tiny_SPH: Computación Paralela 2025 - FaMAF, UNC

Bruno M. Celiz[1], & Daniela Couriel[1]

[1] *Facultad de Matemática, Astronomía, Física y Computación, UNC, Argentina*

Contacto / bruno.celiz@mi.unc.edu.ar - https://github.com/BrunoCeliz/tiny_SPH

Resumen / En el marco del curso de posgrado "Computación Paralela 2025" de la Facultad de Matemática, Física, Astronomía y Computación de la Universidad Nacional de Córdoba, presentamos un código flexible que permite resolver las ecuaciones de la hidrodinámica para fluidos a partir de la técnica lagrangiana de partículas suavizadas (SPH por sus siglas en inglés). El código está escrito en `C++` y se basa originalmente en el repositorio de Matthias Varnholt (2016), el cual incluye una visualización en tiempo real del sistema implementando `Qt` y `OpenGL`.

Abstract / Within the postgraduate course "Parallel Computing 2025" of FaMAF - UNC, we present a flexible code able to solve the fluid hydrodynamic equations using the lagrangian scheme of Smoothed Particle Hydrodynamics (SPH). The code is written in C++ and is originally obtained from the GitHub repository of Matthias Varnholt (2016), which includes a Graphical User Interface (GUI) that allows an on-the-fly visualization of the system being evolved through `Qt` and `OpenGL` dependencies.

Keywords / methods: numerical — software and its engineering: software post-development issue

1. Introducción

La Hidrodinámica de Partículas Suavizadas (SPH, por sus siglas en inglés) es una técnica para resolver numéricamente las ecuaciones de la hidrodinámica a partir de un enfoque lagrangiano (Monaghan, 1992; Gnedin et al., 2016). En esta implementación, un fluido se representa como un sistema de partículas sobre las cuales se computan propiedades físicas como la velocidad ($\vec{v}$), densidad (ρ), presión (P), temperatura, etc., a partir de la resolución de las ecuaciones de Euler de continuidad:

$$\frac{d\rho}{dt} + \rho\nabla\cdot\vec{v} = 0 \qquad \text{(Masa)} \quad (1)$$

$$\frac{d\vec{v}}{dt} + \frac{\nabla P}{\rho} = 0 \qquad \text{(Momento)} \quad (2)$$

$$\frac{du}{dt} + \frac{P}{\rho}\nabla\cdot\vec{v} = 0 \qquad \text{(Energia)} \quad (3)$$

donde $d/dt = \partial/\partial t + \vec{v}\cdot\nabla$ es la derivada convectiva y u es la energía interna. Este sistema de ecuaciones diferenciales parciales provienen del Lagrangiano

$$L = \int \rho\left(|\vec{v}|^2/2 - u\right) dV \quad (4)$$

La idea principal de SPH es evitar el uso de mallas o grillas, aprovechando a las partículas como puntos de interpolación (Gingold & Monaghan, 1977; Lucy, 1977). Más aún, en esta aproximación facilita la conservación de e.g. momento lineal y angular, masa y entropía, siendo un esquema completamente invariante bajo transformaciones galileanas. El carácter Lagrangiano permite que la resolución local siga automáticamente a las concentraciones del fluido, favoreciendo situaciones de gran contraste de densidad, como pueden ser sistemas astrofísicos.

Los cimientos de la técnica SPH yace en una función núcleo $W(\vec{r}, h)$ (*"kernel"* de ahora en adelante) de la cuál se utiliza para estimar la densidad, base del formalismo variacional que se busca implementar. Para dado campo físico $F(\vec{r})$ se define su estimación interpolada a partir de la convolución con el kernel:

$$F(\vec{r}) = \int F(\vec{r'})W(\vec{r} - \vec{r'}, h)d\vec{r'} \quad (5)$$

donde h es el tamaño característico del kernel. El kernel debe ser normalizado a la unidad, aproximarse a la función δ de Dirac cuando $h \to 0$, simétrico y suave (al menos 2 veces diferenciable). Por convergencia, se suele recurrir a kernels de soporte compacto i.e., $W(d, h) = 0$ si d es mayor a un límite impuesto (ver e.g. Read et al., 2010).

A partir de una integración Montecarlo es posible discretizar estos campos en los puntos de interpolación (partículas) del fluido en un sistema. Para una aproximación correcta a segundo orden de la verdadera función subyacente, se requieren un mínimo de 32 puntos vecinos[⋆]. Los kernels de soporte compacto permiten que sólo aquellas partículas que se encuentren a un radio $r < 2h$ sean consideradas para el cómputo de propiedades de una partícula, restringiendo el cómputo computacional a $\mathcal{O}(N_{\text{vecinos}}N)$ en vez de la complejidad $\mathcal{O}(N^2)$ de considerar todas el resto de partículas.

La densidad de un fluido en la $i$ésima partícula es entonces obtenida a partir de la siguiente ecuación, considerando sólo sus vecinos (subíndice j):

$$\rho(\vec{r_i}) = \sum_j m_j W(\vec{r_i} - \vec{r_j}, h) \quad (6)$$

[⋆]Si los puntos están representados en un espacio Cartesiano.

la cual se utiliza para obtener el resto de propiedades, que se calculan de forma similar.

Este proyecto entonces presenta un código base en C++ que permite integrar un sistema de partículas bajo las ecuaciones de la hidrodinámica, aproximando la evolución temporal de un fluido. El mismo proporciona una determinada flexibilidad a la hora de definir las condiciones iniciales, las principales propiedades del fluido (viscocidad, incompresibilidad, ablandamiento y distancia característica entre vecinos). Además, incluye una interface visual (GUI por sus siglas en inglés) para observar el sistema durante la evolución temporal del mismo, a partir de dependencias como `Qt` y `OpenGL`.

2. Implementación

El código está construido principalmente sobre los objetos `SPH` y `Particle`. `SPH` se encarga de contener las constantes y parámetros físicos del sistema, así como la cantidad total de pasos y de partículas. `Particle` es la estructura que representa a una partícula, y transporta la posición, velocidad, aceleración, masa y densidad.

Primeramente, se inicializa la posición y velocidad de las partículas, se fija el contexto del sistema (e.g. fluido contenido adentro de una caja bajo gravedad constante, o libres bajo un potencial estático) y constantes como viscosidad, rigidez y densidad en equilibrio del fluido, y parámetros como la cantidad de vecinos máxima y la distancia dentro de la cual el kernel de interpolación es mayor a 0.

2.1. Búsqueda de vecinos

Para comenzar a evolucionar el sistema, interpolamos las propiedades del fluido en las partículas SPH utilizando partículas vecinas. Como primera aproximación, fijamos la cantidad de vecinos máxima N_{vec} y la distancia máxima de interpolación h.

Para facilitar la búsqueda de vecinos de la $i$ésima partícula, siendo aquellos que se encuentren a una distancia $d_{ij} < h$ recurrimos a una grilla Cartesiana uniforme, que divide al espacio en celdas 3D. La cantidad de celdas N_{cell} y su tamaño l_{cell} se mantienen fijas a lo largo de la simulación. Una vez generada la grilla, se le asigna un índice de celda a cada partícula SPH.

Para cada partícula, los posibles vecinos se encuentran en la misma celda o en celdas contiguas. La búsqueda se hace aleatoriamente, cambiando en cuál de las posibles celdas se consdieran vecinos para cada partícula y en cada intervalo de tiempo.

2.2. Cómputo de la densidad e hidrodinámica

Una vez definidos los vecinos de cada partícula, computamos la densidad de la $i$ésima partícula y la presión en ese punto

$$\rho_i = \sum_j m_j W(\vec{r_i} - \vec{r_j}, h) \tag{7}$$

$$P_i \approx \kappa \cdot (\rho_i - \rho_0) \tag{8}$$

donde j es el subíndice que recorre los vecinos de la misma, ρ_0 es la densidad del fluido en reposo, κ la rigidez del mismo y el kernel $W(r, h)$ utilizado es

$$W(r,h) = \frac{315}{64\pi h^9} \cdot (h^2 - |\vec{r_i} - \vec{r_j}|^2)^3 \tag{9}$$

La densidad ρ_i debe ser calculada para todas las partículas para luego resolver las ecuaciones diferenciales parciales de la hidrodrinámica. La ecuación de movimiento es, entonces,

$$\rho \cdot \left[\frac{d\vec{v}}{dt} + \vec{v} \cdot \nabla\vec{v}\right] = \rho \cdot \vec{g} - \nabla P + \mu\nabla^2\vec{v} \tag{10}$$

donde $\vec{v}$ es la velocidad, $\vec{g}$ es la fuerza gravitatoria y μ es la viscosidad del fluido. Esta ecuación puede ser escrita, para una única partícula, como

$$\frac{D\vec{v_i}}{Dt} = \vec{g} - \frac{\nabla P_i}{\rho_i} + \frac{\mu}{\rho_i}\nabla^2\vec{v_i} \tag{11}$$

con D/Dt la derivada convectiva. Aprovechando el kernel de interpolación para obtener las propiedades del fluido en cada partícula, aproximamos los diferentes términos (conservación de momento y el término de viscosidad) considerando los vecinos de las mismas:

$$\frac{\nabla P_i}{\rho_i} \approx \sum_j m_j(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}) \cdot \nabla W(\vec{r_i} - \vec{r_j}, h) \tag{12}$$

$$\frac{\mu}{\rho_i}\nabla^2\vec{v_i} \approx \frac{\mu}{\rho_i}\sum_j m_j(\frac{\vec{v_j} - \vec{v_i}}{\rho_j}) \cdot \nabla^2 W(\vec{r_i} - \vec{r_j}, h) \tag{13}$$

2.3. Cómputo de la gravedad e integración numérica

El término de la aceleración gravitatoria $\vec{g}$ depende del contexto en el cual se quiera evolucionar el sistema. Por ejemplo, si queremos modelar un fluido sobre la superficie terrestre, el campo gravitatorio será constante, uniforme y apuntará hacia abajo i.e. $|\vec{g}| = -9.8\hat{k}$ m s^{-2}. En cambio, si deseamos modelar un fluido alrededor de un potencial estático $\Phi(\vec{r})$, $\vec{g}$ cambiará según la posición de cada partícula tal que $\vec{g_i} = -\nabla\Phi$. El código no resuelve el problema de N-cuerpos por defecto, pero incluye una dsitancia de ablandamiento ε ("softening length") para evitar divergencias con distancias muy pequeñas.

Una vez obtenida la aceleración de cada partícula $\vec{a_i} = d\vec{v_i}/dt$, sólo resta evolucionarla en el tiempo a través de un integrador numérico. En este caso, se emplea un esquema LeapFrog Kick-Drift-Kick (Yoshida, 1990), requiriendo el cálculo de un paso intermedio. Para cada coordenada, las nuevas posiciones y velocidades ($\vec{x}_1$ y $\vec{v}_1$, respectivamente) se obtienen a partir de los valores de las mismas en el paso actual ($\vec{x}_0$ y $\vec{v}_0$) de acuerdo a

$$v_{1/2} = v_0 + \frac{a_0}{2} \cdot \Delta t \tag{14}$$

$$x_1 = x_0 + v_{1/2} \cdot \Delta t \tag{15}$$

$$v_1 = v_{1/2} + \frac{a_1}{2} \cdot \Delta t \tag{16}$$

donde $v_{1/2}$ es la velocidad de la partícula en un intervalo de tiempo intermedio. La presencia de a_1 implica que debemos volver a calcular la aceleración del sistema. Esto puede solventarse guardando la información de la aceleración anterior o sólo re-computando la aceleración

gravitatoria (y olvidarse del cómputo de la hidrodinámica en el paso intermedio).

Previamente mencionamos que la aproximación SPH permite un buen comportamiento en la conservación de cantidades físicas gracias a su formalismo variacional. El esquema de integración elegido, LF-KDK, también favorece la conservación de energía del sistema al ser una transformación simpléctica del Hamiltoniano (Tuckerman, 2010).

3. Parámetros iniciales

A continuación se listan los valores de parámetros físicos y constantes que se emplean por defecto, en unidades de convención astronómica (e.g. [pc; km/s; Myr; $M_\odot$])

- $h = 0.1$ (tamaño del kernel)
- $N_{particles} = 16 \times 1024$ (cantidad de partículas del sistema)
- $N_{vec,max} = 32$ (cantidad máxima de vecinos)
- $N_{cell} = 32$ (numero de celdas en cada eje)
- $l_{cell} = 2h$ (tamaño (lado) de cada celda)
- $T_{tot} = 10^0$ (tiempo de evolución del sistema)
- $\Delta t = 10^{-4}$ (intervalos de tiempo, fijos e iguales para todas las partículas)
- $\rho_0 = 100.0$ (densidad de equilibrio del fluido)
- $\varepsilon = h/2$ (parámetro de ablandamiento de la fuerza gravitacional)
- $m_i = 1.0$ (masa de las partículas, fija e igual para todas)
- $\kappa = 0.1$ (rigidez del fluido)
- $\mu = 100.0$ (viscosidad del fluido)
- $G = 4.3009 \times 10^{-3}$ (constante universal de gravitación en [pc km^2 s^{-2} $M_\odot$])

Referencias

Gingold R.A., Monaghan J.J., 1977, MNRAS, 181, 375

Gnedin N.Y., et al., 2016, Star Formation in Galaxy Evolution: Connecting Numerical Models to Reality: Saas-Fee Advanced Course 43. Swiss Society for Astrophysics and Astronomy

Lucy L.B., 1977, AJ, 82, 1013

Monaghan J.J., 1992, ARA&A, 30, 543

Read J.I., Hayfield T., Agertz O., 2010, MNRAS, 405, 1513

Tuckerman M., 2010, *Statistical Mechanics: Theory and Molecular Simulation*, OUP Oxford

Yoshida H., 1990, Physics Letters A, 150, 262