

Práctica 3 – Modelos de Colisión de Partículas

Duración: 2 Sesiones

Objetivos:

- Implementar, visualizar y analizar el comportamiento de dos modelos de simulación de colisión de partículas.
- Analizar y mejorar la escalabilidad de la simulación cuando el número de partículas crece.

Entrega: 1 semana después de finalizar la práctica. Se debe entregar el código y una memoria/documento donde se analicen los resultados obtenidos y se interpreten/justifiquen esos resultados.

Evaluación: a través de la entrega del código y del documento de análisis.

Introducción

Esta práctica consta de tres partes diferenciadas. La primera es la implementación y análisis del modelo de colisión entre partículas basado en velocidades. Este se expone a lo largo de la asignatura y se basa en reproducir el comportamiento de los choques semi-elásticos entre partículas. Para ilustrar este modelo de colisiones se propone como ejercicio la simulación de un billar francés.

En la segunda parte de la práctica se implementará el modelo de colisión basado en muelles. Este modelo también se expone a lo largo de la asignatura y básicamente consiste en la creación de muelles entre las partículas cuando sus posiciones se aproximan demasiado. Para ilustrar este modelo de colisiones se propone como ejercicio la simulación de un recipiente lleno de partículas que caen dentro del mismo, de modo que el conjunto de partículas se comporte como un fluido.

En la tercera parte de la práctica, una vez implementado este segundo modelo de colisiones, nos centraremos en aumentar la escalabilidad del sistema. Para ello utilizaremos métodos de detección rápida de colisiones que aumentan el rendimiento y permiten una mejor escalabilidad de la simulación. Concretamente, implementaremos y compararemos la gestión de colisiones mediante la estructura de *grid* y mediante la de tabla *hash*. Ambas estructuras de datos se exponen también en la asignatura.

Las simulaciones se deben realizar en un espacio 2D. Se proporciona una pequeña porción de código como material de apoyo para la práctica.

1- Billar Francés - Modelo de Colisión Basado en Velocidades

En esta parte, se debe implementar un modelo de colisiones basado en velocidades con sus tres pasos: (i) detección de la colisión, (ii) restitución a la posición previa a la colisión y (iii) cálculo de la velocidad de salida.

Para ello, se debe implementar una simulación de billar francés (sin troneras), en el que 5 bolas se sitúen sobre una mesa de billar, en posiciones inicialmente aleatorias (ver Figura 1). Se deben utilizar las dimensiones y masas reales de las bolas y la mesa de un billar francés.

Una vez las bolas estén colocadas, el programa permitirá seleccionar una de ellas y simular que es golpeada con un taco de billar. La simulación del golpeo consistirá en modificar la velocidad instantánea de la bola en la dirección del taco, de modo que, al moverse, rebote con otras bolas y/o con los bordes (llamados bandas) de la mesa y se produzcan colisiones. La interfaz elegida para elegir la bola y usar el taco es libre, aunque debe ser explicada con un mensaje de ayuda visible en la pantalla. También debe implementarse la posibilidad de que todas las bolas adquieran velocidades aleatorias y comiencen a moverse simultáneamente (con la tecla M). Finalmente, debe ser posible elegir (con la tecla C) si se computan o se ignoran las colisiones entre las bolas, y debe poderse resetear la simulación (tecla R) de modo que se genere una nueva disposición de las mismas.

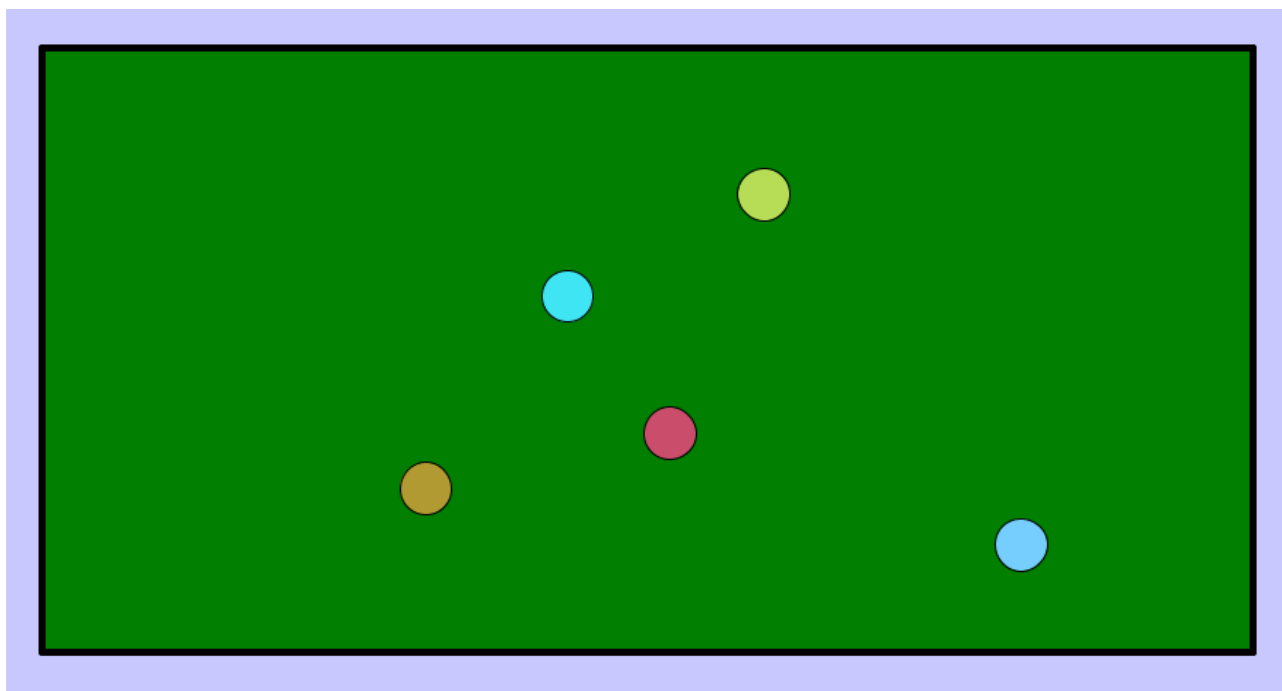


Figura 1 – Simulación de un billar francés.

En esta parte de la práctica, la detección de colisiones entre bolas debe seguir un esquema de comprobación de “todas contra todas”. Es decir, para cada bola se comprueba si existe colisión con todas las demás (además de con las bandas).

Se asumirán las siguientes condiciones:

- Existirá una fricción entre las bolas y la superficie de la mesa, que será proporcional a la fuerza normal ejercida por la mesa sobre las bolas. La constante de proporcionalidad será μ_d . Esta fuerza podrá habilitarse/deshabilitarse con la tecla F .
- Existirá una pérdida de energía por cada colisión de cada bola con las bandas de la mesa. Esto se debe modelar mediante una constante C_{r1} (con valores entre 0 y 1) que represente la pérdida de energía sufrida en cada colisión bola-banda.
- Existirá una pérdida de energía por cada colisión entre bolas. Esto se debe modelar mediante una constante C_{r2} (con valores entre 0 y 1) que represente la pérdida de energía sufrida en cada colisión entre bolas.

Una vez la simulación funcione con 5 bolas, se debe probar a incrementar el número de bolas (n) hasta probar con $n = 100$. ¿Observas alguna diferencia en la estabilidad del sistema según aumenta el número de bolas? ¿Por qué? ¿Cambia algo si no hay fricción? ¿Por qué?

Finalmente, para comprobar mejor la estabilidad de la simulación, se debe simular el efecto de atraer las bolas hacia una zona de la mesa, de modo que todas las partículas se acumulen y golpeen constantemente entre sí y con las bandas. Por simplicidad, modelaremos este efecto como una fuerza constante \vec{F}_c en una dirección fija que apunte hacia alguna de las esquinas de la mesa. Comienza con $n = 5$ bolas, y ve incrementando el número de bolas ¿Observas alguna diferencia en la estabilidad del sistema con respecto al caso anterior? ¿Por qué?

2- Simulación de un Fluido - Modelo de Colisión Basado en Muelles

En esta segunda parte, la simulación consistirá en un recipiente definido por varios planos (ver Figura 2) que formarán el fondo y las paredes del mismo, sobre el que dejaremos caer una fuente de partículas, de masa m y radio r , que representarán un fluido. Las partículas caerán hacia el fondo del recipiente por acción de la gravedad. Estas partículas, además, colisionarán con las paredes del mismo y también entre sí. El tamaño de las partículas será el adecuado para poder introducir en el interior del recipiente **cientos o incluso miles de partículas**. Es importante tener en cuenta que **el radio, la masa de las partículas y otros factores influyen en la estabilidad del modelo, por lo que deben ser constantes del problema, para que se puedan modificar fácilmente**.

En esta parte de la práctica, la detección de colisiones entre partículas debe seguir un esquema de comprobación de “todas contra todas”. Es decir, para cada partícula se

comprueba si existe colisión con todas las demás (además de con las paredes del recipiente).

Se asumirán las siguientes condiciones:

- Existirá un rozamiento entre las partículas y su entorno, que será linealmente proporcional a la velocidad de las mismas. La constante de proporcionalidad será K_d .
- Existirá una pérdida de energía por cada colisión de cada partícula con las paredes del recipiente. Esto se debe modelar mediante una constante C_r (con valores entre 0 y 1) que represente la pérdida de energía sufrida en cada colisión partícula-pared.
- El comportamiento de los choques entre partículas vendrá determinado por la constante elástica de los muelles de colisión (K_e), y por la distancia a la que se activen los muelles (d_m), que representará la distancia mínima a la que se permite que se acerquen las partículas sin que exista repulsión elástica. Esta distancia suele tener como valor la suma de los radios de las dos partículas, pero se tiene que poder cambiar, por lo que d_m será un parámetro del problema.

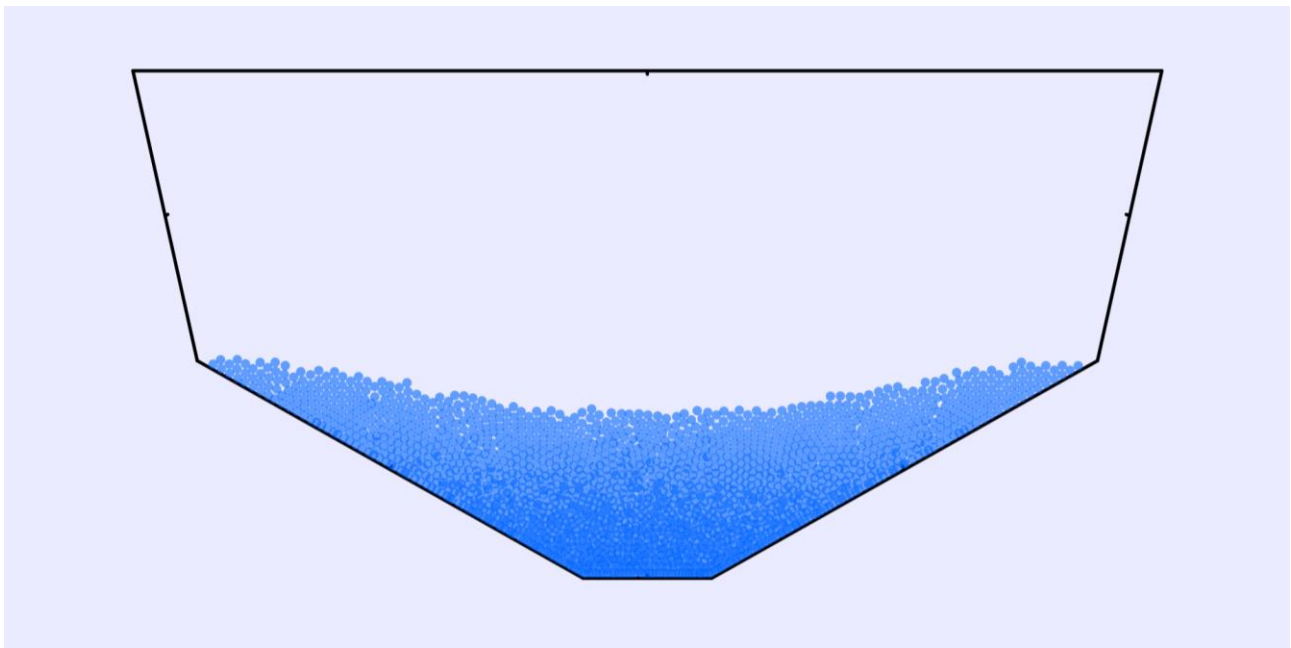


Figura 2 - Forma del recipiente y simulación del fluido mediante partículas.

El trabajo de esta segunda parte de la práctica se dividirá en varias fases:

1. Implementar la colisión partícula-plano para modelar los rebotes con las paredes del recipiente. Esto se implementará usando un modelo basado en velocidades, exactamente igual al de la primera parte de la práctica.
2. Implementar la colisión entre partículas mediante un modelo basado en fuerzas

elásticas (muelles), de manera que se aplicará una fuerza elástica repulsiva si la partícula se acerca demasiado a otra partícula (es decir, si la distancia entre ellas es inferior a d_m), pero no se aplicará una fuerza elástica atractiva (como sí ocurriría en un muelle real) si se separan más allá de d_m .

3. Se deberá implementar, con la tecla P , la eliminación del plano inferior del recipiente, para que las partículas escapen del mismo (ver Figura 3).

4. Se deberá implementar la parametrización del fluido para comportarse como un gas (tecla 1), como un líquido acuoso (tecla 2) o como un fluido viscoso (tecla 3). Lo único que debe hacerse para implementar esto es modificar los parámetros del problema (m , r , K_e , K_d , d_m , etc.) adecuadamente.

5. Analizar el rendimiento obtenido por el método de simulación de colisiones basado en fuerzas elásticas. Para ello, se realizarán las siguientes gráficas:

- a) Gráfica del número de partículas (n) frente a frames por segundo (fps).
- b) Gráfica del número de partículas (n) frente a tiempo de computación por cada paso de simulación (T_c). Este tiempo incluye todo el tiempo necesario para calcular/gestionar colisiones e integrar ecuaciones, **pero no incluye el tiempo de dibujado**.
- c) Gráfica del número de partículas (n) frente al tiempo total de cada iteración (T_T). Este es el tiempo total necesario para completar una iteración de la simulación (incluyendo esta vez el dibujado).

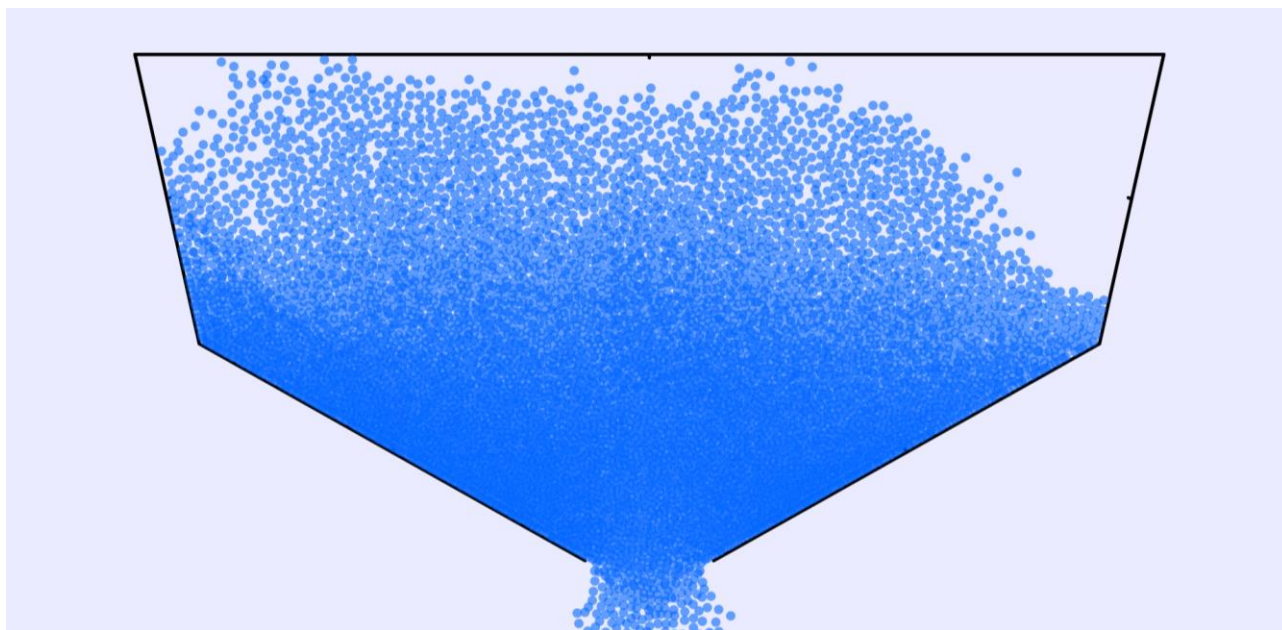


Figura 3 – Apertura de la pared inferior del recipiente.

Se deben generar gráficas de modo que se pueda deducir de qué orden es el coste computacional de la comprobación y simulación de colisiones. Se recomienda usar la parametrización tipo *líquido*, y sobre todo promediar varias medidas para obtener resultados fiables para cada valor de n . En principio, el coste debería crecer proporcionalmente al cuadrado del número de partículas (orden $O(n^2)$). Demuestra con las gráficas que esto es efectivamente así. Sea o no sea así, explica por qué.

3- Estructuras de Datos para Mejorar la Escalabilidad

La estrategia de comprobación de fuerza bruta (“todas contras todas”) genera, en principio, un coste de comprobación de colisiones de $O(n^2)$, donde n es el número de partículas. Cuando n es muy grande, esta estrategia no es en absoluto adecuada.

Para mejorar la escalabilidad del sistema se pueden utilizar estructuras de datos para agrupar las partículas vecinas y comprobar las colisiones sólo en el entorno/vecindario de cada partícula. Estos métodos, aunque llevan un coste asociado a la actualización de las estructuras de datos en cada iteración de la simulación, son más eficientes que la estrategia de fuerza bruta de comprobar todas contra todas si el número de partículas n es suficientemente grande.

El trabajo de esta parte se dividirá en varias fases:

1. Implementar los dos métodos estudiados en la asignatura para la gestión/detección eficiente de colisiones: el método basado en *grid* y el método basado en tabla *hash*.

2. Integrar ambos métodos en el código que has elaborado en la parte 2 y estudiar la escalabilidad del sistema. Para ello, se deben realizar varias gráficas donde se represente el número de partículas (n) frente al tiempo de computación por cada paso de simulación (T_c) para cada uno de los métodos de gestión/detección de colisiones. ¿Alguno de los dos métodos (*grid* o tabla *hash*) es más eficiente en términos de coste computacional temporal? ¿Por qué? Se debe hacer un análisis para diferentes tamaños de celda y/o número de celdas, de modo que se pueda observar cómo depende el rendimiento de estos parámetros.

Compara los resultados con los obtenidos en la parte 2 (con la estrategia de fuerza bruta). Comprueba si el coste computacional temporal es ahora de orden $O(n^2)$ o es diferente. Sea o no sea diferente, justifica cuál es, si es mejor o peor que antes, y por qué.

Para las simulaciones en las que se utilizan estructuras de datos (*grid* o tabla *hash*) para el tratamiento de colisiones, se deben **dibujar del mismo color aquellas partículas que se agrupen en una misma celda** de la estructura de datos (ver Figura 4). De esa forma se podrá comprobar visualmente si el método está correctamente implementado.

Material a Entregar

Se debe entregar el código con las diferentes implementaciones solicitadas. En la implementación de la parte 1, el programa debe permitir resetear la simulación, y debe indicar gráficamente el número de bolas presentes.

Las implementaciones de las partes 2 y 3 deben entregarse ambas juntas en un solo programa, de tal manera que se **pueda elegir interactivamente qué estructura de datos de gestión de colisiones (*grid* (tecla *G*), *hash* (tecla *H*) o ninguna (tecla *N*)) se utiliza para mejorar su escalabilidad**. En la implementación de las partes 2 y 3, el programa debe permitir resetear la simulación, y debe indicar gráficamente el número de partículas, el frame-rate y el tiempo (**en milisegundos**) que cuesta en cada iteración actualizar las estructuras de datos de gestión de colisiones, computar las colisiones, calcular la simulación (integración numérica de las ecuaciones), y dibujar la simulación. **Cada uno de esos tiempos debe indicarse por separado.**

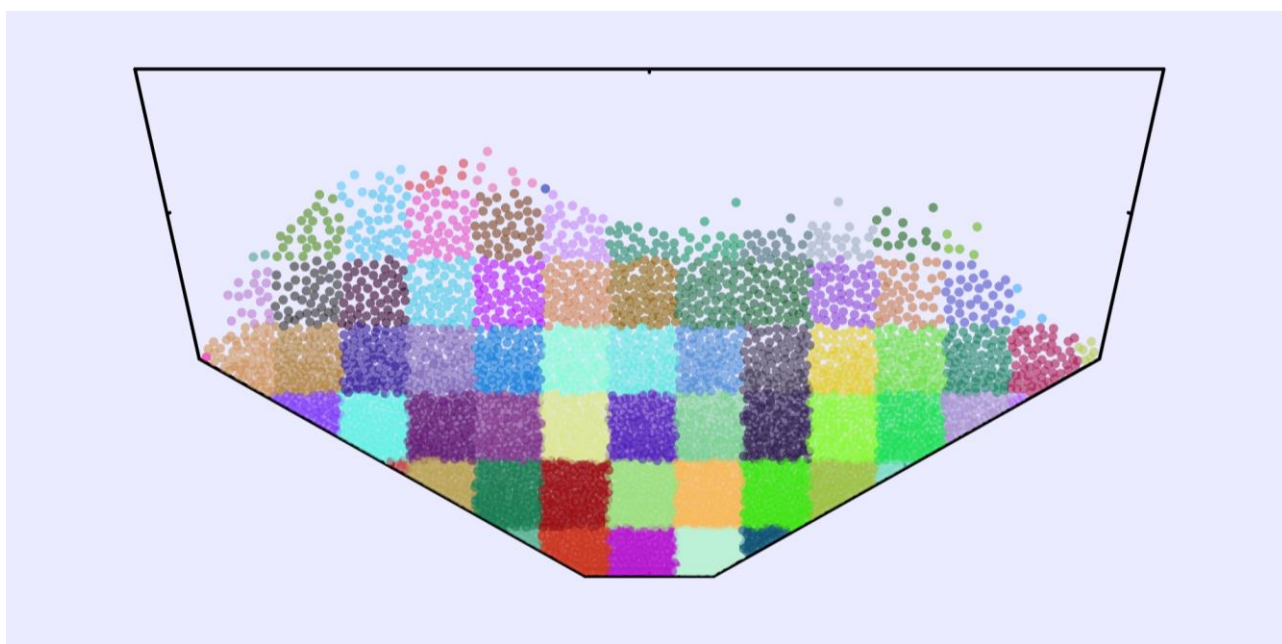


Figura 4 – Coloreado de partículas para evidenciar la estructura de datos de gestión de colisiones.

Se debe entregar también un documento donde se muestren las gráficas solicitadas y se expliquen, interpreten y justifiquen los resultados obtenidos. Debe quedar claro qué se muestra en cada gráfica, y qué conclusiones se obtienen a partir de ellas. También deben contestarse todas las preguntas y cuestiones formuladas en el enunciado de la práctica.

Nota importante: indica en el documento la configuración del ordenador donde se han hecho las pruebas, y también los parámetros que sean de interés para la comparación de

gráficas como, por ejemplo: tamaño de las celdas, número de celdas, radio de las partículas, paso de simulación, valores de las constantes del problema, etc.

Opcional 1: modifica la simulación de la parte 1 para que el billar sea un billar americano (con troneras), de modo que las bolas puedan caer en dichas troneras.

Opcional 2: modifica la simulación de la parte 2 para que sea en 3D.