# **Apuntes – Módulo 6**

## **Curso: Modeling and Simulation of Natural Processes**

**Universidad de Ginebra - Prof. Bastien Chopard et al.** **Semana 6 – Partículas y Objetos Puntuales**

## **1. Introducción y conceptos generales**

### **Características principales de los sistemas estudiados:**

* Gran cantidad de objetos: Simulamos sistemas con millones o miles de millones de partículas
* Mecánica Newtoniana: Todos los modelos se basan en las leyes clásicas de Newton
* Objetos puntuales: Simplificamos objetos complejos a puntos con propiedades básicas (posición, velocidad, masa)
* Interacciones por pares: Las fuerzas se calculan entre pares de objetos y luego se suman

### **Ejemplos representativos:**

**1. Problema de n-cuerpos (galaxias)**

* Modelamos estrellas como puntos que interactúan gravitacionalmente
* Ejemplo: colisión entre galaxias (incluyendo nuestra Vía Láctea con Andrómeda en ~2 mil millones de años)
* Nota interesante: Durante estas "colisiones", las estrellas no chocan físicamente debido al enorme espacio vacío entre ellas

**2. Moléculas en un gas**

* Partículas que siguen trayectorias rectas la mayor parte del tiempo
* Interacción solo cuando están muy cerca (colisiones)
* Cambio de dirección y velocidad durante las colisiones

**3. Glóbulos rojos en arterias**

* Simplificación extrema: ~1 mil millones de células modeladas como puntos
* Interacción partícula-partícula + acoplamiento partícula-fluido
* Aplicación práctica: modelado de coagulación sanguínea

## **2. Leyes de Newton y fuerzas**

### **Las tres leyes fundamentales:**

**Primera ley (Inercia):**

* Un objeto en reposo permanece en reposo
* Un objeto en movimiento continúa con velocidad constante
* A menos que actúe una fuerza externa

**Segunda ley (Fundamental):**

* **F = ma**
* La suma vectorial de fuerzas es igual a masa × aceleración
* Esta es la ecuación que integramos numéricamente

**Tercera ley (Acción-Reacción):**

* Toda acción tiene una reacción igual y opuesta
* Importante para conservar el momento del sistema

### **Consideraciones importantes:**

* Nos enfocamos en fuerzas de interacción entre objetos
* Aproximamos efectos no-clásicos (como mecánica cuántica) con fuerzas newtonianas
* Descomponemos interacciones complejas en sumas de interacciones por pares

## **3. Tipos de potenciales y fuerzas**

### **3.1 Fuerza gravitacional**

F\_ij = G \* m\_i \* m\_j \* (x\_j - x\_i) / |x\_j - x\_i|³

**Características de esta fuerza:**

* Fuerza de largo alcance(decrece como r⁻²)
* Siempre atractiva
* Actúa a grandes distancias
* Potencial: V = -G*m\_i*m\_j/r

### **3.2 Ley de Coulomb (Electrostática)**

F\_ij = (q\_i \* q\_j)/(4πε₀) \* (x\_j - x\_i) / |x\_j - x\_i|³

**Características de la ley:**

* Misma forma matemática que la gravedad
* Puede ser atractiva (cargas opuestas) o repulsiva (cargas iguales)
* Fuerza de largo alcance (r⁻²)
* Constante diferente: 1/(4πε₀) vs G

### **3.3 Potencial de Lennard-Jones**

V\_ij = 4ε[(σ/r)¹² - (σ/r)⁶]

**Componentes:**

* **Término r⁻¹²**: Repulsión de Pauli (evita superposición)
* **Término r⁻⁶**: Atracción de Van der Waals (fuerza atractiva)
* **Fuerza de corto alcance**: Decrece rápidamente con la distancia

**Parámetros:**

* **ε**: Profundidad del pozo de potencial
* **σ**: Tamaño efectivo del átomo/molécula

**Notas técnicas:**

* El exponente 12 se eligió por eficiencia computacional (es el cuadrado del término r⁻⁶)
* A distancia 2.5σ, el potencial es solo ~1% de ε

## **4. Comparación de Fuerzas**

| **Tipo** | **Comportamiento** | **Alcance** | **Aplicación** |
| --- | --- | --- | --- |
| Gravitacional | r⁻² | Largo | Astronomía, n-cuerpos |
| Coulomb | r⁻² | Largo | Partículas cargadas |
| Lennard-Jones | r⁻⁶ | Corto | Dinámica molecular |

**Implicaciones prácticas:**

* Fuerzas de largo alcance: Todas las partículas sienten todas las demás
* Fuerzas de corto alcance: Solo partículas cercanas interactúan significativamente

## **5. Integración temporal - algoritmo de Verlet**

### **Principios generales:**

* Discretizamos el tiempo en pasos δt
* Estado del sistema definido por posiciones x\_i y velocidades v\_i
* El algoritmo de Verlet es eficiente y preciso

### **5.1 Versión Original de Verlet**

**Concepto:**

* No almacena velocidades explícitamente
* Usa posiciones en dos tiempos consecutivos: t-δt y t

**Ecuación clave:**

a\_i(t) ≈ [x\_i(t+δt) - 2x\_i(t) + x\_i(t-δt)] / δt²

**Algoritmo:**

x\_i(t+δt) = δt²·a\_i(t) + 2x\_i(t) - x\_i(t-δt)

### **5.2 Algoritmo "Leap-Frog"**

**Ventaja:** Almacena velocidades explícitamente

**Estructura temporal:**

* Posiciones definidas en t, t+δt, t+2δt, ...
* Velocidades definidas en t+δt/2, t+3δt/2, ...

**Ecuaciones:**

v\_i(t+δt/2) ≈ [x\_i(t+δt) - x\_i(t)] / δt

x\_i(t+δt) = x\_i(t) + δt·v\_i(t+δt/2)

v\_i(t+3δt/2) = v\_i(t+δt/2) + δt·a\_i(t+δt)

**Notas personales:**

* El leap-frog es más intuitivo porque maneja velocidades explícitas
* Ambos métodos tienen precisión de segundo orden
* La elección depende de si necesitamos velocidades en cada paso

## **6. Complejidad computacional y optimizaciones**

### **6.1 Problema de complejidad**

**Cálculo de fuerzas:**

* N objetos requieren N cálculos de fuerza
* Cada fuerza requiere considerar (N-1) otros objetos
* Complejidad total: O(N²)

**Implicaciones prácticas:**

* Con 1 millón de partículas: ~10¹² operaciones por paso de tiempo
* Esto limita sistemas a ~1 millón de objetos en supercomputadoras

### **6.2 Optimización para Lennard-Jones: Distancia de corte**

**Concepto:**

* A distancia d\_c = 2.5σ, el potencial es despreciable
* Ignoramos partículas más allá de esta distancia

**Problema:**

* Buscar partículas cercanas podría seguir requiriendo O(N²) comparaciones

Solución: Método de Grilla

### **6.3 Método de grilla**

**Implementación:**

1. Dividir el espacio en celdas de tamaño > d\_c
2. Cada celda almacena IDs de partículas contenidas
3. Para una partícula, solo revisar 9 celdas (2D) o 27 celdas (3D)

**Ventajas:**

* Actualizar grilla: O(N)
* Buscar vecinos: O(1) promedio por partícula
* Complejidad total: O(N) si N\_celdas ~ N

**Notas prácticas:**

* La grilla debe actualizarse cuando las partículas se mueven
* El tamaño de celda debe ser mayor que la distancia de corte
* Funciona bien cuando la distancia de corte es pequeña comparada con el dominio

**7. Métodos de Partículas y N-Cuerpos**

¡Uf, este módulo ha sido intenso! Pasamos de sistemas continuos a la interacción entre un montón de partículas. La clave parece ser la eficiencia, porque con tantos elementos, los cálculos pueden salirse de control muy rápido.

### **1. Integración Temporal de Ecuaciones de Movimiento**

**Apuntes clave:**

* **Sistemas de muchas partículas:** Aquí, el estado del sistema se define por la posición y velocidad de *cada* objeto. ¡Imaginemos la cantidad de datos!
* **Segunda Ley de Newton:** La base de todo. La aceleración depende de las fuerzas, y las fuerzas dependen de las posiciones relativas de las partículas.
* **Discretización del tiempo:** Como ya vimos en módulos anteriores, dividimos el tiempo en pasos Δt. El objetivo es ir del tiempo t a t+Δt, de forma iterativa.
* **Algoritmo de Verlet (o "leapfrog"):** Este fue el método que se presentó como el *go-to* para este tipo de problemas.
  + **¿Cómo funciona?** Actualiza la posición en un momento t+Δt usando la velocidad en t+Δt/2. Luego, actualiza la velocidad en t+3Δt/2 usando la aceleración en t+Δt. Es como "saltar" entre las posiciones y velocidades en diferentes puntos intermedios del tiempo.
  + **Ventajas:** Es eficiente y preciso. Esto es CRUCIAL cuando se manejan muchas partículas.

**Notas y consideraciones generales:**

En el algoritmo de Verlet la idea de "leapfrogging" (salto de rana) lo hace mucho más intuitivo. Parece que la clave de su eficiencia es que evita calcular la velocidad y la posición en el mismo instante, lo que simplifica los cálculos. Queda la cuestión de si habrá situaciones donde este método no sería el ideal, o si hay otros aún más eficientes para tipos de fuerzas muy específicas.

### **2. El Potencial de Lennard-Jones: Introducción de una distancia de corte**

**Apuntes clave:**

* **Interacciones entre N partículas:** Si cada partícula interactúa con todas las demás, el cálculo de fuerzas es de orden N2 (por cada una de las N partículas, miramos otras N-1). ¡Esto escala fatal!
* **Potencial de Lennard-Jones:** Un ejemplo de potencial de interacción. Se usa para modelar interacciones entre átomos o moléculas.
* **El "cuello de botella" computacional:** Calcular las fuerzas es la parte más costosa de la simulación.
* **Distancia de corte (rc​):** ¡La solución ingeniosa! Si la interacción disminuye muy rápidamente con la distancia (como en el potencial de Lennard-Jones), podemos asumir que las partículas más allá de cierta distancia ya no interactúan significativamente.
* **Celdas de la cuadrícula:** Para implementar la distancia de corte de forma eficiente, se divide el espacio en celdas. Solo necesitamos comprobar las partículas en la celda actual y sus celdas vecinas.
* **Complejidad del algoritmo:** Al usar una distancia de corte y una cuadrícula, la complejidad se reduce a orden N. ¡De N2 a N es una mejora GIGANTESCA! Si Δx (tamaño de celda) es mayor que rc​, solo se necesitan verificar 9 celdas en 2D (la propia celda y 8 vecinas) o 27 en 3D.

**Notas y consideraciones generales:**

La idea de la distancia de corte me parece que es una forma de aprovechar la física del problema (la rápida caída del potencial) para optimizar la computación. Me hace pensar en otros sistemas donde se podría aplicar un principio similar. Es fundamental que la longitud de corte sea mucho menor que el tamaño total del dominio para que esta optimización realmente valga la pena.

### **3. El Problema de los N-Cuerpos: Evaluación de las fuerzas gravitatorias**

**Apuntes clave:**

* Fuerzas de largo alcance: A diferencia del Lennard-Jones, en la gravedad *todas* las partículas importan. No podemos usar una distancia de corte.
* Comportamiento emergente: La rotación de las galaxias alrededor de un centro de masa común es un resultado de las interacciones individuales por pares.
* Algoritmos de árbol (Barnes-Hut): Una aproximación para reducir la complejidad en problemas de N-cuerpos con fuerzas de largo alcance.
  + Subdivisión del espacio: Se divide el espacio en "árboles cuádruples" (quadtrees en 2D) u "octárboles" (octrees en 3D).
  + Nodos interiores y hojas: Las hojas del árbol son los cuerpos individuales. Los nodos interiores representan grupos de cuerpos.
  + Centro de masa y masa total: Para los nodos interiores, se calcula un centro de masa y una masa total.
  + Simplificación de interacciones: Si un grupo de cuerpos (representado por un nodo interior) está *lo suficientemente lejos* de la partícula sobre la que estamos calculando la fuerza, podemos tratar todo ese grupo como una sola partícula puntual ubicada en su centro de masa. Esto reduce drásticamente el número de cálculos.
  + Criterio de proximidad (parámetro θ): Define "lo suficientemente lejos". Si un cuadrante está lo suficientemente lejos, se usa su centro de masa. Si no, se subdivide y se examinan sus hijos.

**Consideraciones generales del módulo:**

1. **Simplicidad vs. Realidad**: Modelos simples pueden capturar comportamientos complejos
2. **Escalabilidad**: La complejidad computacional determina el tamaño máximo del sistema
3. **Aproximaciones físicas**: Necesarias para hacer factibles las simulaciones
4. **Importancia del algoritmo**: La elección del método numérico afecta precisión y eficiencia

La forma en que se estructuran los datos para optimizar los cálculos es súper inteligente. El algoritmo de Barnes-Hut es un ejemplo perfecto de cómo un buen diseño de datos puede transformar un problema intratable en uno manejable. La idea de que puedes representar una galaxia entera como un solo punto para efectos gravitacionales a larga distancia es alucinante.