

# Reporte proyecto intermedio - Simulación dispersión de un liquido en un contenedor

Pinilla Correa Santiago, Murcia Pimentel Sergio Alfonso, Penuela Rodriguez Oscar David y Rojas Rojas Francisco Josse \*

*Introducción a la Computación Científica de Alto Rendimiento, Departamento de Física, Universidad Nacional de Colombia-Bogotá, Cundinamarca, Colombia.*

---

## I. Resultados de la simulación:

El presente trabajo tuvo como objetivo lograr una simulación de la difusión de un conjunto de partículas en un contenedor, emulando el comportamiento el comportamiento de la difusión de un liquido en otro. Se buscó comprobar mediante la simulación predicciones hechas por la mecánica estadística, en específico mostrar cómo un sistema tiende al equilibrio.

Para la simulación se uso como base la programación orientada a objetos que permite  $C++$ , de manera que se lograra un código más fácil de modularizar. Es así que se definió el objeto partícula, el cual tiene como propiedades una posición en  $x$  y  $y$ , así como una función que cambia alguno de estos valores aleatoriamente en una unidad para lograr un caminante aleatorio. Dado que se quiere realizar una simulación con múltiples partículas, la facilidad que otorga éste enfoque es la posibilidad de trabajar con un vector de partículas accediendo fácilmente a las propiedades de cada una.

Una vez hecho esto, se crean funciones que trabajen con las posiciones de estas partículas. Para comprobar el correcto funcionamiento de la simulación se verificaron diferentes predicciones teóricas. En primer lugar la entropía de todo sistema tiende a crecer en el tiempo y llegar a un máximo eventualmente, el cual se interpreta como el estado más probable en que se puede encontrar el sistema. En éste caso se puede encontrar el valor de la entropía mediante la formula:

$$S = - \sum_i P_i \ln(P_i) \quad (1)$$

Siendo  $P_i$  la probabilidad de encontrar la partícula en una

región del contenedor. Para la presente simulación se usó como estándar dividir el contenedor en 64 partes, es decir una cuadrícula  $8 \times 8$ . Se tiene que el estado de máxima entropía será en donde todas las regiones tengan la misma probabilidad, es decir la misma cantidad de partículas. Si se tienen  $n$  partículas la entropía será:

$$\begin{aligned} S_{max} &= - \sum_{i=1}^{64} \frac{n/64}{n} \ln\left(\frac{n/64}{n}\right) \\ S_{max} &= - \sum_{i=1}^{64} \frac{1}{64} \ln\left(\frac{1}{64}\right) = -\ln\left(\frac{1}{64}\right) \\ S_{max} &\approx 4.16 \end{aligned}$$

Por lo tanto, independientemente de las condiciones del sistema, la entropía tiende a éste valor. Como se observa en la figura 1, la entropía efectivamente tiende a este valor:

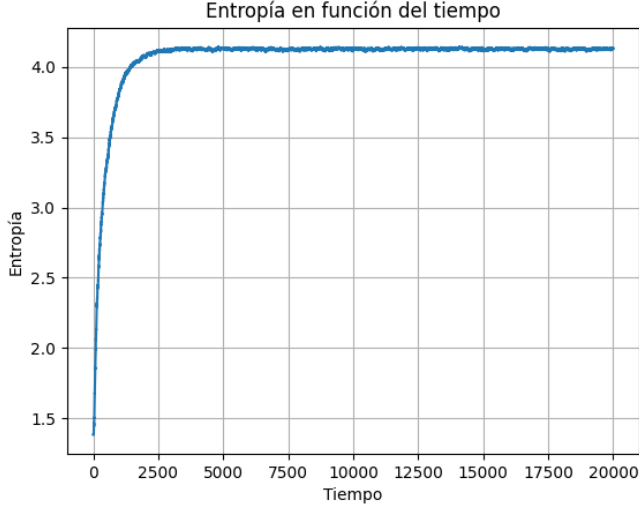


Figura 1: Entropía en función del tiempo para 1000 partículas en la simulación y un tamaño del contenedor de  $100 \times 100$

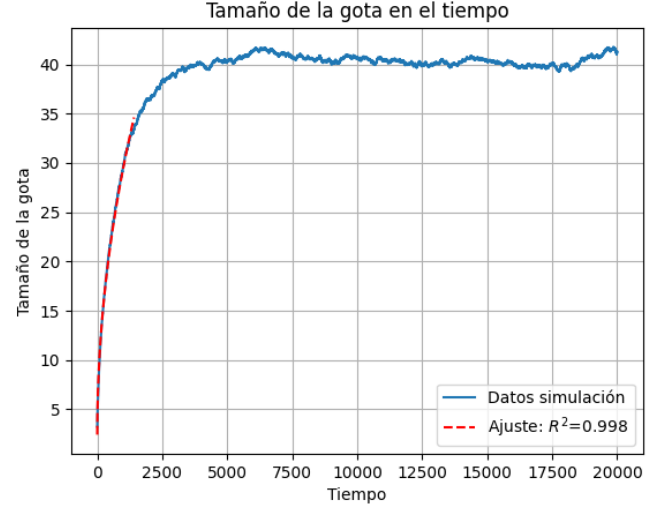


Figura 2: Tamaño del conjunto de partículas en función del tiempo para 1000 partículas en la simulación y un tamaño del contenedor de  $100 \times 100$

Como ya se mencionó el equilibrio se alcanza cuando las partículas están equitativamente distribuidas en el contenedor. Por lo que midiendo el tamaño del conjunto de partículas con la aproximación:

$$\sqrt{\sum_i^N \frac{r_i^2}{N}} \quad (2)$$

Estas deben alcanzar su tamaño máximo para el tiempo en donde se alcanza el equilibrio. Además se tiene que el crecimiento de estas va con  $\sqrt{t}$ , por lo que haciendo un ajuste del tipo  $A\sqrt{t} + b$  se obtuvo un ajuste con  $R^2 = 0.998$ . Para obtener el tiempo de equilibrio se buscó el tiempo en donde la entropía  $S = 4$ , dado que en momentos posteriores cercanos al equilibrio el comportamiento deja de ser el mencionado. Estos resultados se pueden observar en la figura 2:

Finalmente, también se tiene que el tiempo en donde la gota alcanza el equilibrio termodinámico es proporcional al área del contenedor. Por lo que ejecutando la simulación para diferentes tamaños de contenedores y realizando un ajuste lineal se obtuvieron los resultados esperados:

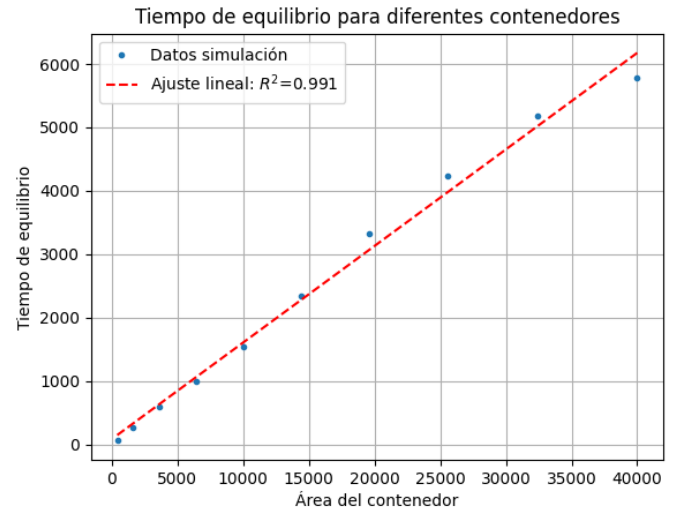


Figura 3: Tiempo de equilibrio en función del área del contenedor para 1000 partículas en la simulación

Por otro lado, se realizó una simulación suponiendo un agujero de longitud 10 en el borde inferior del contenedor dado que

se espera que el número de partículas dentro del contenedor decrezca de manera exponencial cómo  $Ae^{-\frac{t}{\tau}}$  donde  $\tau$  es el tiempo característico de escape. Se verificó el resultado esperado encontrando un  $\tau = 14754$  para 1000 partículas iniciales y un contenedor de  $100 \times 100$ :

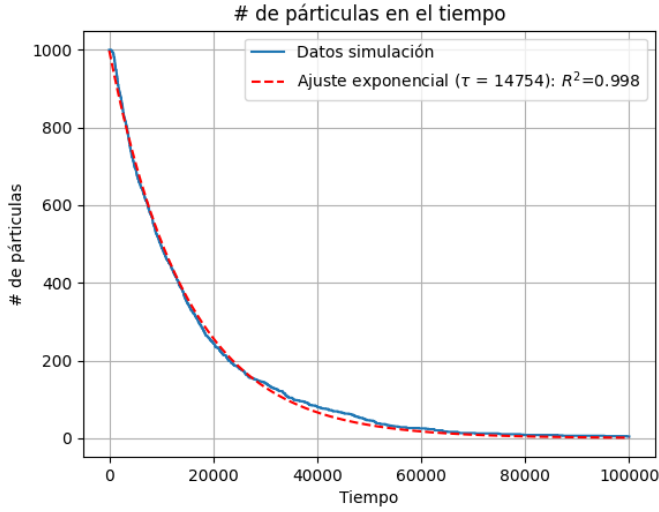
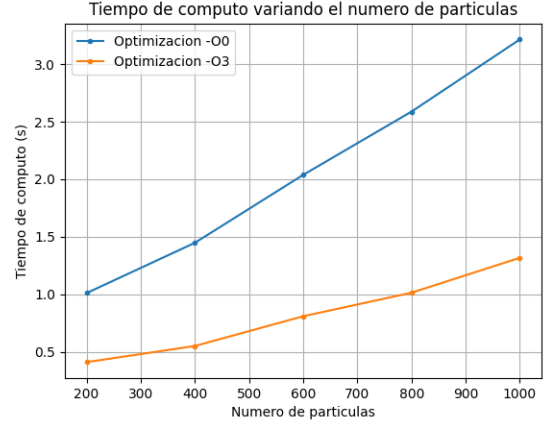


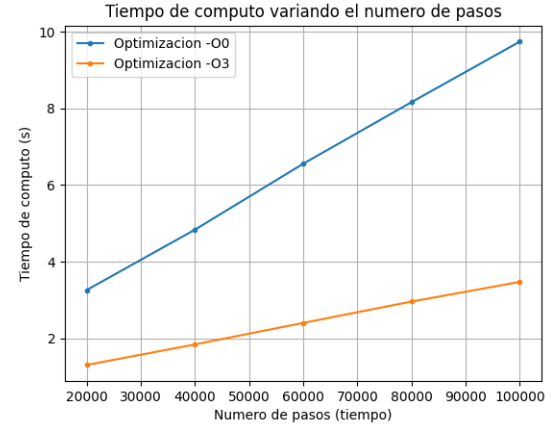
Figura 4: Número de partículas dentro del contenedor ( $100 \times 100$ ) en función del tiempo

## II. Optimización:

Para evaluar el efecto de las optimizaciones al compilar con diferentes banderas, se realizó una duplicación del programa original. Este nuevo programa es capaz de recibir el número de partículas y la cantidad de pasos de tiempo como argumentos desde la consola, dado que estos son los parámetros que tienen el mayor impacto en el tiempo de ejecución. De esta manera, se procedió a compilar el programa variando estos argumentos, y los resultados obtenidos se presentan en la figura 5.



(a) Con 20000 pasos de tiempo



(b) Con 1000 partículas

Figura 5: Tiempo de cómputo en función del tamaño del sistema

Como se puede apreciar, la compilación con la bandera -O3 tiene un impacto significativo, especialmente a medida que el tamaño del sistema aumenta. Además, se observa que al variar el número de pasos de tiempo, el tiempo de ejecución también aumenta. Esto se debe a que la escala del número de pasos requeridos para observar los resultados esperados es mayor que el número necesario de partículas. En las siguientes gráficas se muestran los resultados mencionados en la primera sección, pero con la ejecución del programa utilizando 400 partículas. A pesar de esto, los resultados obtenidos continúan siendo consistentes con las predicciones realizadas.

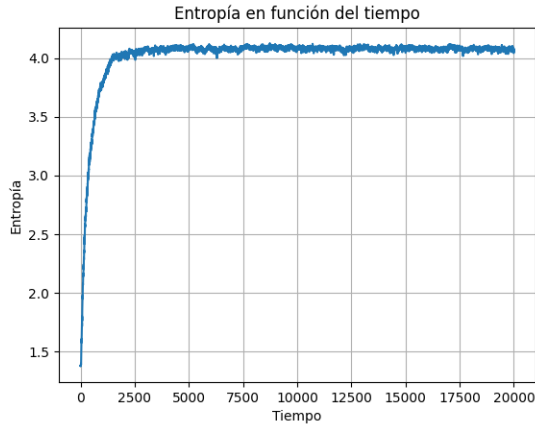


Figura 6: Entropía en función del tiempo para 400 partículas en la simulación y un tamaño del contenedor de  $100 \times 100$

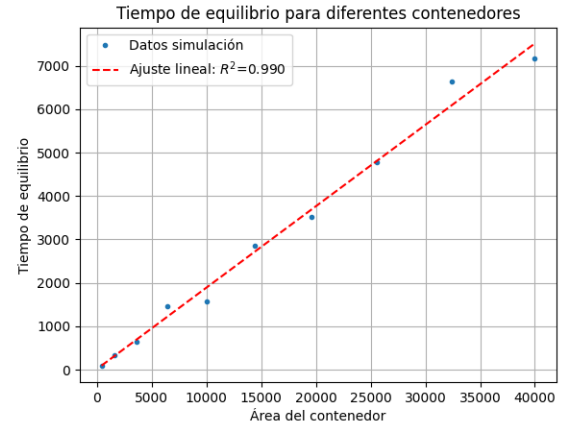


Figura 8: Tiempo de equilibrio en función del área del contenedor para 400 partículas en la simulación

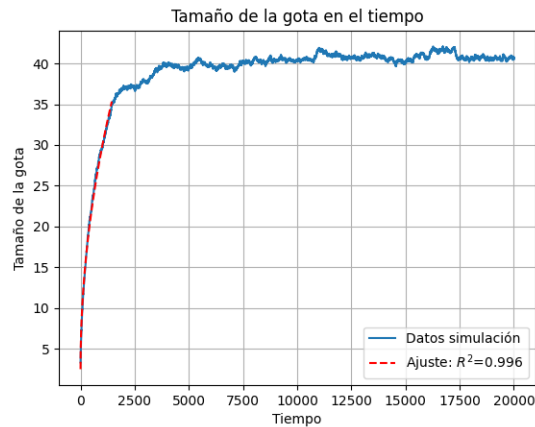


Figura 7: Tamaño del conjunto de partículas en función del tiempo para 400 partículas en la simulación y un tamaño del contenedor de  $100 \times 100$

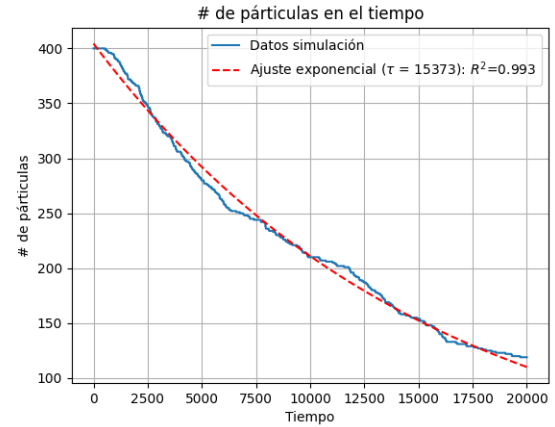


Figura 9: Número de partículas dentro del contenedor ( $100 \times 100$ ) en función del tiempo

Por este motivo, se hizo un esfuerzo por reducir la cantidad de bucles de tiempo necesarios en el código. Esto se logró al combinar la simulación estándar con la simulación que incluye un agujero en el borde inferior dentro del mismo bucle. No obstante, fue necesario llevar a cabo otra iteración sobre el tiempo para determinar el tiempo de equilibrio para diferentes tamaños del contenedor. sin embargo en cada iteración, una vez obtenido el tiempo de equilibrio, se detuvo el proceso. Sin haber realizado esta optimización el tiempo de computo aumentaba en casi un factor de 6 al aumentar los pasos como se observa en la figura 11:

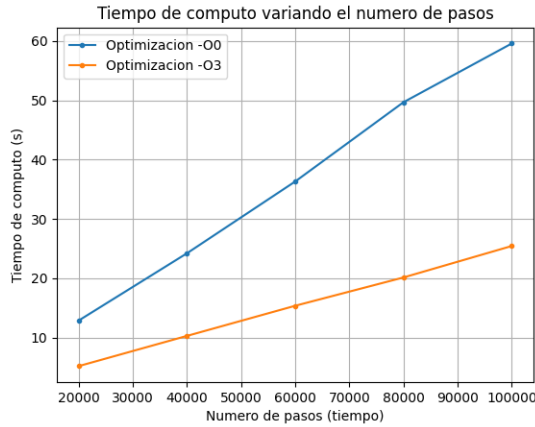


Figura 10: Tiempo de computo sin detener la segunda iteración de tiempo

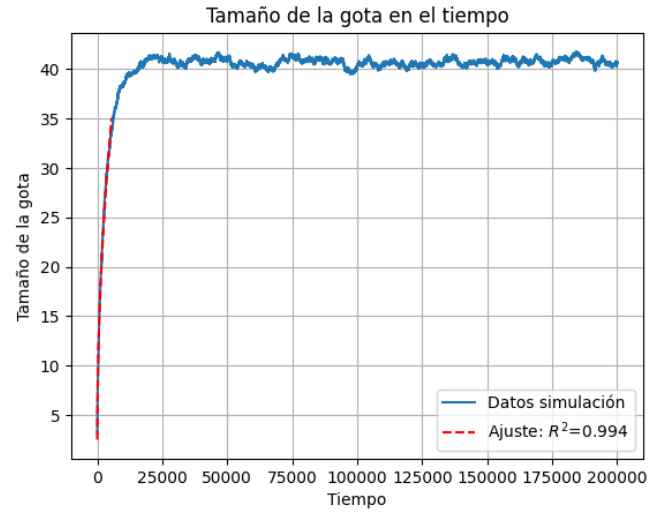


Figura 11: Tamaño del conjunto de partículas en función del tiempo para 1000 partículas en la simulación, un tamaño del contenedor de  $100 \times 100$ , y el paso del caminante igual a 0.5

Finalmente, se implementaron optimizaciones evidentes, como el uso de matrices unidimensionales para almacenar el número de partículas en las divisiones del contenedor y la probabilidad de cada división. Además, desde el principio se priorizó la utilización de vectores para almacenar información de manera contigua en la memoria, lo que contribuyó a mejorar la eficiencia del programa.

En línea con estas optimizaciones, otra mejora implementada fue fijar el tamaño del paso del caminante aleatorio en una unidad. Esta medida se adoptó debido a que permite reproducir con precisión las predicciones. Si bien reducir el tamaño del paso del caminante también logra este objetivo, aumenta la cantidad de pasos en el tiempo necesarios para alcanzar resultados comparables. Por ejemplo, si el paso del caminante se fija en 0.5, el tiempo en toda la simulación se escala en un aproximadamente factor de 5. Esto es evidente al estudiar el tamaño de la gota en función del tiempo, donde con un paso de una unidad alcanzaba su tamaño máximo alrededor de los 2500 pasos de tiempo, mientras que con esta modificación se alcanza en los 12000 pasos: