Simulazione SPH (Smoothed Particle Hydrodynamics) 1D

Claudia Murro

March 7, 2025

1 Introduzione

Il codice presentato implementa una simulazione numerica basata sul metodo Smoothed Particle Hydrodynamics (SPH) in un dominio unidimensionale, con condizioni periodiche. In particolare, il codice è progettato per simulare il comportamento di un tubo di shock, dove le particelle interagiscono tra loro in un fluido che evolve nel tempo secondo le leggi della dinamica dei fluidi. Il codice utilizza il metodo di integrazione KDK per aggiornare le posizioni e velocità delle particelle e un kernel cubic spline per calcolare le forze di interazione.

2 Struttura Generale del Codice

Il codice si articola in vari passaggi chiave:

- **Impostazione delle condizioni iniziali del tubo di shock**: Si definiscono due zone, una ad alta pressione e alta densità (zona sinistra) e una a bassa pressione e bassa densità (zona destra), per simulare uno shock che si propaga attraverso il fluido.
- **Ricerca dei vicini per ciascuna particella**: Ogni particella interagisce con un numero definito di vicini. I vicini vengono trovati con due metodi differenti: nel primo si usa un qsort() e nel secondo una griglia.

- **Calcolo della densità e della pressione**: La densità di ciascuna particella viene calcolata utilizzando la somma dei contributi delle particelle vicine, la pressione viene calcolata utilizzando l'equazione di stato del fluido.
- **Calcolo dell'accelerazione e della viscosità**: L'accelerazione di ciascuna particella viene calcolata in base alle forze di pressione e viscosità che derivano dalle interazioni con i vicini.
- **Integrazione temporale con il metodo KDK**: Il passo temporale viene calcolato in modo che la simulazione rispetti la stabilità numerica, le posizioni e velocità delle particelle vengono aggiornate seguendo il metodo KDK.
- **Scrittura dei dati**: I dati relativi alla posizione, velocità, densità, pressione e vicini trovati vengono scritti su file a intervalli di tempo regolari, per permettere l'analisi post-simulazione.

3 Costanti e Parametri Predefiniti con `#define`

Nel codice sono definiti vari parametri e costanti utilizzando la direttiva `#define`.

- `LZ_DENSITY`: Densità iniziale nella zona sinistra (LZ), fissata a 1.0 kg/m^3 (in code unit).
- `RZ_DENSITY`: Densità iniziale nella zona destra (RZ), fissata a 0.125 kg/m^3 (in code unit).
- `LZ_PRESSURE`: Pressione iniziale nella zona sinistra (LZ), fissata a 1.0 (in code unit).
- `RZ_PRESSURE`: Pressione iniziale nella zona destra (RZ), fissata a 0.1 (in code unit).
- `LZ_VELOCITY,RZ_VELOCITY`: Velocità iniziale nella zona sinistra e destra (LZ e RZ) fissata a 0.0 m/s, quindi nulla in entrambe le zone.
- `TUBE_LENGTH`: Lunghezza totale del tubo, fissata a 1.0 m. Questa costante definisce la lunghezza del dominio fisico del tubo.

- `PRESSURE_CONVERSION`: Fattore di conversione della pressione in Pascal, fissato a 100000.0. Questo valore serve per la conversione della pressione dalle unità normalizzate a Pascal.
- `ENERGY_CONVERSION`: Fattore di conversione dell'energia in Joule per chilogrammo, fissato a 100000.0.

Inoltre, sono definiti i seguenti parametri relativi al numero di particelle e alla massa delle particelle, che sono cruciali per la simulazione:

- `N`: Il numero totale di particelle è fissato a 3600. Questo valore è scelto per garantire una discretizzazione adeguata del tubo, bilanciando la risoluzione della simulazione con le risorse computazionali disponibili. Il numero di particelle influisce direttamente sulla precisione delle interazioni tra particelle e sulla qualità dei risultati.
- `MASS`: La massa di ciascuna particella è fissata a 0.00015625 kg. Questo valore è stato calcolato in modo che, distribuita lungo il tubo, la massa delle particelle rispetti le densità iniziali nelle zone LZ e RZ. La massa di ogni particella è stata determinata per ciascuna delle due zone in modo da ottenere una densità totale di 1.0 kg/m^3 nella zona LZ e 0.125 kg/m^3 nella zona RZ, tenendo conto che il tubo è diviso tra LZ e RZ e ha un diverso numero di particelle nelle due zone.

L'uso di costanti attraverso la direttiva `#define` garantisce che i valori siano facilmente modificabili e consistenti in tutto il codice, senza necessità di cambiamenti ripetuti nelle varie sezioni.

4 Impostazione delle Condizioni Iniziali

La simulazione parte con l'impostazione delle condizioni iniziali per il tubo di shock. Le particelle sono distribuite lungo un dominio unidimensionale con due zone distinte: una a sinistra con alta densità e alta pressione (LZ) e una a destra con bassa densità e bassa pressione (RZ). Le particelle sono posizionate uniformemente nel dominio, e la velocità iniziale di ciascuna particella è settata a zero.

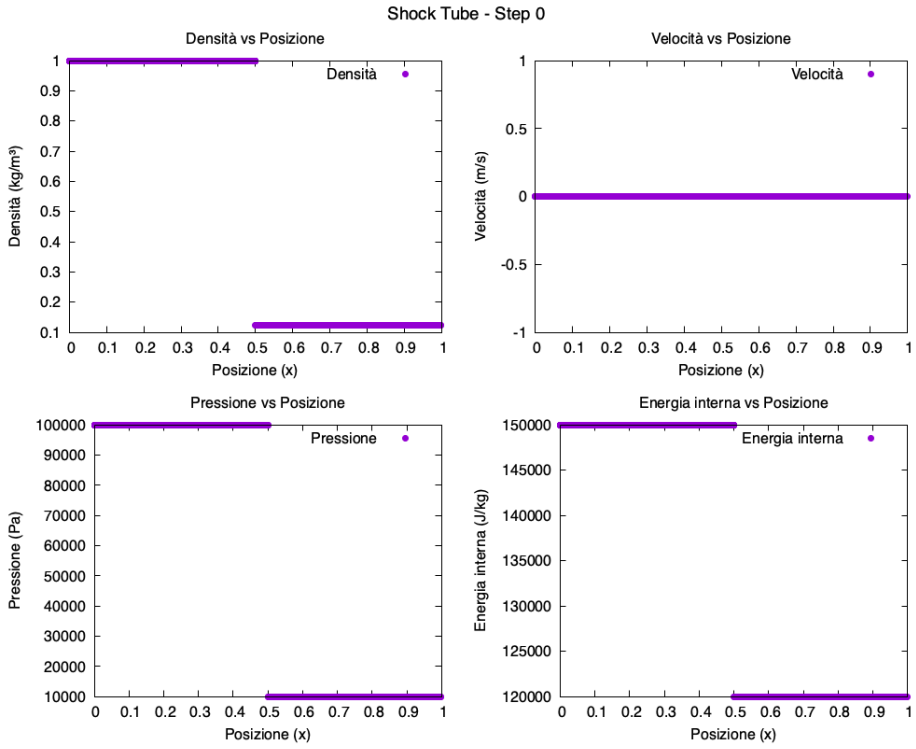

Figure 1: Condizioni iniziali

5 Ricerca dei vicini con qsort()

Per ogni particella, vengono calcolate le distanze tra di essa e tutte le altre particelle nel dominio. Questo calcolo tiene conto delle condizioni periodiche: quando una particella si avvicina al bordo del dominio, la distanza viene "avvolta" per rispettare la periodicità. La distanza tra due particelle viene quindi calcolata come la differenza tra le loro posizioni, con correzioni periodiche se necessario.
Una volta calcolate tutte le distanze, queste vengono ordinate in ordine crescente utilizzando l'algoritmo di ordinamento `qsort`, e vengono selezionati i 64 vicini più prossimi per ciascuna particella. Inoltre, il valore di h (smoothing length) per ogni particella viene aggiornato in modo che la distanza massima tra particelle vicine sia $2h$.

6 Ricerca dei vicini tramite griglia adattiva

La ricerca dei vicini per ogni particella viene effettuata utilizzando una griglia adattiva e un parametro h variabile, che viene calcolato per ogni particella in modo da mantenere un numero costante di vicini. Il processo di ricerca dei vicini è diviso in tre fasi principali:

6.1 Calcolo del lunghezza di smorzamento h per ogni particella

Per ogni particella, viene calcolato un valore iniziale di h, denotato come h_{start}. Il valore iniziale di h_{start} è dato da una formula basata sulla lunghezza del tubo e sul numero di particelle, N. Successivamente, viene eseguito un ciclo iterativo per adattare il valore di h_{start} in modo che l'intervallo $[x-2h, x+2h]$ contenga esattamente un numero prefissato di vicini, MAX_NEIGHBORS. Se il numero di vicini trovati è troppo basso, il valore di h_{start} viene aumentato, altrimenti se i vicini sono troppi, h_{start} viene ridotto. Questo processo di adattamento continua fino a raggiungere il numero desiderato di vicini o fino a un limite massimo di iterazioni.

6.2 Calcolo del valore massimo di h

Una volta che la lunghezza di smorzamento h è stata calcolata per tutte le particelle, il valore massimo di h tra tutte le particelle viene determinato. Questo valore massimo, denotato come H_{MAX}, viene utilizzato per definire la dimensione delle celle della griglia di $2H_{\text{MAX}}$.

6.3 Ricerca dei vicini tramite la griglia

Dopo aver determinato H_{MAX}, viene creata una griglia di celle per la ricerca dei vicini. Ogni particella viene assegnata a una cella della griglia in base alla sua posizione, utilizzando una divisione della lunghezza del tubo per la dimensione delle celle. Successivamente, per ogni particella, vengono cercati i vicini non solo nella cella di appartenenza, ma anche nelle celle adiacenti. Questo permette di garantire che tutte le particelle che si trovano vicino l'una all'altra, anche se si trovano in celle differenti, vengano correttamente identificate come vicini.
Poiché il dominio di simulazione ha condizioni periodiche, la ricerca dei vicini

tiene conto di queste condizioni durante il calcolo delle distanze.
La ricerca dei vicini avviene confrontando la distanza periodica tra ogni coppia di particelle e verificando se essa è inferiore a $2h$, dove h è la lunghezza di smorzamento della particella considerata. Se la distanza è inferiore, la particella viene aggiunta alla lista dei vicini.
Infine, dopo aver trovato tutti i vicini per ogni particella, la memoria utilizzata per la griglia viene liberata, completando così la ricerca dei vicini.
Questa tecnica di ricerca dei vicini utilizza una griglia adattiva per ridurre la complessità computazionale rispetto a un approccio di ricerca a forza bruta, mantenendo al contempo una buona precisione nella determinazione dei vicini.

7 Conclusioni e risultati

Il codice implementato per la simulazione SPH in 1D ha mostrato di essere efficace nel riprodurre il comportamento teorico atteso per il sistema simulato. In particolare, i grafici ottenuti dalle simulazioni, che rappresentano grandezze come la densità, la pressione e l'energia, sono molto simili a quelli previsti dalla teoria per le condizioni iniziali di tubo di shock.
Di seguito è mostrato l'ultimo grafico ottenuto alla fine della simulazione.

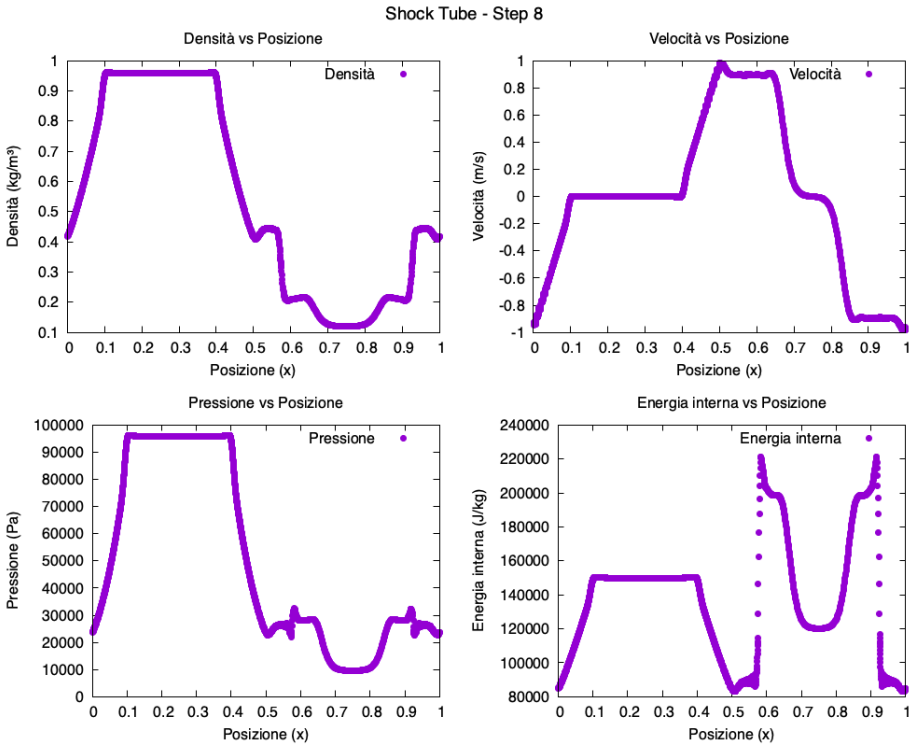

Figure 2: Risultato

8 Organizzazione

Ci sono 2 codici che sono identici eccetto che per la ricrca dei vicini che viene fatta con i due metodi spiegati sopra.
Per ogni codice gli script sono organizzati in 2 cartelle 'estesa' e 'divisa'. Nella cartella "estesa" c'è la prima versione dello script, tutta in unico file. Nella cartella "divisa" c'è l'ultima versione del codice che è stato spezzato in più file che vengono compilati con un Makefile.
In entrambi i casi i comandi per compilazione, esecuzione e graficazione sono specificati. Ogni cartella contiene tutti i file necessari.