

ITBA

Simulación de Sistemas

Trabajo práctico final

Flujos Granulares 3D: Caída de una bala de cañón

Alumnos:

Tomás Dallas

56436

Tomás Dorado

56594

Grupo 6

18 de Julio de 2023

Índice

1 Introducción	2
2 Modelo	2
2.1 Flujos granulares	2
2.2 Modelado en 3 dimensiones	2
2.3 Fuerzas que intervienen en los choques	3
2.4 Fuerzas totales	4
2.5 Simulación dirigida por el paso temporal	4
2.6 Método de Integración	4
3 Implementación	5
3.1 Simulador	5
3.2 Post procesamiento	6
3.3 Animaciones	6
4 Simulaciones	6
5 Resultados y Conclusiones	7
5.1 Variación de gamma	7
5.2 Variación de diámetros	11
5.3 Variación de ángulo de caída de la bala	12
5.3 Variación de cantidad de partículas	14
5.4 Conclusiones	17
6 Referencias	17

1 Introducción

El presente informe se centra en el estudio del comportamiento de flujos granulares tridimensionales (3D), específicamente en la caída de una bala de cañón sobre un lecho granular. El objetivo principal de este trabajo es emplear simulaciones parametrizables para observar y analizar las ventajas y desventajas del modelo propuesto, así como de las técnicas utilizadas.

El informe está estructurado en cuatro secciones principales. En primer lugar, se presenta el modelo y los fundamentos teóricos que respaldan este trabajo. A continuación, se describen los detalles de implementación que se llevaron a cabo para desarrollar el sistema encargado de simular el modelo propuesto. Posteriormente, se presentan los resultados obtenidos a partir de las pruebas realizadas para validar y caracterizar el modelo. Por último, se exponen las conclusiones derivadas del trabajo realizado.

2 Modelo

El sistema implementado consiste en un cajón lleno de partículas de diferentes tamaños, donde se simula la interacción entre la bala de cañón y las partículas durante su descenso. A continuación, se presentarán las explicaciones detalladas sobre las implementaciones realizadas y las consideraciones relevantes para el desarrollo del sistema de simulación.

2.1 Flujos granulares

Un flujo granular se refiere a un sistema compuesto por numerosas partículas macroscópicas, como granos de arena o polvo, que coexisten dentro de un entorno físico denominado contenedor. Estas partículas macroscópicas interactúan entre sí y con el contenedor a través de fuerzas generadas por colisiones. Estas fuerzas de interacción siempre involucran a dos entidades, ya sea dos partículas o una partícula y el contenedor, lo que implica que las colisiones ocurren en pares.

En el contexto de este trabajo, al modelar la caída de una bala de cañón en un lecho granular, es crucial tener en cuenta el efecto de la gravedad. Esta fuerza actúa de forma independiente a las fuerzas de interacción mencionadas anteriormente y se mantiene constante para todas las partículas presentes en el sistema.

2.2 Modelado en 3 dimensiones

En primer lugar, se establece la convención adoptada para la orientación de los ejes en el modelado tridimensional:

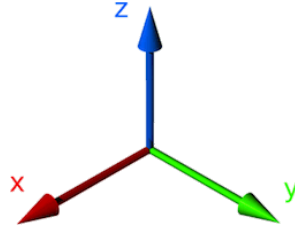


Fig. 1: Ejes en 3 dimensiones

Como se muestra en la Fig. 1, el efecto de la gravedad se refleja en la componente Z de las partículas. Las posiciones, velocidades, aceleraciones y fuerzas tienen sus valores en X, Y, Z.

2.3 Fuerzas que intervienen en los choques

En un escenario ideal las fuerzas que determinan estas interacciones son fuerzas de contacto normales (deformación viscoelástica) y tangenciales (rozamiento). Bajo estas fuerzas, estas interacciones son altamente disipativas, lo que significa que el sistema llega al reposo si no recibe energía de otra fuerza (por ejemplo, de la fuerza gravitatoria).

En el caso de un choque entre una partícula i con una partícula j las fuerzas se calculan de la siguiente manera:

$$\overline{Fn_{ij}} = -k_n \xi - \gamma \hat{v}_{ij}^n \quad (1)$$

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

Donde Fn_{ij} es la fuerza normal entre las partículas i y j , k_n es el coeficiente de rigidez normal, ξ es el solapamiento entre las partículas, γ es el coeficiente de amortiguamiento y \hat{v}_{ij}^n (3) es la velocidad relativa en la dirección normal. La fuerza tangencial Ft_{ij} se calcula de manera similar, utilizando el coeficiente de rigidez tangencial k_t y el solapamiento ξ .

$$\hat{v}_{ij}^n = \overline{v_{ij}} \cdot \hat{r}_{ij} \quad (3)$$

Las fuerzas totales en la colisión quedan entonces:

$$\overline{F_{ij}} = \overline{Fn_{ij}} \cdot \hat{r}_{ij} + \overline{Ft_{ij}} \cdot \hat{v}_{ij} \quad (4)$$

siendo: $\hat{v}_{ij}^t = \overline{v_{ij}} - \hat{v}_{ij}^n \cdot \hat{r}_{ij}$ el cálculo del versor $\hat{v}_{ij} = \frac{\hat{v}_{ij}^t}{|\hat{v}_{ij}^t|}$, la velocidad relativa $\overline{v_{ij}} = \overline{v_i} - \overline{v_j}$, el

versor normal $\hat{r}_{ij} = \frac{\overline{r_{ij}}}{|\overline{r_{ij}}|}$ y por último $\overline{r_{ij}} = \overline{r_i} - \overline{r_j}$

2.4 Fuerzas totales

Las fuerzas totales se calculan sumando las fuerzas de la gravedad y las fuerzas de choques. Los choques con las paredes se calculan de la misma manera que con otras partículas tomando a las paredes como otra partícula más..

$$\overline{F}_i = \overline{Fg}_i + \sum_j \overline{F}_{ij} \quad (5)$$

siendo $\overline{Fg}_i = m_i \overline{g}$ y tomando a la gravedad de la tierra como g

Por último se puede calcular la aceleración de la partícula se calcula a partir de la segunda ley de Newton

$$\overline{a}_i = \frac{\overline{F}_i}{m_i} \quad (6)$$

2.5 Simulación dirigida por el paso temporal

En el contexto de este modelo, se considera apropiado emplear una simulación basada en avance temporal discreto, ya que esta estrategia es adecuada cuando las partículas se encuentran en contacto la mayor parte del tiempo y cuando el tiempo de vuelo es considerablemente menor que la duración de las colisiones.

Bajo esta metodología, el tiempo avanza en incrementos discretos Δt , conocidos como "paso temporal". En cada paso temporal, se calcula el nuevo estado del sistema en función del tiempo y del estado en pasos anteriores. Esto permite simular la evolución dinámica de las partículas y su interacción a lo largo del tiempo de manera precisa y eficiente.

2.6 Método de Integración

El cálculo del estado en cada paso temporal se realiza mediante la aplicación de un método de integración específico. Este método permite llevar a cabo la integración numérica de las ecuaciones del movimiento. En el presente trabajo, se empleó el método de Beeman [1] como método de integración y se utilizó la variante predictor-corrector que sirve para tener mejor precisión en fuerzas que dependen de la velocidad como es este caso.

En la ecuación (6) podemos ver cómo se calcula la posición y en la ecuación (7) podemos ver cómo se calcula la predicción de velocidad. Estos son usados para hacer el cálculo de la corrección de velocidad que se puede ver en la ecuación (8).

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{2}{3}a(t)\Delta t^2 - \frac{1}{6}a(t - \Delta t)\Delta t^2 \quad (6)$$

$$v(t + \Delta t)_{(predicted)} = v(t) + \frac{2}{3}a(t)\Delta t - \frac{1}{2}a(t - \Delta t)\Delta t \quad (7)$$

$$v(t + \Delta t)_{(corrected)} = v(t) + \frac{1}{3}a(t + \Delta t)\Delta t + \frac{5}{6}a(t)\Delta t - \frac{1}{6}a(t - \Delta t)\Delta t \quad (8)$$

3 Implementación

Se implementó el simulador en Kotlin, el postprocesamiento en Python y las animaciones en Ovito. El código fuente e instrucciones de uso se encuentra en el repositorio: <https://github.com/tdorado/ss-final>

3.1 Simulador

Para la simulación se generó una caja cuadrada de lados 0.4 mts y altura infinita para simplificar el modelo. Se utiliza un $\Delta t = 1E-5$ segundos, y una escritura a archivo cada 0.001 segundos.

En el instante $t_0=0$ se generan las partículas. Cada partícula es generada en una posición aleatoria, con velocidad 0 m/s y un diámetro en metros que se distribuye uniformemente en un intervalo d . Luego de generadas las partículas, se las libera del reposo hasta que se estabilizan.

Luego de estabilizadas, se agrega la bala al sistema y se simula el disparo. En cada paso temporal se chequea si se ha alcanzado la condición de corte. Esta condición de corte se compone por las siguientes componentes:

- Corte por energía cinética: Una vez que la energía cinética del sistema llegó a un límite bajo de $1E-3$ J y corrió un mínimo de tiempo de 0.6 segundos se corta la simulación.
- Corte por estancamiento: Si el sistema no llegó al límite de energía cinética pero los últimos 100 pasos temporales tuvieron una variación de energía cinética menor a $1E-3$ J y se corrió un mínimo de simulación de 0.6 segundos se corta la simulación.
- Además, si durante un paso de tiempo hubo alguna partícula que salió por fuera de los límites definidos de la caja, se corta la simulación para poder identificarla rápidamente y repetirla.

La simulación se corre en paralelo utilizando todos los cores disponibles de la CPU para acelerar los tiempos, dada la complejidad del sistema. Por ejemplo, una simulación con 4500 partículas haciendo 5 repeticiones tarda alrededor de 3.5 hs corriendo en paralelo haciendo uso intensivo de los recursos, en cambio, de no correrse en paralelo la simulación tardaría aproximadamente 4 a 5 veces más.

En cada paso temporal se realizan las siguientes acciones:

- Si la condición de corte no se cumple:
 - Para cada partícula se obtiene la fuerza resultante de sus colisiones en este Δt siguiendo las ecuaciones definidas anteriormente.
 - Se aplica el método integrador de Beeman para cada partícula obteniendo una nueva posición y velocidad.

El cálculo de la fuerza resultante para una partícula en un intervalo de tiempo se separa en las siguientes funciones:

- Fuerza gravitacional aplicada sobre la partícula.
- Fuerza de choque de la partícula con otras partículas vecinas si hay superposición entre ellas.
- Fuerza de choque de la partícula con las paredes si hay superposición.
- Finalmente, se suman las 3 fuerzas dando lugar a la fuerza resultante aplicada sobre la partícula en el intervalo de tiempo Δt .

Todos los parámetros del sistema que se pueden variar son parametrizables en la llamada al simulador, por lo que para simplificar las ejecuciones se desarrolló un script *ejecutor* en Python donde se definen las variaciones a realizar y el mismo script ejecuta el jar del motor de simulación con los parámetros definidos, la cantidad de repeticiones que se hayan definido para cada simulación.

3.2 Post procesamiento

Se realizó todo el post procesamiento en Python utilizando el siguiente pipeline para el análisis:

1. Se define la variación a analizar y la cantidad de repeticiones de la variación.
2. Se abren los archivos de cada repetición y se agregan los valores necesarios generando los datasets necesarios para los gráficos.
3. Se grafican los observables definidos anteriormente.

3.3 Animaciones

Para visualizar las animaciones se utilizó el formato XYZ [2]. Para ello se leen los archivos .xyz de salida de cada simulación y se observa en 3D cómo evoluciona la misma en el tiempo.

4 Simulaciones

Las partículas se generan de forma aleatoria con un diámetro distribuido uniformemente entre $[p_{\text{LowDiam}}, p_{\text{UpperDiam}}]$, además, la masa de cada partícula se genera en base a su diámetro mediante una regla de 3 simple: Se define la masa para la partícula de menor diámetro y para la partícula con un diámetro $d_i \neq p_{\text{LowDiam}}$ se define m_{pi} la masa para la partícula i cómo:

$$m_{pi} = d_i * p_{\text{mass}} / p_{\text{LowDiam}}$$

Una vez inicializadas todas las partículas, se simula la caída de las mismas sin la bala para permitir estabilizar el sistema antes de agregar la caída de la bala.

Una vez estabilizado el sistema, se agrega la bala y se libera con velocidad inicial $ball_{\text{velocity}}$. Todas las partículas antes estabilizadas comienzan con la posición obtenida posterior estabilización. La bala es disparada en la posición $p_b = (\text{boxWidth} / 2, \text{boxWidth} / 2, p_{z\text{Highest}} + 0.1)$. Es decir, la bala se centra en el cajón en x e y y se dispara a una distancia en z igual a la posición de la partícula que se encuentra en el punto más alto en z más una distancia de 0.1 mts para generar un mínimo recorrido sin colisiones.

Los parámetros bases utilizados para toda la simulación se definen como:

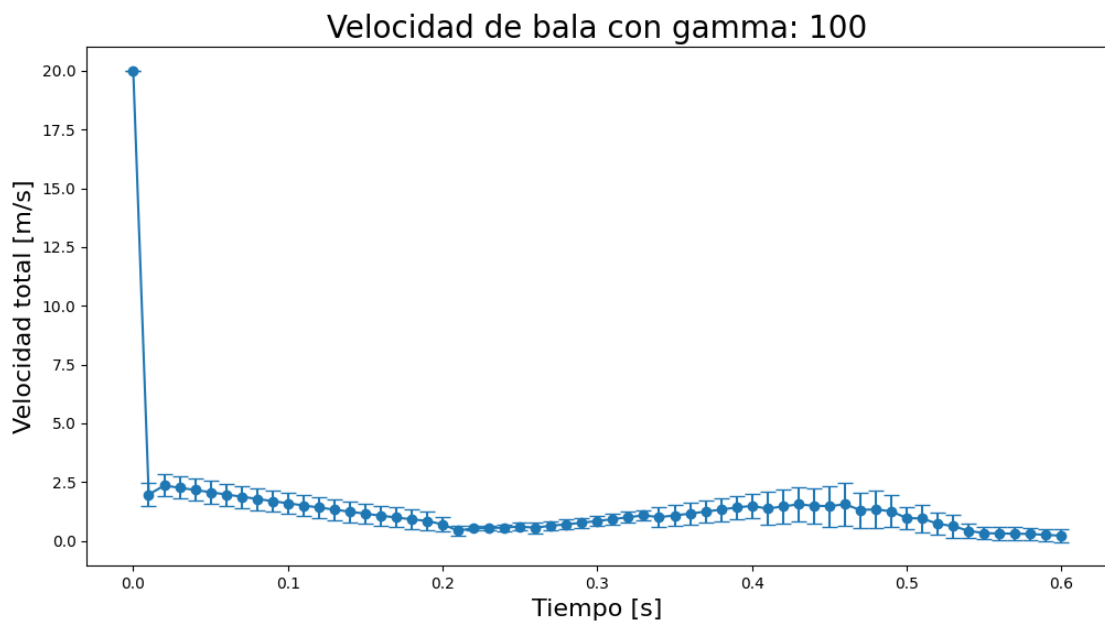
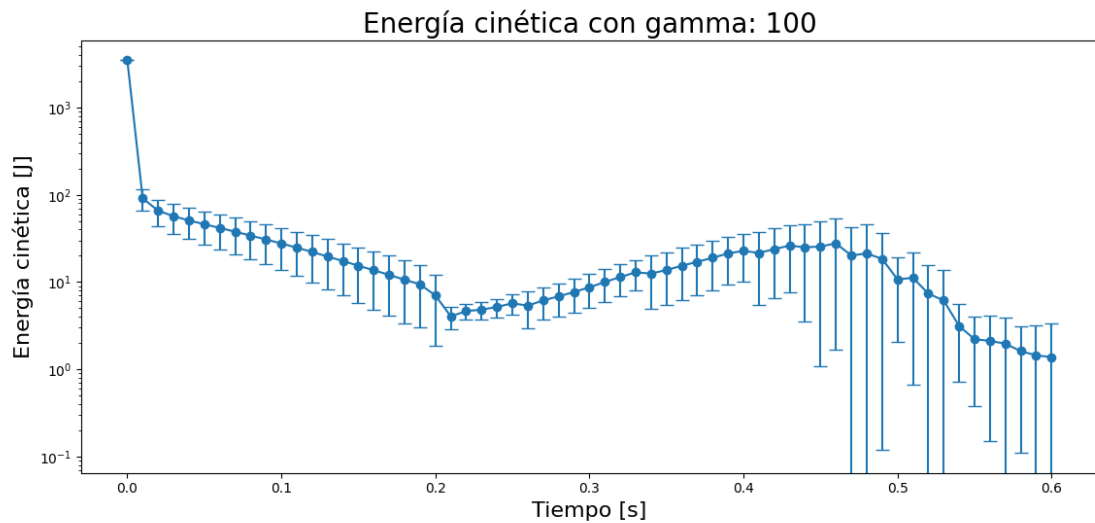
- $n_{\text{particles}}$ (número de partículas contenidas en el cajón) = 2000
- p_{mass} (masa mínima de las partículas) = 0.085 kg
- p_{LowDiam} (extremo menor de intervalo de diámetro) = 0.015 mts
- $p_{\text{UpperDiam}}$ (extremo mayor de intervalo de diámetro) = 0.03 mts
- p_{Kn} (K_n utilizado para las partículas) = $2E6$
- p_{Kt} (K_t utilizado para las partículas) = $2 * p_{Kn}$
- p_{Gamma} (Gamma) = 100
- $wall_{Kn}$ (K_n utilizado para las paredes) = p_{Kn}
- $wall_{Kt}$ (K_t utilizado para las paredes) = $2 * wall_{Kn}$
- $ball_{\text{mass}}$ (Masa de la bala) = 17.5 kg
- $ball_{Kn}$ (K_n utilizado para la bala) = p_{Kn}
- $ball_{Kt}$ (K_t utilizado para la bala) = $2 * ball_{Kn}$
- $ball_{\text{Gamma}}$ (Gamma utilizado para la bala) = 50
- $ball_{\text{Angle}}$ (Ángulo de caída de la bala respecto del eje z) = 90 grados
- $ball_{\text{Velocity}}$ (Magnitud de la velocidad de la bala) = 20 m/s
- $ball_{\text{Diameter}}$ (Diámetro de la bala) = 0.175 mts
- $gravity$ (Gravedad del sistema) = 9.81
- $boxWidth$ (Lados de la caja contenidos en el plano XY) = 0.4 mts

5 Resultados y Conclusiones

5.1 Variación de gamma

El valor de p_{gamma} se varió entre 100 y 300 en pasos de 25.

Variación con $p_{\text{gamma}}=100$ ([Link](#) a animación)



Variación con $p_{\text{gamma}}=200$ ([Link](#) a animación)

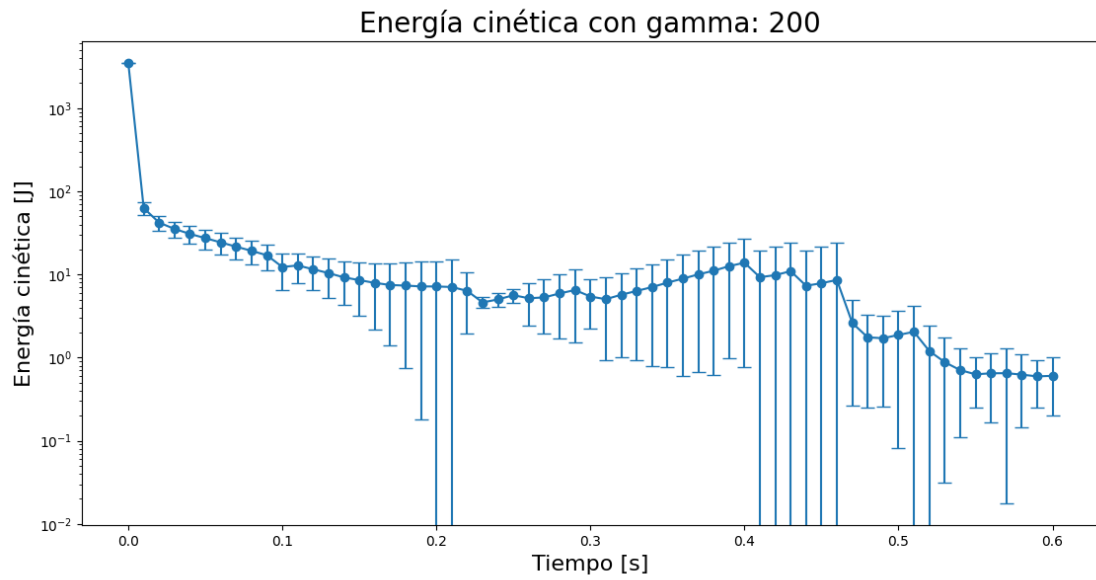


Fig. 4: Energía cinética en el tiempo con gamma 200

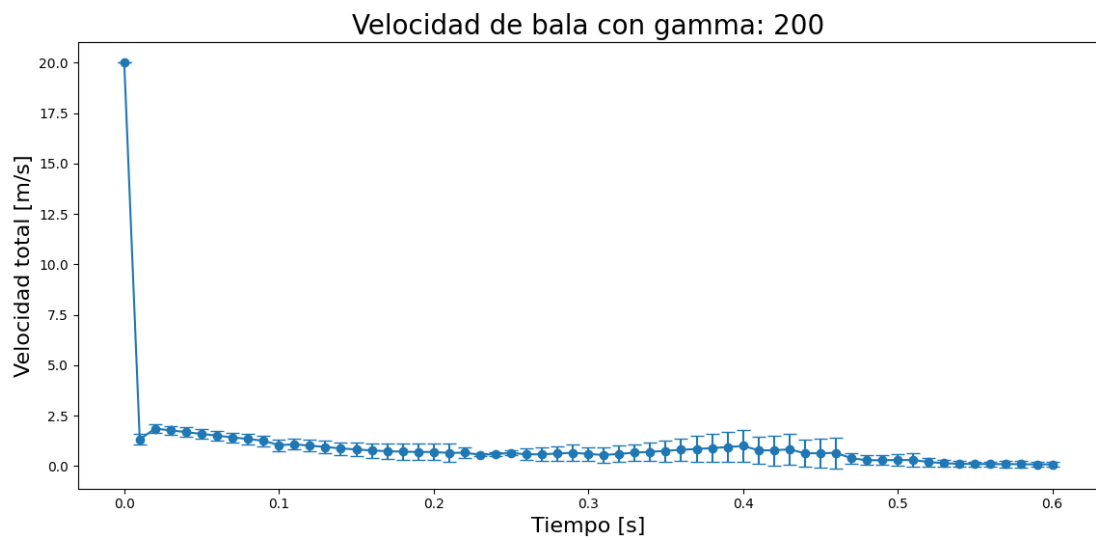


Fig. 5: Velocidad de la bala en el tiempo con gamma 200

Variación con $p_{\text{gamma}}=200$ ([Link a animación](#))

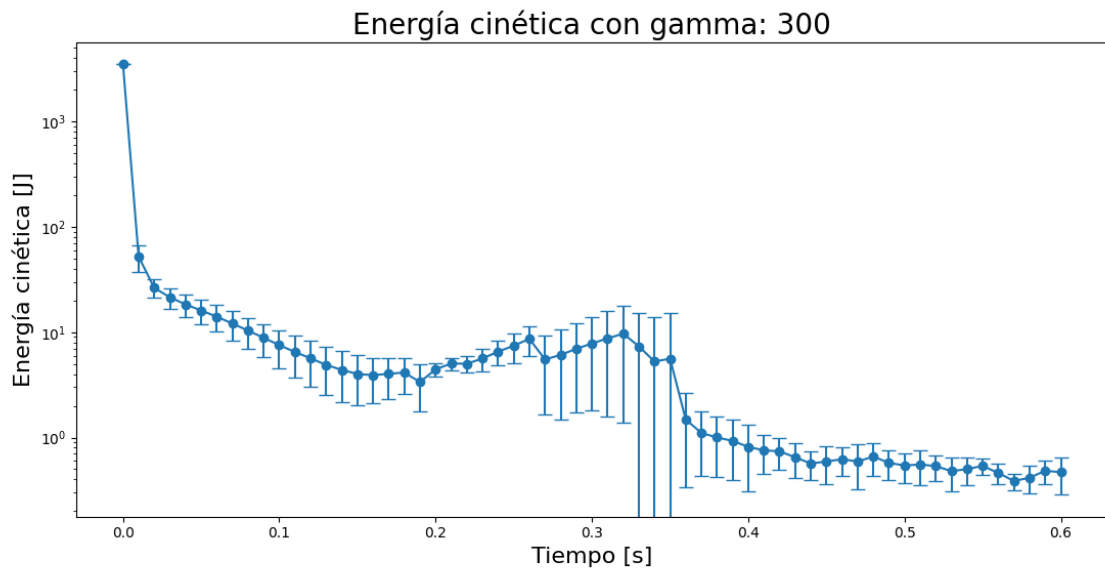


Fig. 6: Energía cinética en el tiempo con gamma 200

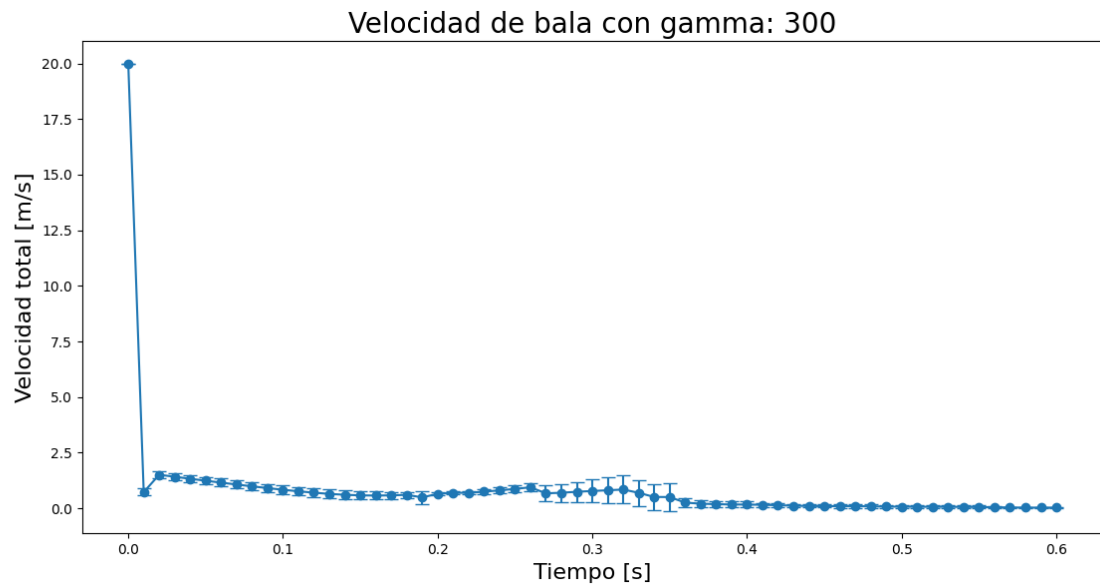


Fig. 6: Energía cinética en el tiempo con gamma 200

Utilizando todas las variaciones entre 100 y 300 en pasos de 25.

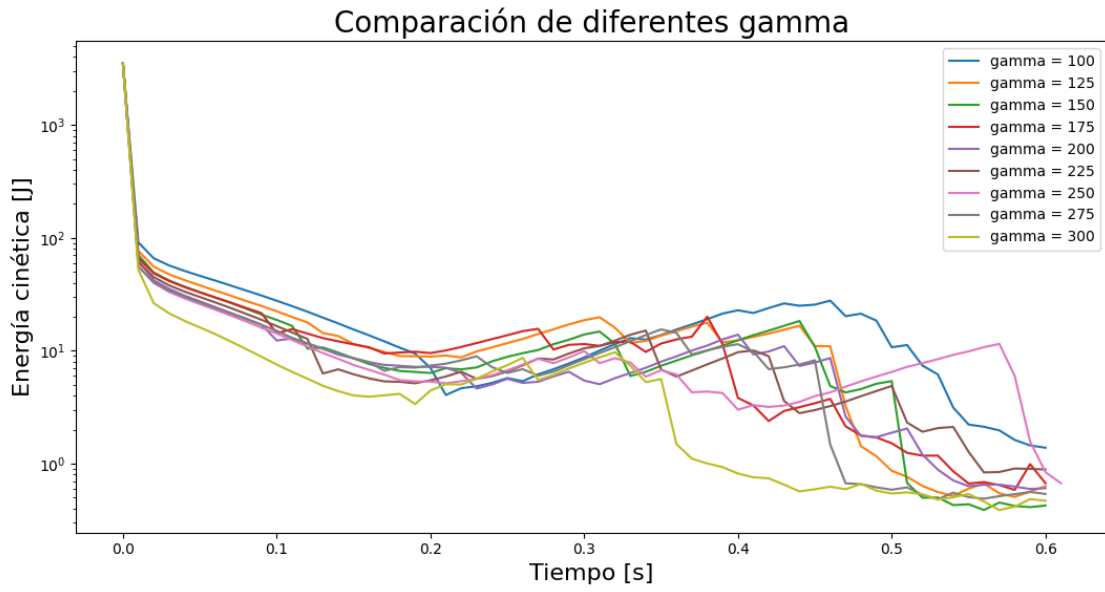


Fig. 7: Energía cinética promedio en diferentes gamma

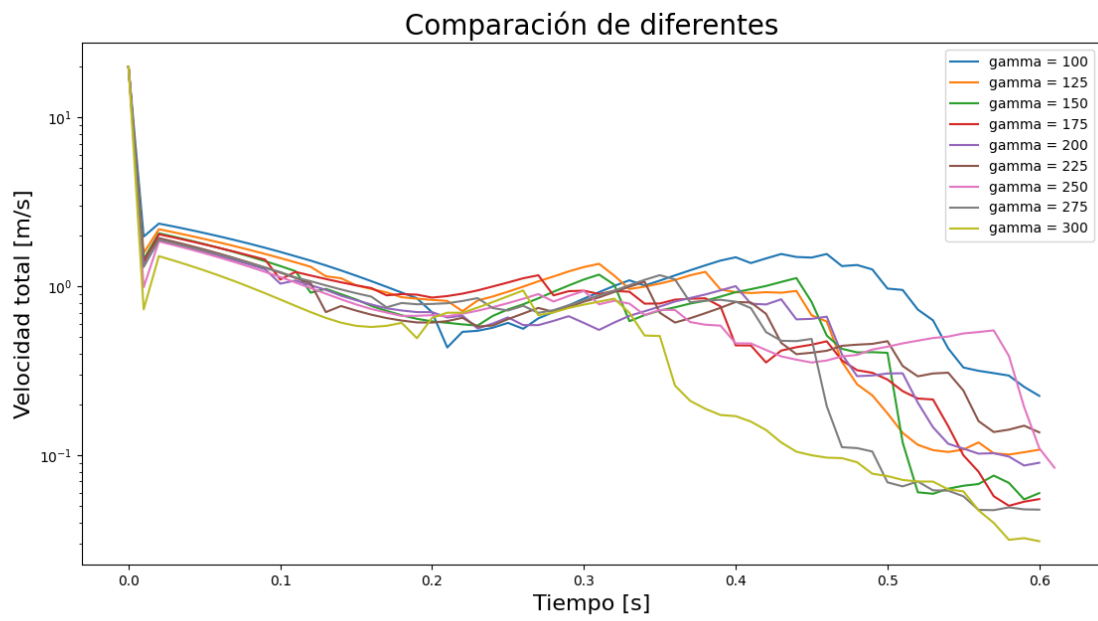


Fig. 8: Velocidad de la bala promedio en diferentes gamma

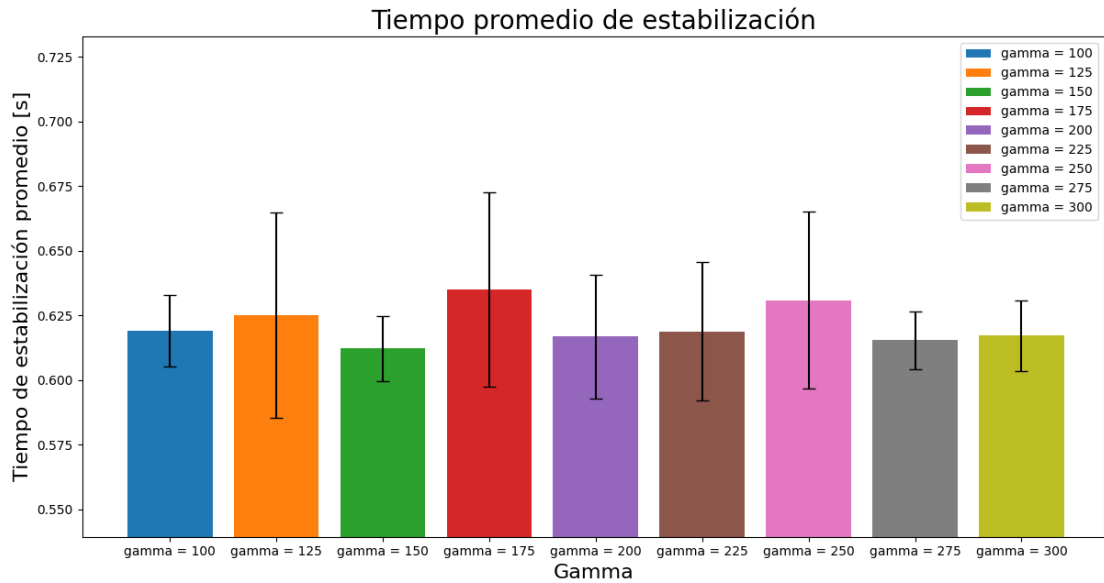


Fig. 9: Tiempo a estabilización promedio en diferentes gamma

5.2 Variación de diámetros

Los diámetros se variaron en 2 intervalos:

- $d_2 = [0.02, 0.025]$
- $d_3 = [0.025, 0.03]$

Es decir, los diámetros de las partículas se inicializan con una distribución uniforme en d_2 y d_3 .

Variación de $d_3 = [0.025, 0.03]$ ([Link](#) a animación)

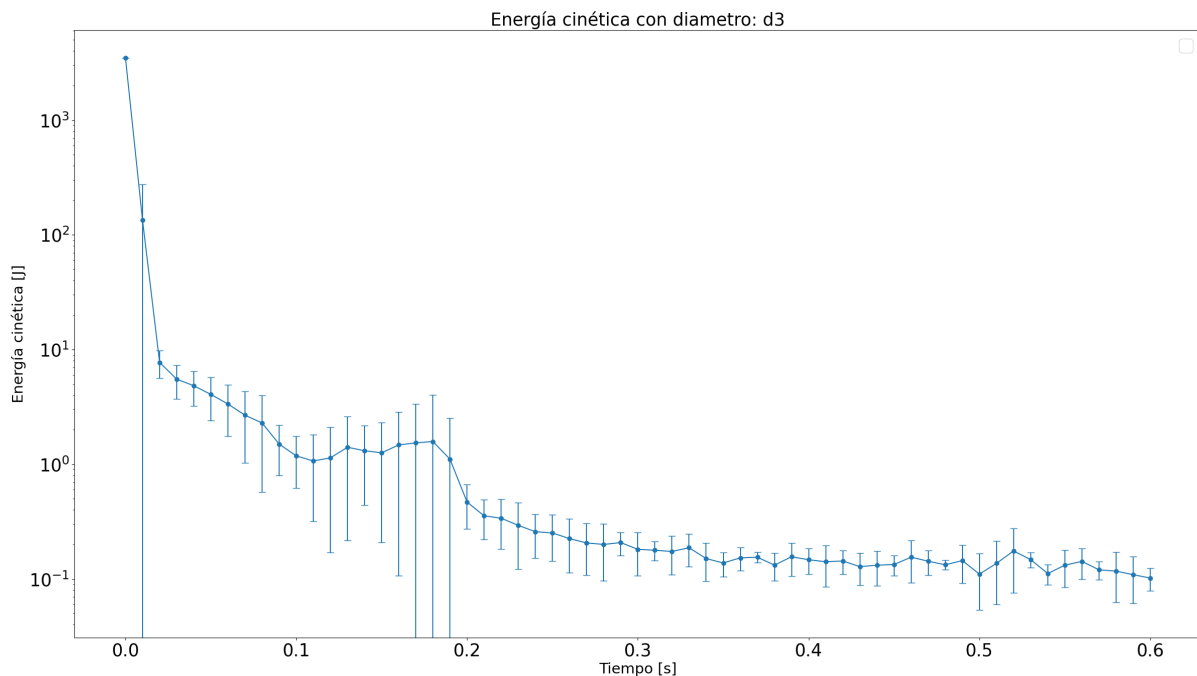


Fig. 10: Energía cinética promedio con diámetros d_3

Utilizando d2 y d3

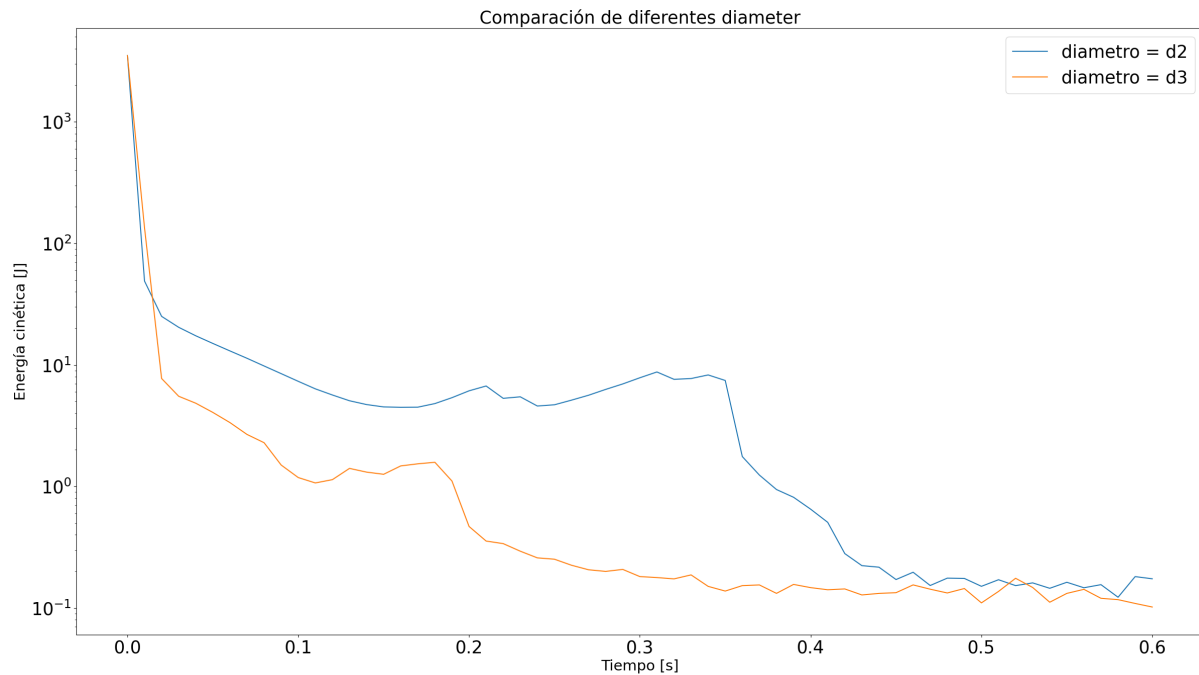


Fig. 11: Energía cinética promedio con diámetros d2 y d3

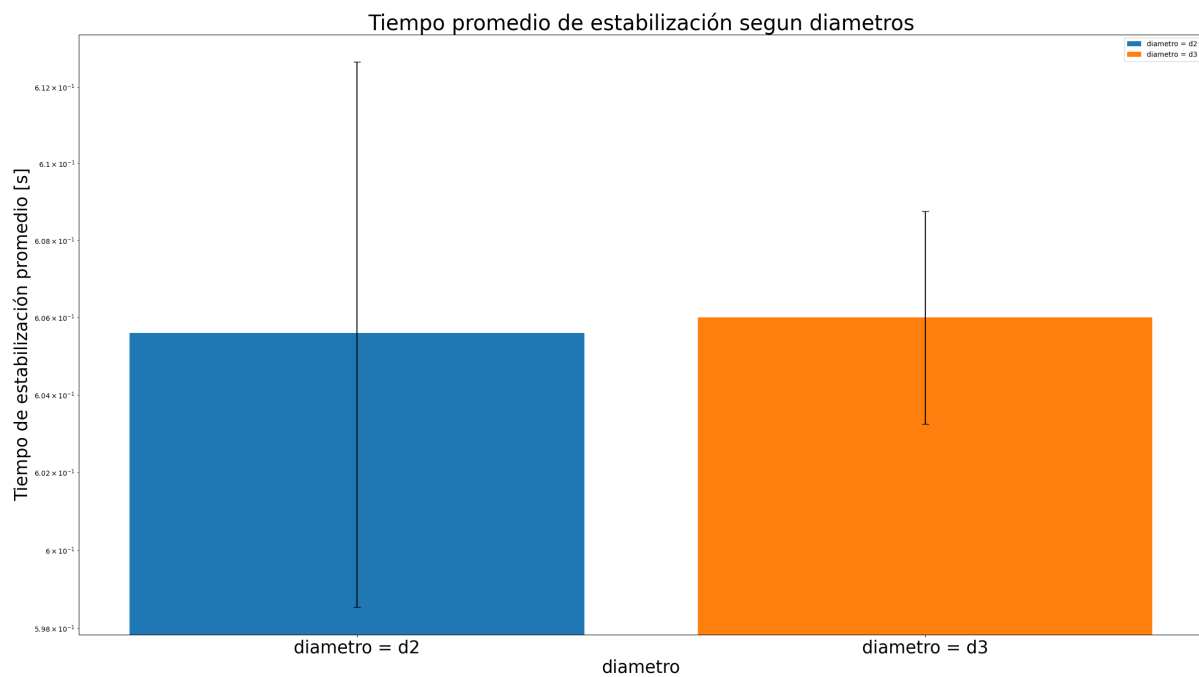


Fig. 12: Tiempo promedio hasta estabilización con diámetros d2 y d3

5.3 Variación de ángulo de caída de la bala

El ángulo de caída de la bala b_{angle} se varió entre 95 y 110 grados de a pasos de 5 grados.

Variación con ángulo de 75 grados ([Link](#) a la animación)

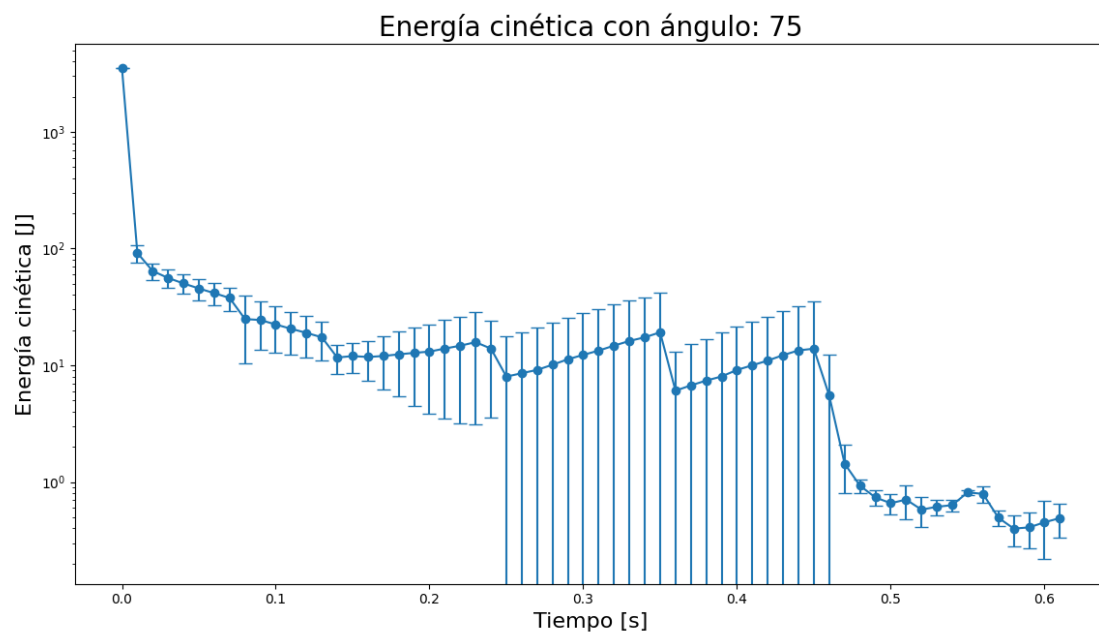


Fig. 13: Energía cinética promedio para ángulo 75

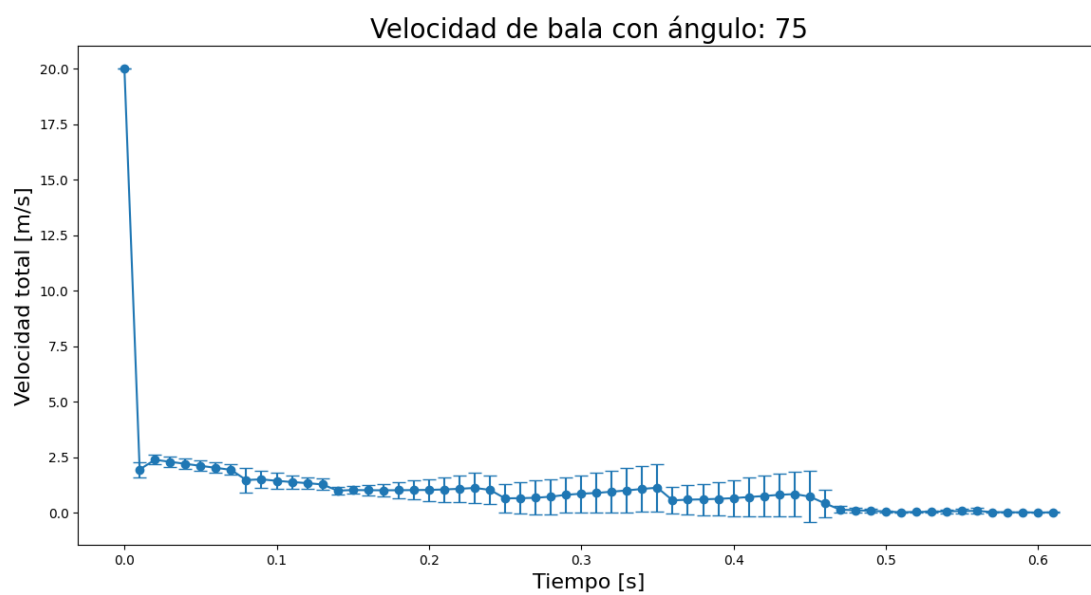


Fig. 14: Velocidad de la bala hasta la estabilización

Con todas las variaciones de 75, 80 y 85 grados

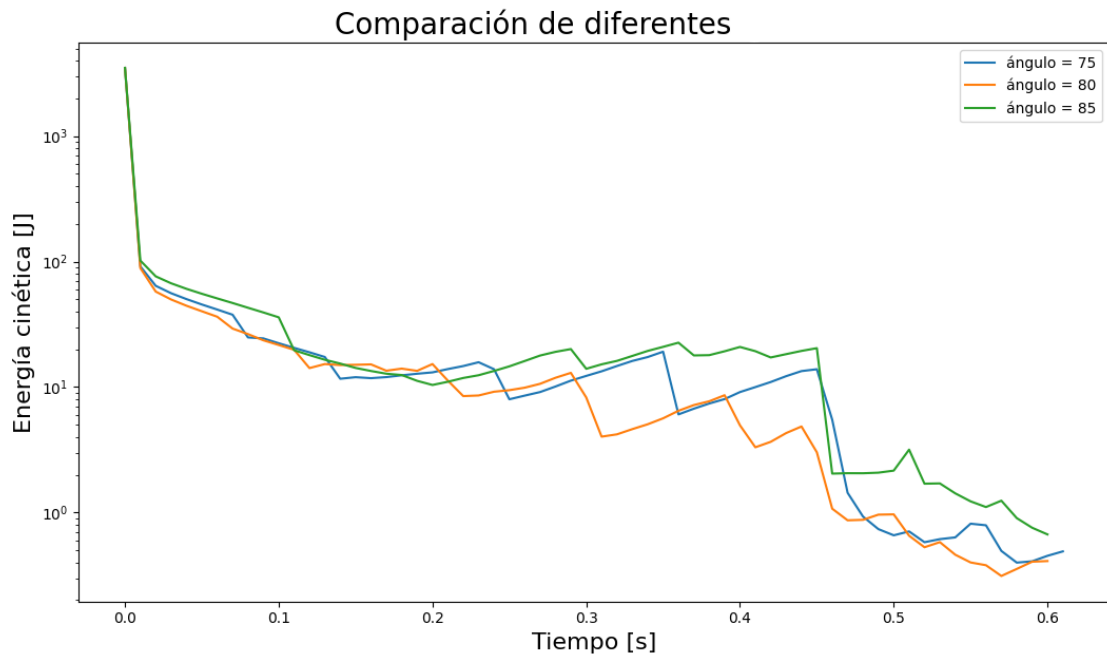


Fig. 15: Comparación la energía cinética promedio de diferentes ángulos de caída de la bala

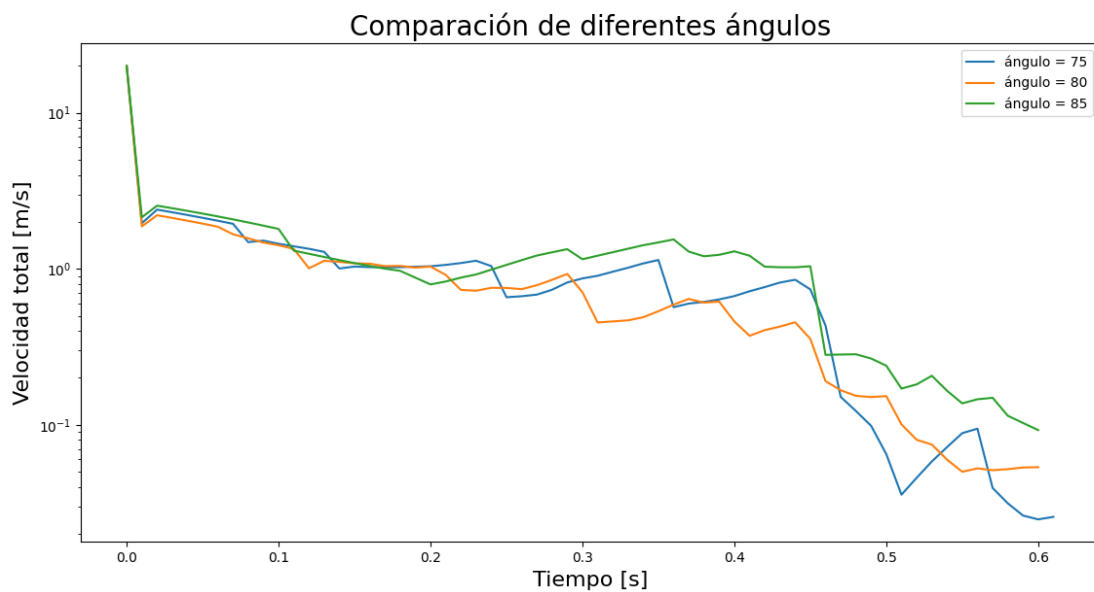


Fig. 16: Comparación la velocidad de la bala promedio de diferentes ángulos de caída de la bala

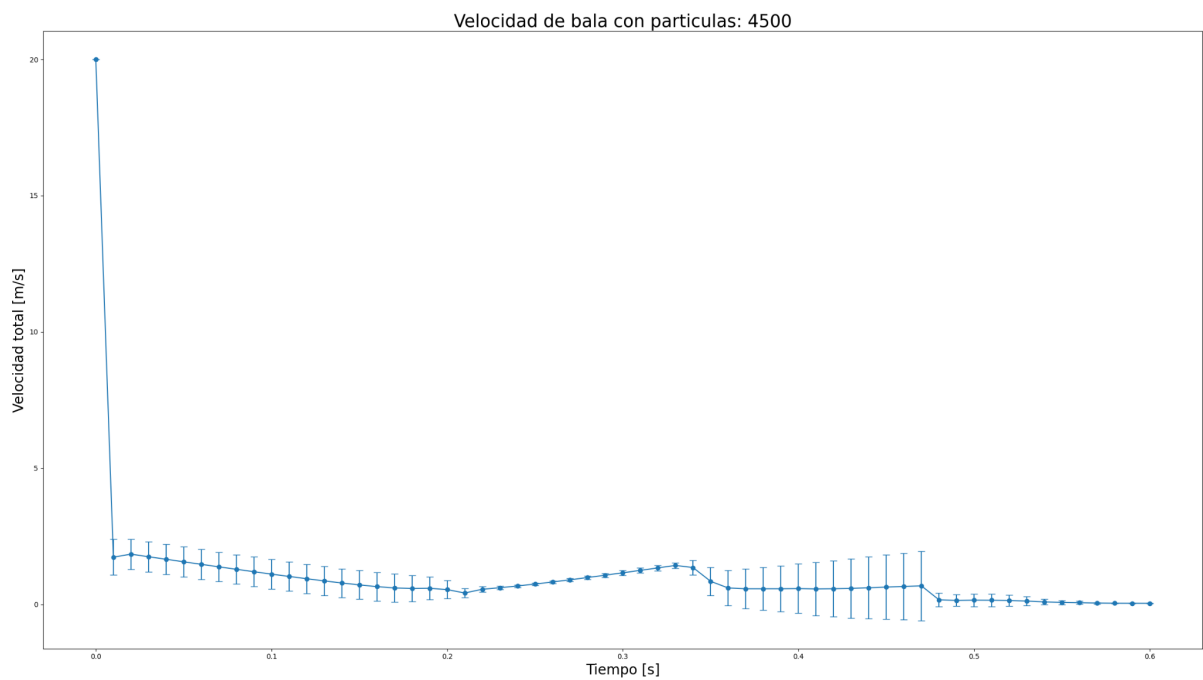
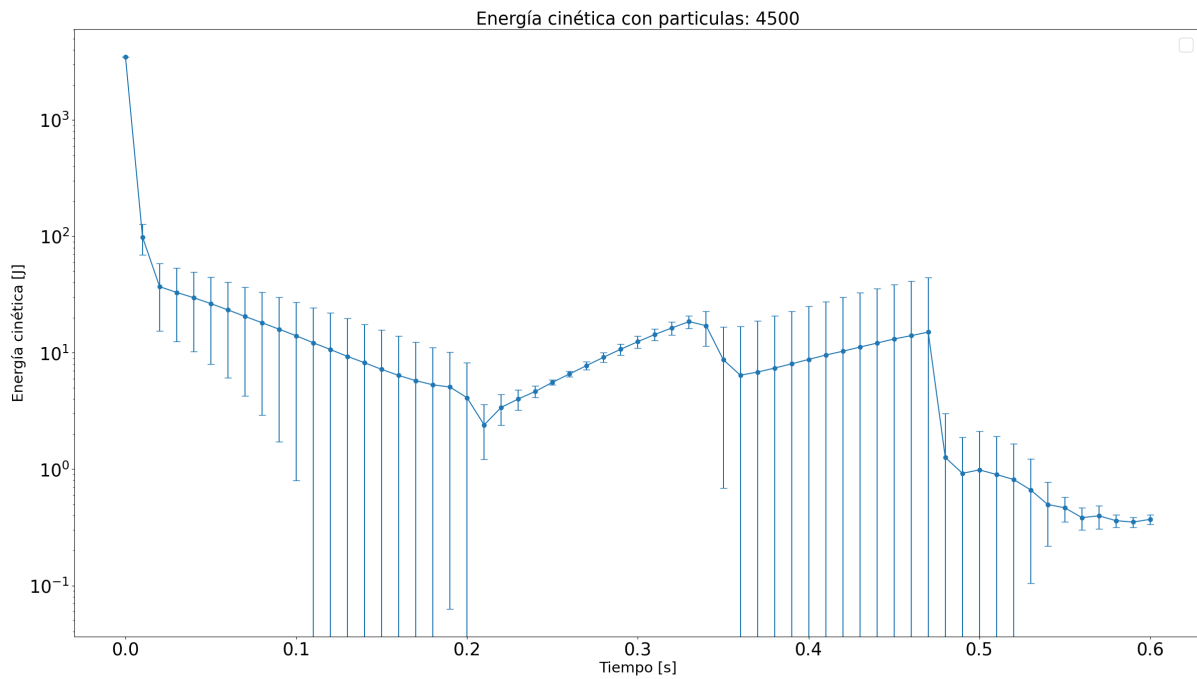
5.3 Variación de cantidad de partículas

La cantidad de partículas $n_{\text{partículas}}$ en el sistema se varió entre los siguientes valores:

- 2000 partículas
- 3000 partículas

- 4500 partículas

Variación con 4500 partículas ([Link](#) a animación)



Utilizando las tres variantes de partículas

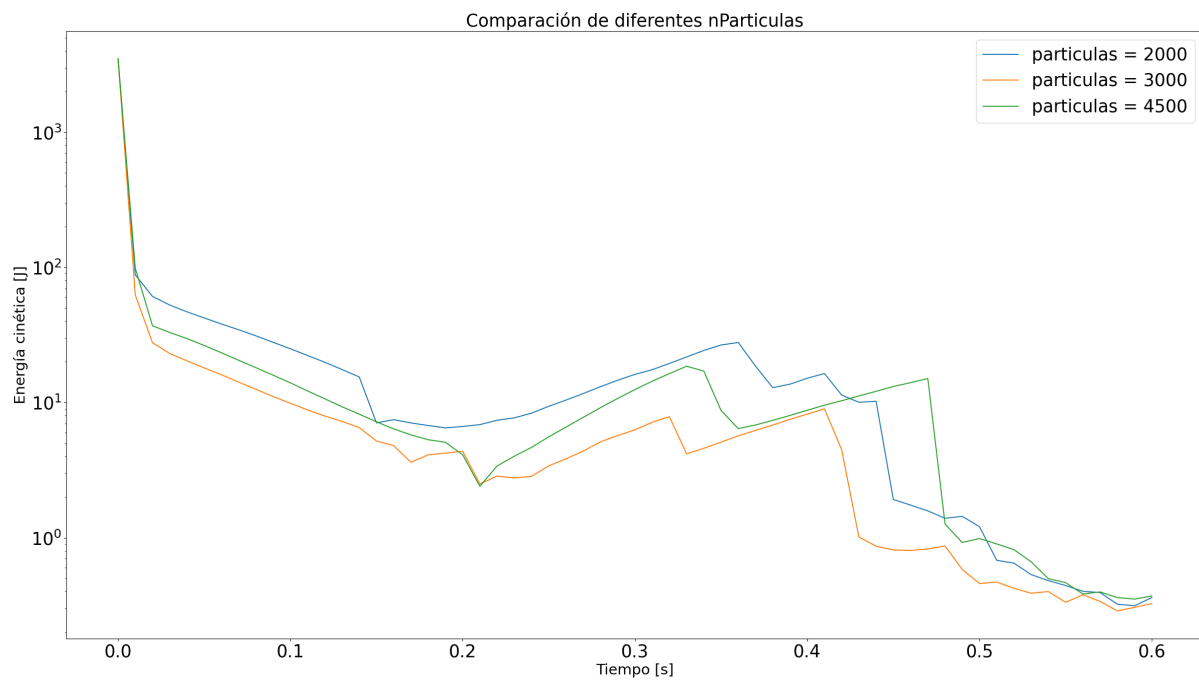


Fig. 19: Energía cinética promedio en el tiempo con diferentes cantidades de partículas

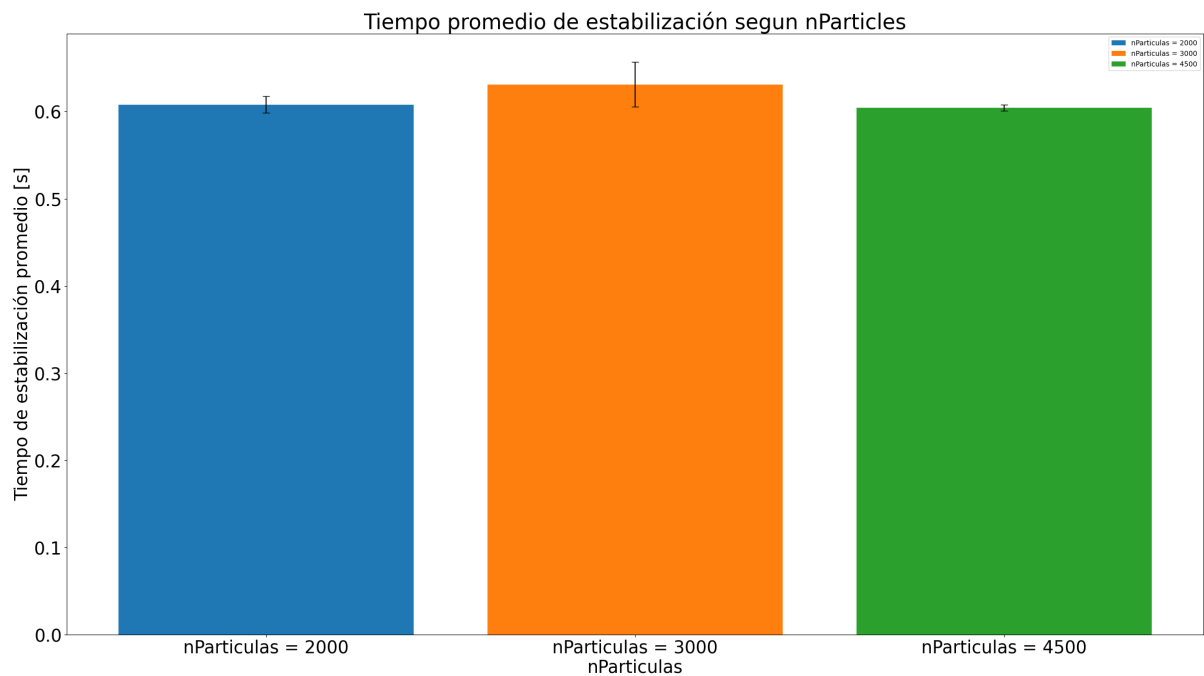


Fig. 20: Tiempo a estabilización con diferentes cantidades de partículas

5.4 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.

6 Referencias

- [1] [Beeman's Algorithm](#)
- [2] [XYZ Format](#)