

Simulación de Sistemas

Trabajo Práctico Nro. 1: Búsqueda Eficiente de Partículas Vecinas

Sea un área cuadrada de lado L que contiene N partículas con radios (r) distintos de cero y con un radio de interacción (r_c).

1- Implementar el algoritmo "Cell Index Method" que tome como input: las posiciones y radios de N partículas y los parámetros N , L , M y r_c (ver punto 5). Las N partículas deben ser generadas en forma aleatoria dentro del área de lado L .

Y como output:

- Una lista tal que para cada partícula indique cuales son las vecinas que distan menos de r_c .
- El tiempo de ejecución.
- Además se debe generar una figura que muestre las posiciones de todas las partículas, y que identifique una de ellas (pasada como input) de un color y sus vecinos correspondientes de otro color.

Las distancias entre partículas deben medirse borde a borde, es decir, considerando el radio (r) además del centro de masa de las mismas. Cómo se modifica el criterio $L/M > r_c$ cuando la partícula no es puntual, es decir tiene un radio ($r > 0$) ? (podría suceder que el centro esté en una celda no vecina pero el borde sí esté en la vecina...?).

Como parámetro adicional considerar dos versiones del algoritmo:

- a- Sin condiciones periódicas de contorno (considerando distancia a los bordes del área: paredes).
- b- Con condiciones periódicas de contorno.

2- Para distintos inputs estudiar la eficiencia del algoritmo (medida en tiempo de cálculo) en función de N y del número de celdas ($M \times M$). Comparar con el método de fuerza bruta que mide las distancias entre todos los pares posibles de partículas. Considerar $L=20$, $r_c=1$ y $r=0.25$.

3- Hallar un criterio para determinar el óptimo número de celdas ($M \times M$) teniendo en cuenta la densidad (N/L^2) de un sistema. Considerar $L=20$, $r_c=1$ y $r=0.25$.

4- Demostración en vivo.

Se realizarán en clase, a medida que los grupos vayan finalizando el T.P. dentro de las fechas estipuladas (ver 7).

Se deberán generar nuevas partículas y grilla según los parámetros (N , L , M y r_c) para verificar el funcionamiento del algoritmo usando el criterio hallado en 3. También se variará M para verificar que el hallado automáticamente sea el óptimo. Para la demostración se usarán los outputs descriptos en el punto 1.

5. Formato tentativo de los archivos:

- Input:

En general para una simulación, el sistema se puede describir con 2 archivos de texto: el estático y el dinámico (consideraremos a estos archivos como el Input para el CIM).

Estático:

- N (Heading con el Nro. total de Partículas)
- L (Longitud del lado del área de simulación)
- r_1 pr_1 (radio y propiedad de la partícula 1)
- r_2 pr_2 (radio y propiedad de la partícula 2)

....
 r_N pr_N (radio y propiedad de la partícula N)

Dinámico:

t_0 (tiempo)
 x_1 y_1 vx_1 vy_1 (partícula 1)
 x_2 y_2 vx_2 vy_2 (partícula 2)
....
 x_N y_N vx_N vy_N (partícula N)
 t_1 (tiempo)
 x_1 y_1 vx_1 vy_1 (partícula 1)
 x_2 y_2 vx_2 vy_2 (partícula 2)
....
 x_N y_N vx_N vy_N (partícula N)

Otra forma de imprimir archivos dinámicos, puede ser generando un archivo por cada tiempo, el cual deberá ser nombrado con las cifras numéricas del tiempo correspondiente (por ejemplo: 1.txt; 5.txt; 10.txt; 15.txt;, si se guardan datos cada 5 unidades de tiempo).

A los fines del presente trabajo se considera un único tiempo (t_0) ya que el método de detección de vecinos se aplica en un determinado estado del sistema en un dado instante.

- Output:

[id de la partícula "i" id's de las partículas cuya distancia borde-borde es menos de r_c].

...

6- Para visualizar las partículas coloreadas se recomienda usar alguna herramienta existente que puede ser independiente del código implementado, como por ejemplo: Ovito (www.ovito.org), Matlab, Octave, Origin, Python, etc.

7- Fecha de Entrega:

Las fechas para la demostración en vivo descrita en el punto 4 se realizará durante las clases de los días 10 y 14 de Marzo 2025.