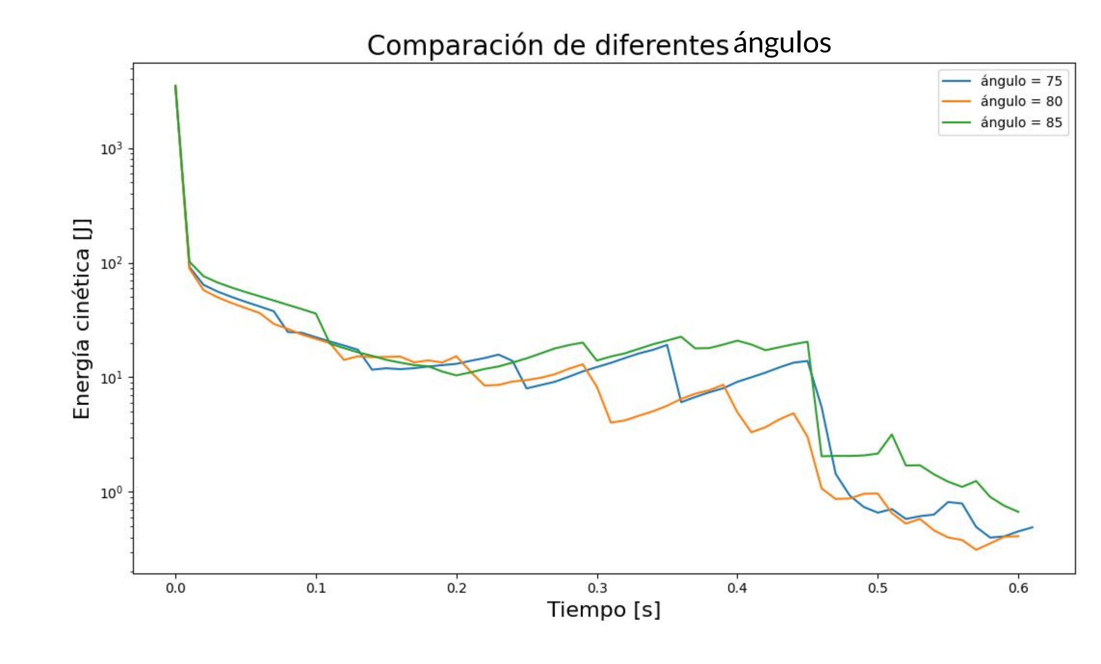
Resultados - Observable

Variación de $ball_{angle}$ = 75 grados



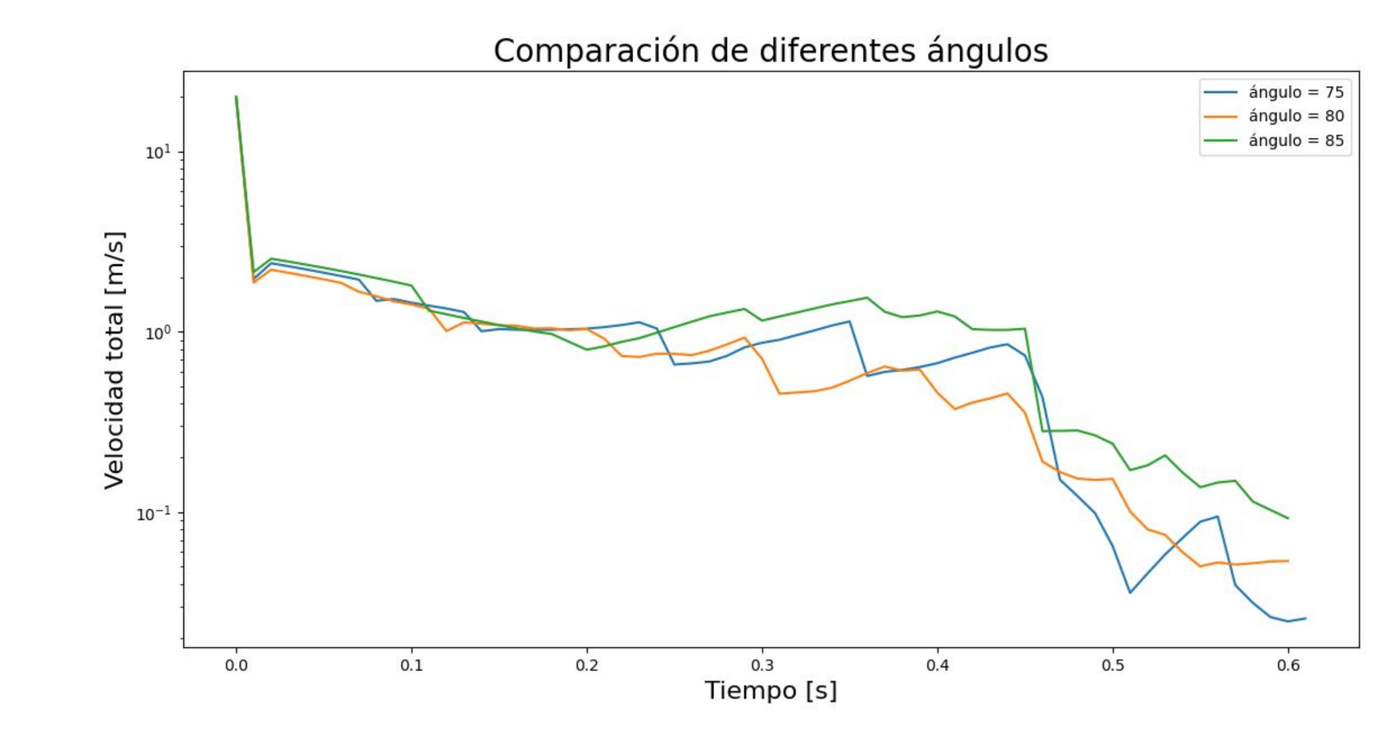
Resultados - Observables

Variación de $ball_{angle}$ = [75, 80, 85] en grados



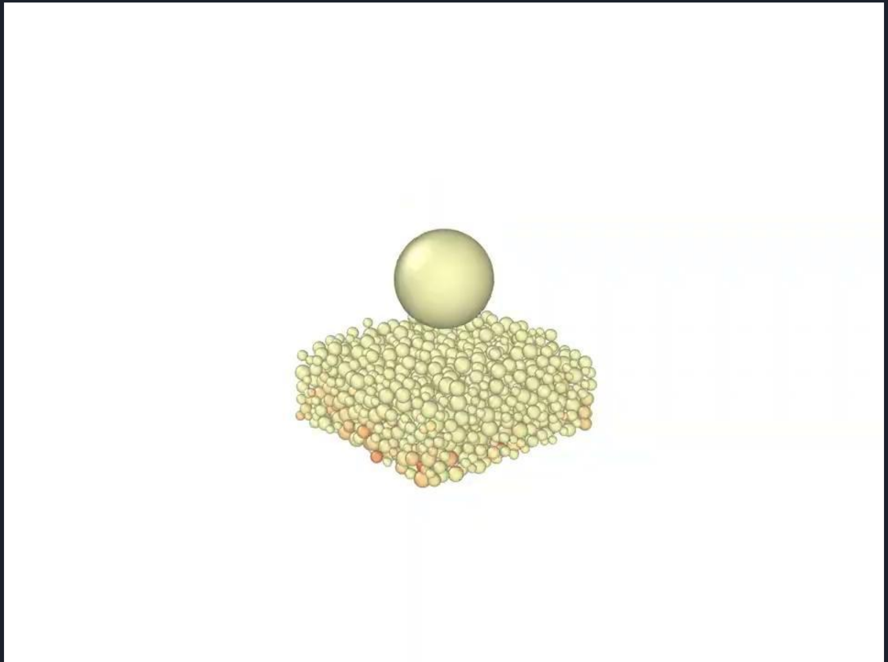
Resultados - Observables

Variación de $ball_{angle}$ = [75, 80, 85] en grados



Resultados - Animaciones

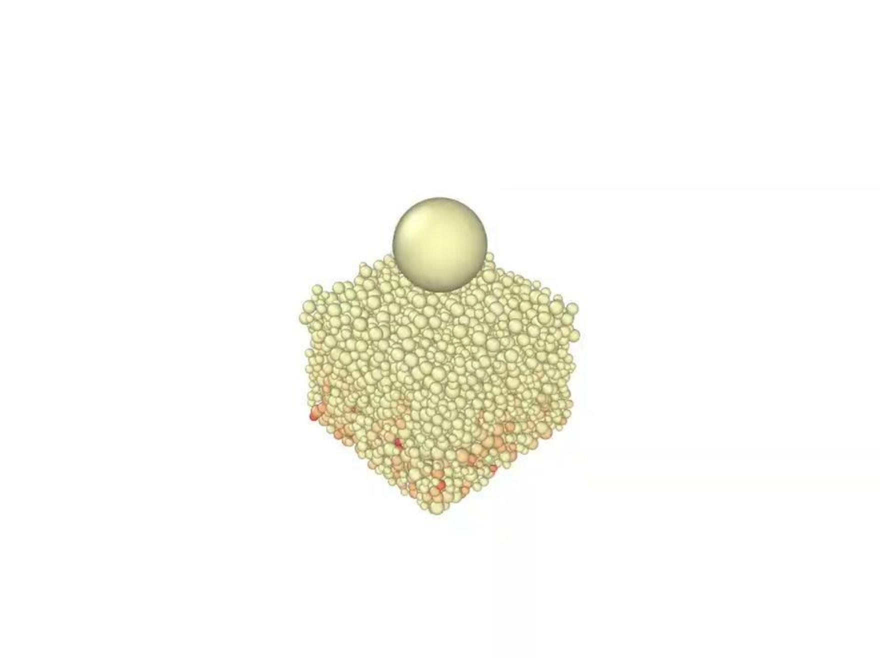
Variación de $n_{particles}$ = [2000, 3000, 4500]



$n_{particles}$=2000

Resultados - Animaciones

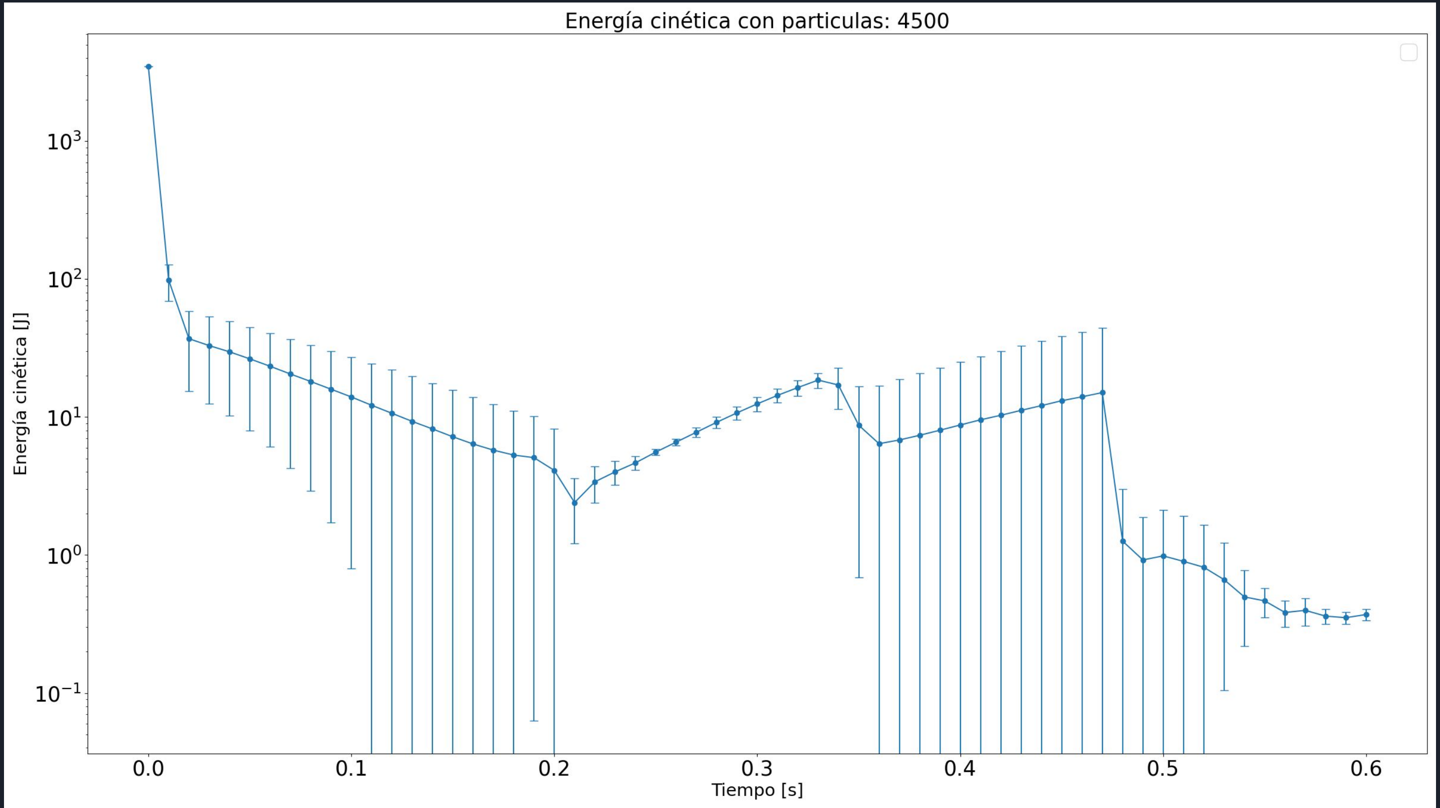
Variación de $n_{particles}$ = [2000, 3000, 4500]



$n_{particles}$=4500

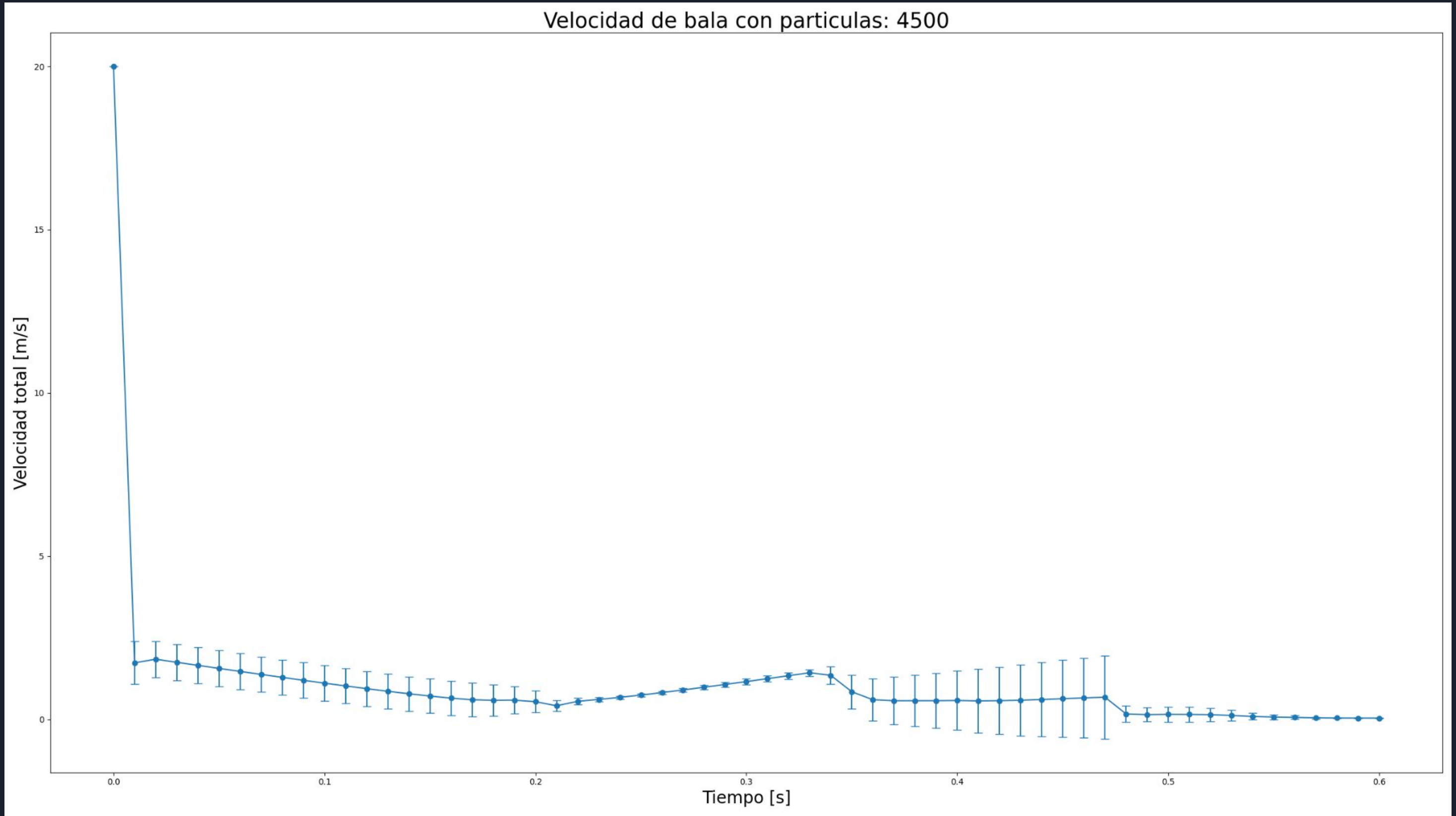
Resultados - Observable

Variación de $n_{particles}$ = [2000, 3000, 4500]



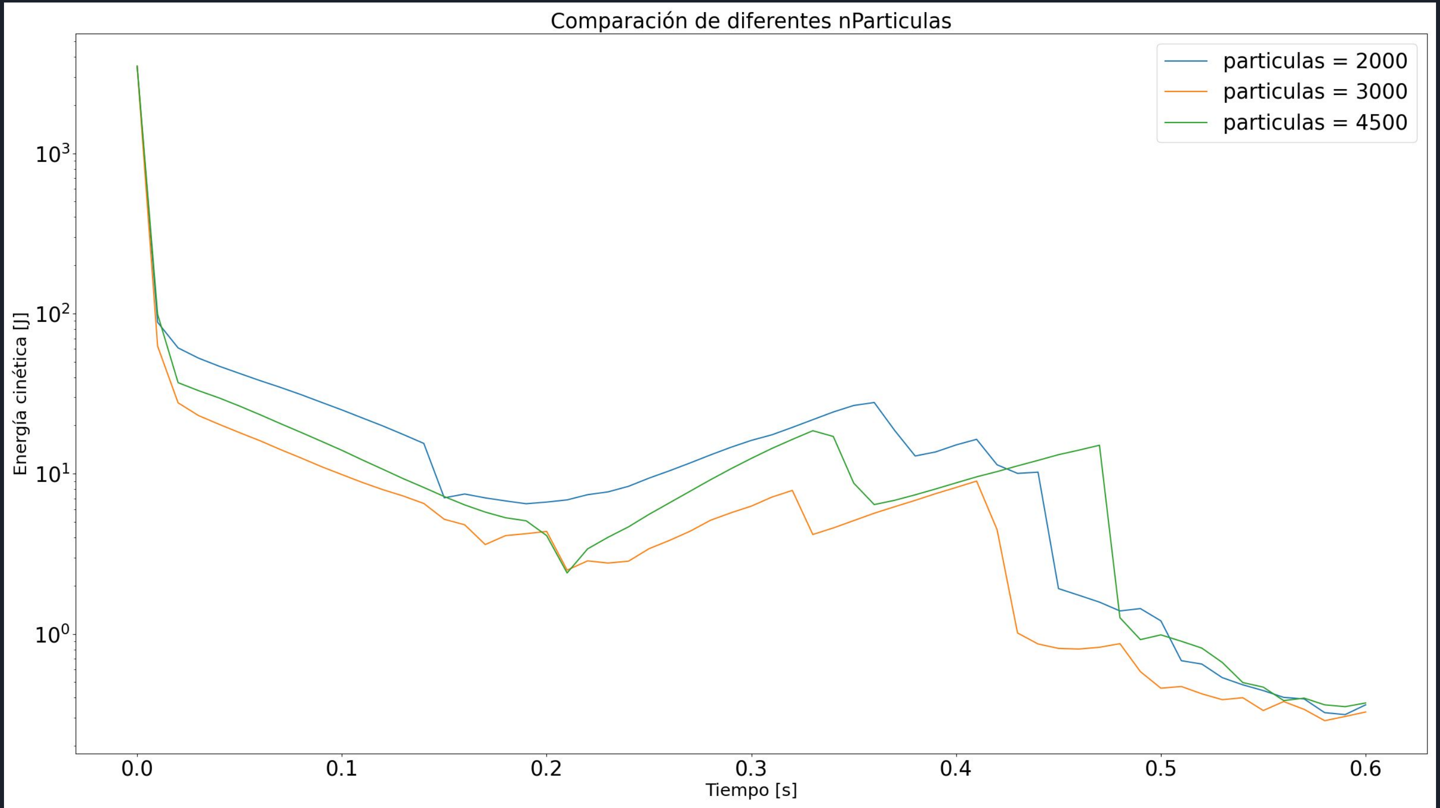
Resultados - Observable

Variación de $n_{particles}$ = [2000, 3000, 4500]



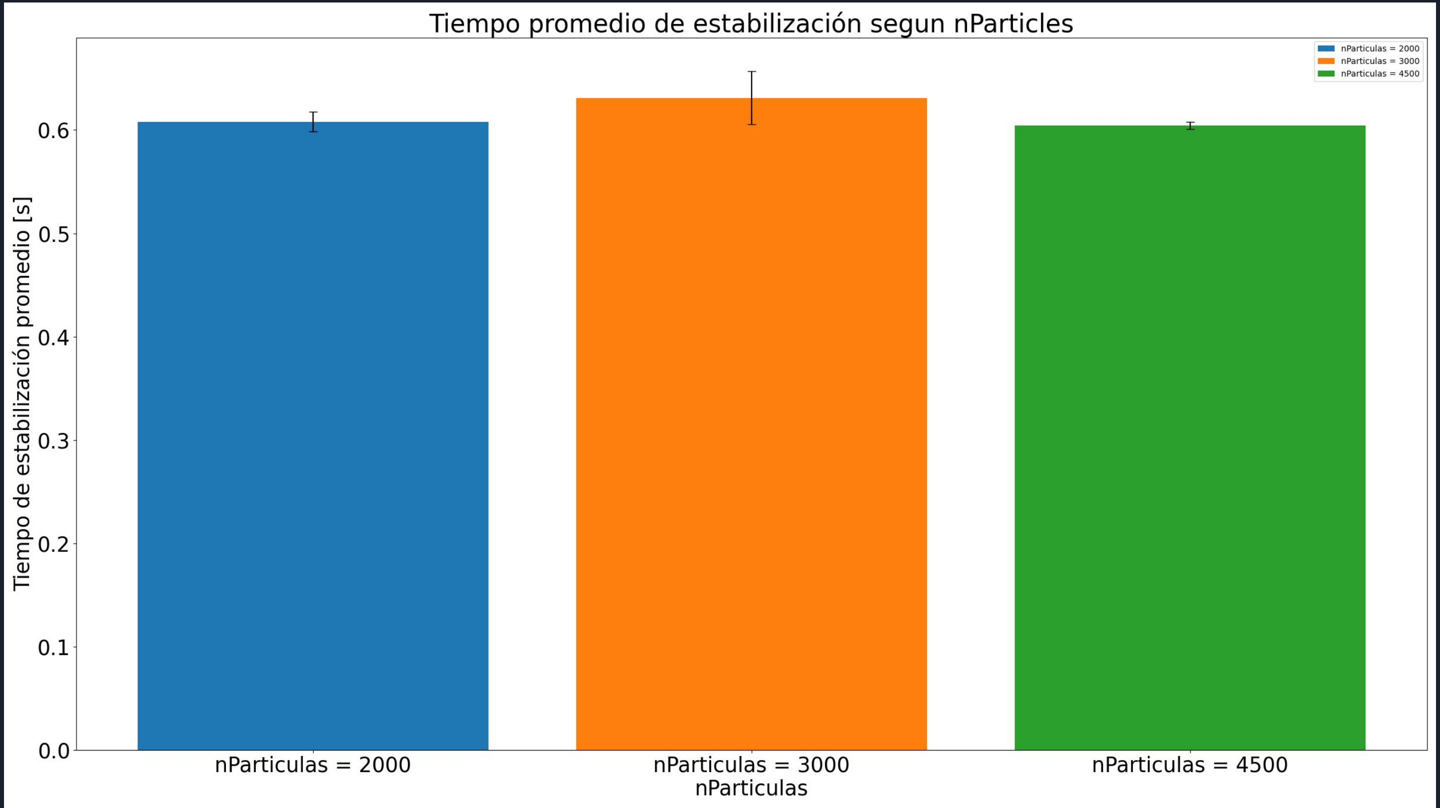
Resultados - Observable

Variación de $n_{particles}$ = [2000, 3000, 4500]



Resultados - Observable

Variación de $n_{particles}$ = [2000, 3000, 4500]



Conclusiones

Conclusiones

- A medida que se aumenta el valor de gamma disminuye la energía cinética del sistema debido a una mayor disipación de energía y también se reduce la velocidad de la bala de cañón.
- Un mayor número de partículas en el sistema, así como también un mayor diámetro de las mismas, disminuye la energía cinética total debido a una mayor probabilidad de colisiones entre las partículas, lo que resulta en transferencias de energía, cambios en la velocidad y dirección de las mismas.
- Cuando el ángulo de caída de una bala es menos perpendicular, se observa un rebote menos enérgico debido a una mayor componente de movimiento en x y en y, lo que reduce la transferencia de energía cinética durante la colisión.

¡Muchas gracias!