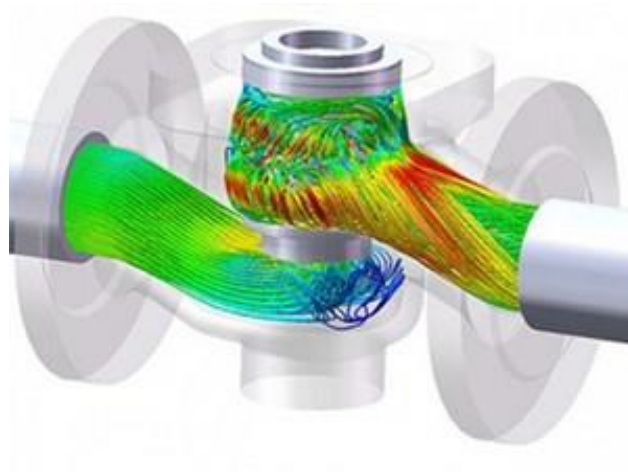


Моделювання фізики рідини за допомогою паралельних обчислень

Ситніков Дмитро Антонович

Навіщо

- Моделювання газів, рідин та м'яких тіл використовується при зйомці фільмів та реклами, розробці комп'ютерних ігор, побудові різних технічних пристроїв (форми аеродинамічних профілів літаків, корпусів спортивних автомобілів, внутрішні структури матеріалів) та в **багатьох** інших галузях науки та виробництва.



Основні властивості рідини

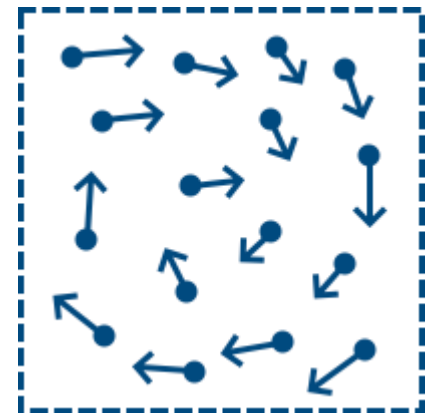
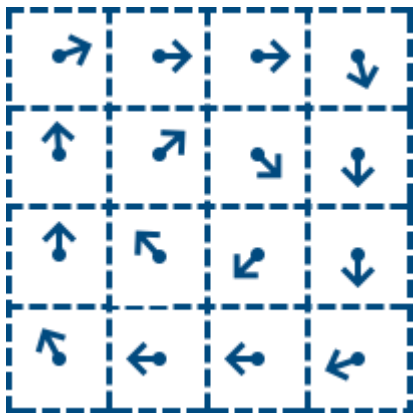
- **Нестисливість:** за умови зміни тиску густина рідини залишається незмінною
- **Здатність до легкої зміни форми:** рідина не є пластичною і легко змінює форму під дією зовнішніх сил.

Визначення рідини

- Рідина займає певний об'єм в просторі, в кожній точці якого визначений стан цієї рідини: швидкість і напрямок руху, густина, температура і т. д.

Види дискретизації об'єму рідини

- Для дискретизації об'єму рідини існують два основні підходи: **метод Ейлера** та **метод Лагранжа**
- В методі Ейлера використовується сітка, в **комірках** якої визначений стан рідини. В методі Лагранжа стан рідини прив'язаний до **частинок**, які рухаються в просторі



Гідродинаміка згладжених частинок (SPH)

- SPH є одним з видів методу Лагранжа
- Рідина представлена множиною частинок.
- Частинки несуть атрибути, що характеризують рідину
- Всередині частинки відсутній рух рідини
- Частинки рухаються і взаємодіють одна з одною

Визначення властивостей рідини

- В загальному вигляді визначення стану рідини в точці А обчислюється за наступною формулою:

$$P_a = \sum_j \frac{m_j}{\rho_j} P_j W_{aj}$$

- P_j - властивість рідини частинки j
- W_{aj} - функція ядра від відстані від точки А до позиції частинки j

Загальний вигляд ітерації алгоритму SPH

- Обчислення густини рідини в позиціях частинок
- Обчислення тиску рідини в позиціях частинок
- Обчислення сил, що діють на частинки і прискорення
- Обчислення нової швидкості та нової позиції частинок

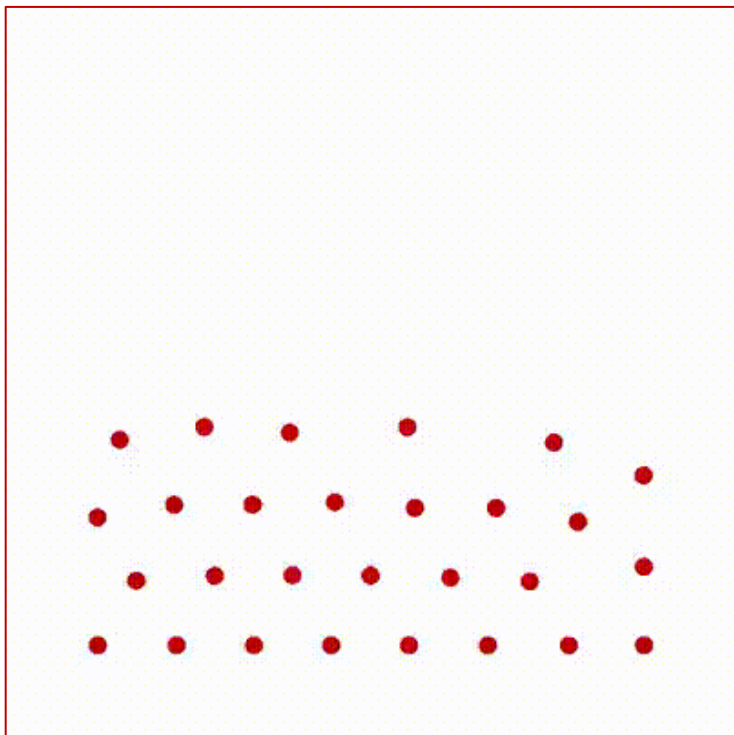
Густина

- Густина в точці рідини обчислюється за формулою

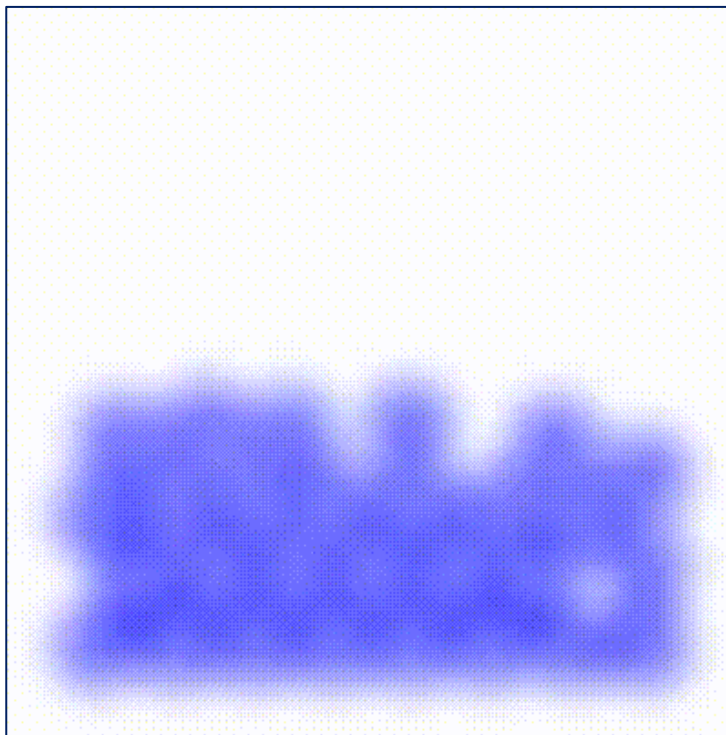
$$\rho = \sum_j \frac{m_j}{\rho_j} \rho_j W_j$$

- m_j - маса частинки j
- ρ_j - густина частинки j

Густина



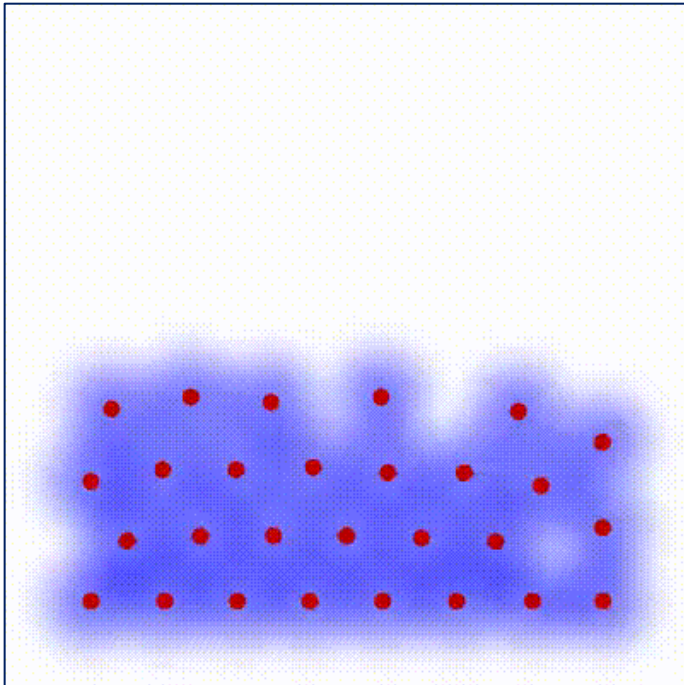
Частинки



Густина

Густина

- Густина в позиції частинки обчислюється за формулою



$$\rho_i = \sum_j \frac{m_j}{\rho_j} \rho_j W_{ij}$$

Тиск

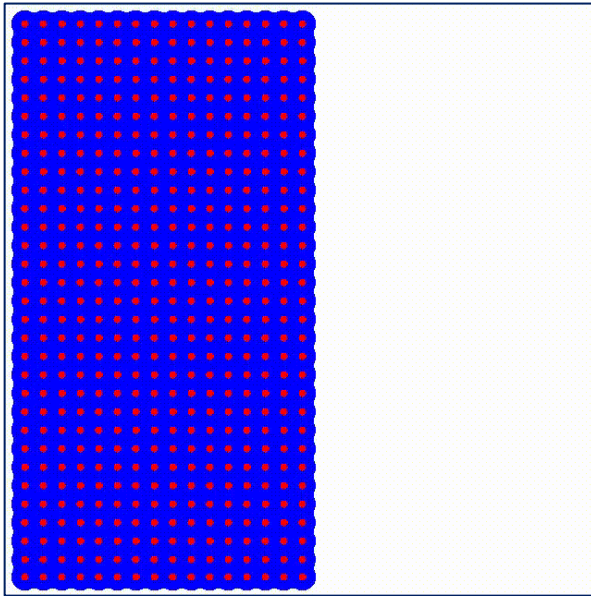
- Тиск в позиції частинки обчислюється за формулою

$$p_i = k(\rho_i - \rho_{rest})$$

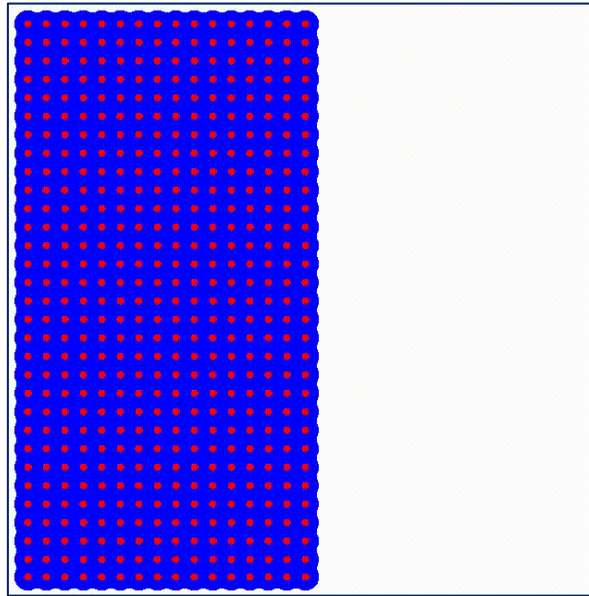
- ρ_i - густина в позиції частинки
- ρ_{rest} - густина спокою для рідини
- k - константа, що визначає рівень нестисливості.

Тиск. Константа нестисливості

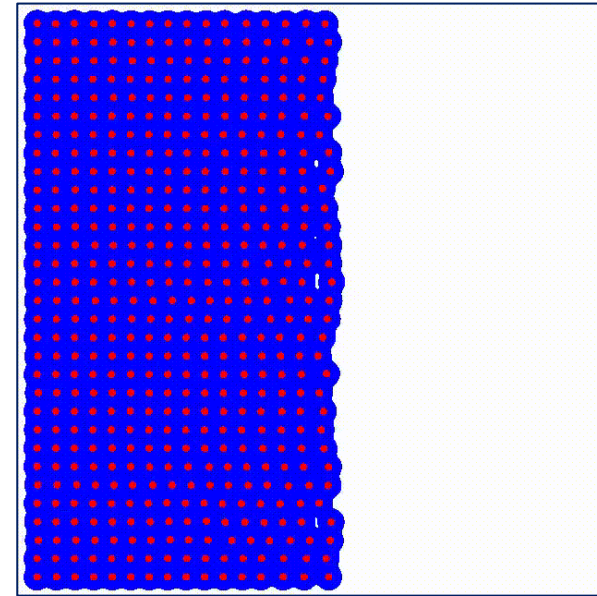
- Для моделювання умовно нестисливих рідин константа повинна мати достатньо велике значення. Проте збільшення константи при незмінній довжині часового кроку викликає дестабілізацію



Мале значення константи



Велике значення константи



Дестабілізація

Сили. Сила тиску

- Сила тиску має напрямок градієнту функції тиску і визначається за формулою

$$F_i = \sum_j \frac{m_j}{\rho_j} \frac{p_i + p_j}{2} \nabla W_{ij}$$

Сили. Сила в'язкості

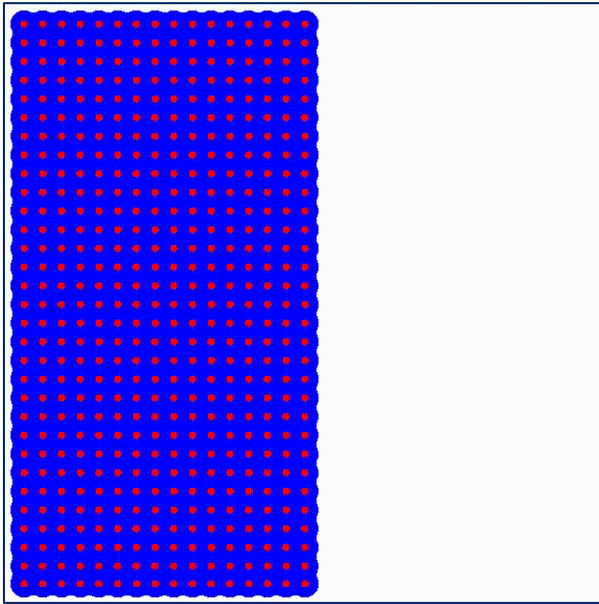
- Сила в'язкості визначається за формулою

$$F_i = \sum_j \frac{m_j}{\rho_j} (V_j - V_i) W_{ij}$$

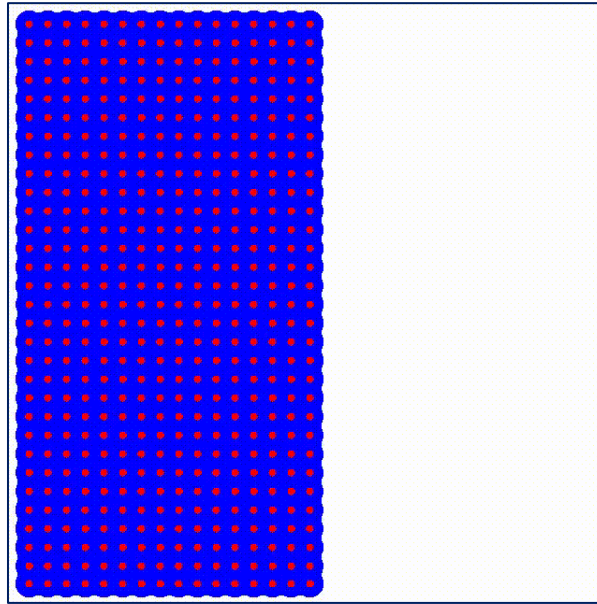
- V_j - швидкість частинки j
- V_i - швидкість частинки i

В'язкість

- В'язкість викликає сповільнення відносного руху близько розташованих частинок. Приклади в'язких речовин: смола, мед. Приклади рідких речовин: вода, олія, спирт.



Низька в'язкість

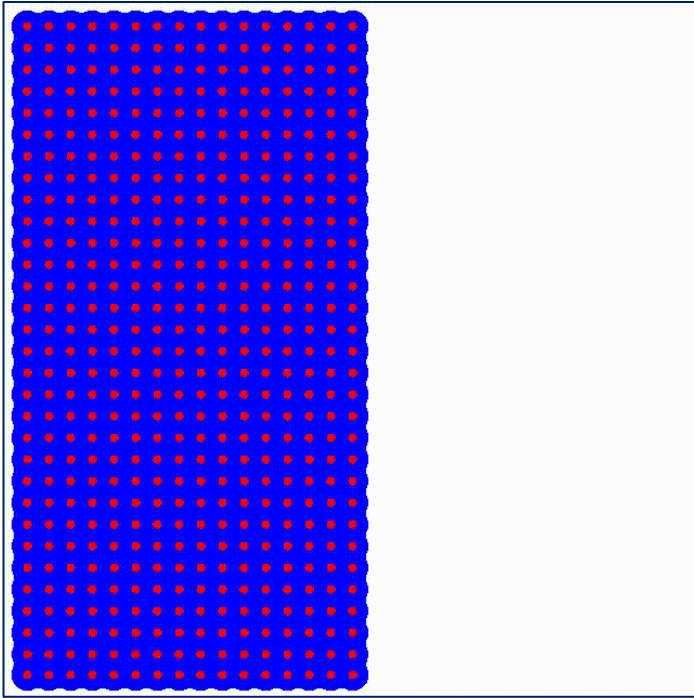


Висока в'язкість

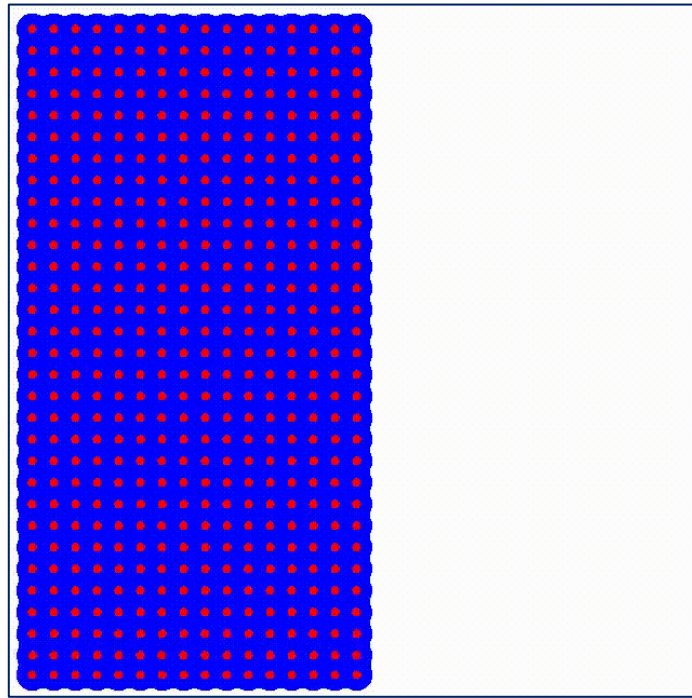
Сили. Сила тяжіння

- Сила тяжіння визначається за формулою

$$F_i = \rho_i g$$



Низька сила тяжіння



Висока сила тяжіння

Прискорення, швидкість та позиція

- Прискорення частинок обчислюється за формулою

$$a_i = f_i / \rho_i$$

- Швидкість частинок обчислюється за формулою

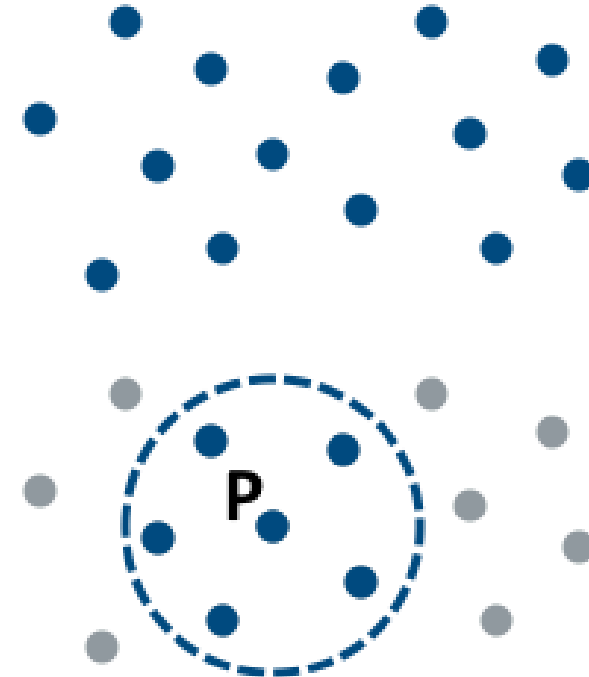
$$V_{next} = V_{prev} + a\Delta t$$

- Позиція частинок обчислюється за формулою

$$P_{next} = P_{prev} + V\Delta t$$

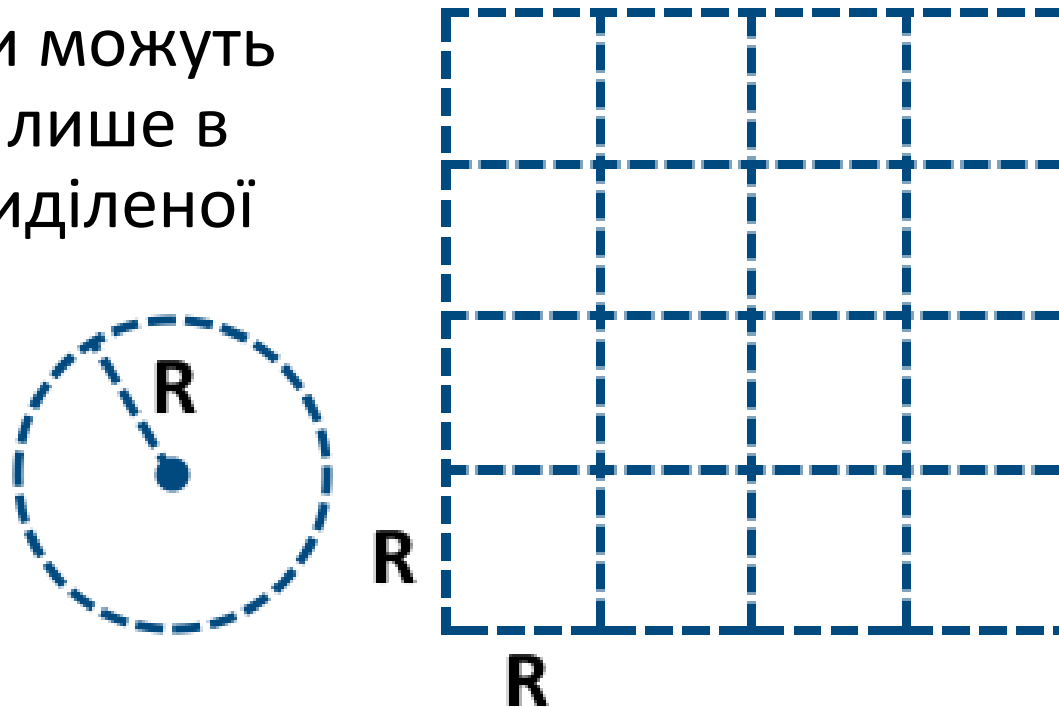
Сусідні частинки

- Взаємодія між частинками послаблюється зі збільшеннями відстані між ними. На певній відстані взаємодія є достатньо малою, щоб її можна проігнорувати з метою оптимізації обчислень
- **Радіусом** будемо називати мінімальну відстань між частинками, на якій припиняється взаємодія між ними
- **Сусідами** частинки **P** будемо називати частинки, відстань від яких до частинки **P** менша за радіус.



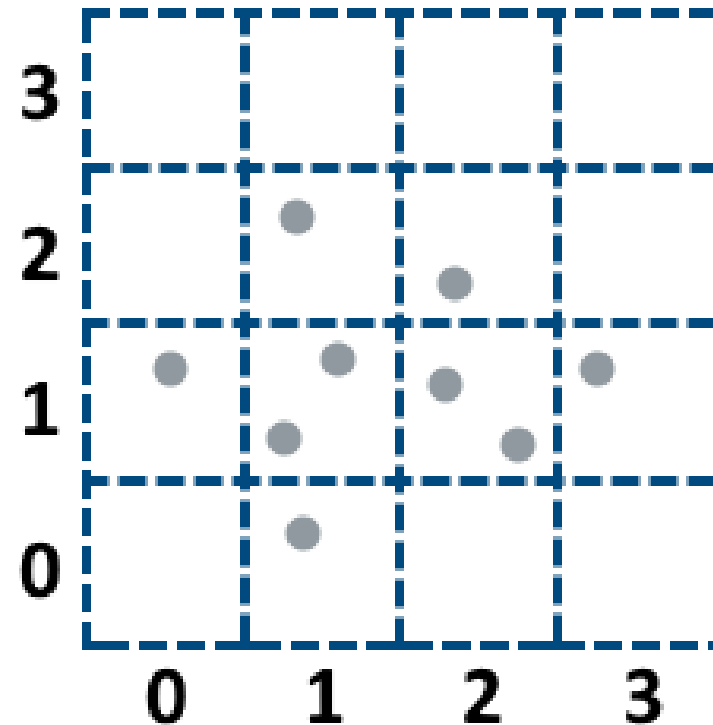
Структура даних. Просторова сітка

- Виділимо **прямокутну область** простору і розіб'ємо її на **квадратні комірки**, довжина сторони яких дорівнює радіусу частинки
- Частинки можуть існувати лише в межах виділеної області

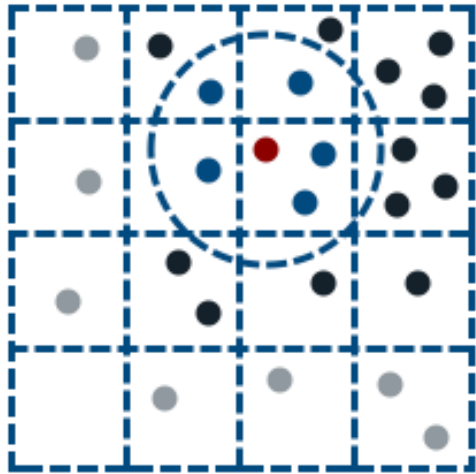


Структура даних. Просторова сітка

- Комірka задається двома або трьома (в залежності від розмірності простору) невід'ємними цілочисельними координатами
- Частинка **належить** певній комірці якщо центр частинки знаходиться всередині цієї комірки
- Кожна з частинок належить строго одній комірці, при цьому одній комірці може належати одразу декілька частинок



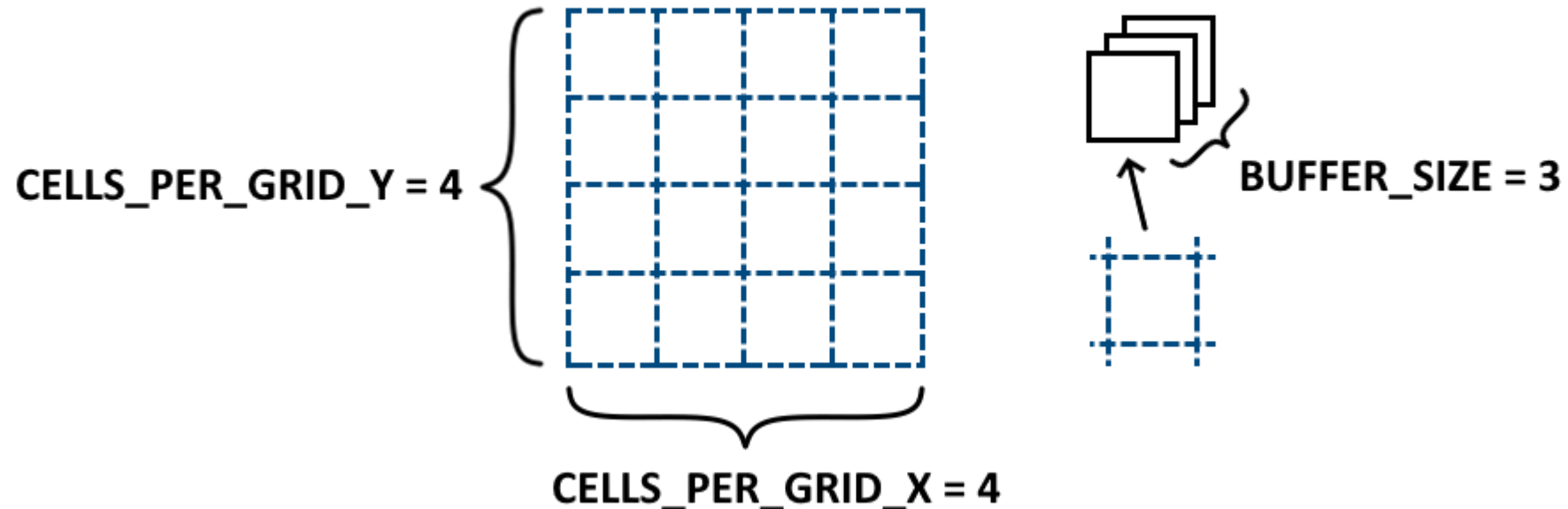
Структура даних. Просторова сітка



- Вибрана частинка
- Сусіди вибраної частинки
- Частинки сусідніх клітинок, що могли б бути сусідами
- Не приймають участь в обчисленнях

Структура даних. Просторова сітка

- Визначимо максимальну кількість частинок, що може знаходитися в одній комірці як **BUFFER_SIZE**
- Розміри сітки по двом осям визначимо як **CELLS_PER_GRID_X**, **CELLS_PER_GRID_Y**



Структура даних. Просторова сітка

- Тоді визначимо структуру даних для зберігання частинок як тривимірний масив частинок та двовимірний масив кількостей частинок:
- `Particle data[CELLS_PER_GRID_X][CELLS_PER_GRID_Y][BUFFER_SIZE];`
- `int size[CELLS_PER_GRID_X][CELLS_PER_GRID_Y];`

Етап 1. Обчислення густини

- Під час першого етапу запускаються потоки для обчислення густину рідини в позиції кожної частинки
- Потік зчитує позицію поточної частинки та сусідніх частинок і записує в результат густину рідини в позиції поточної частинки

Етап 2. Обчислення швидкості та позиції

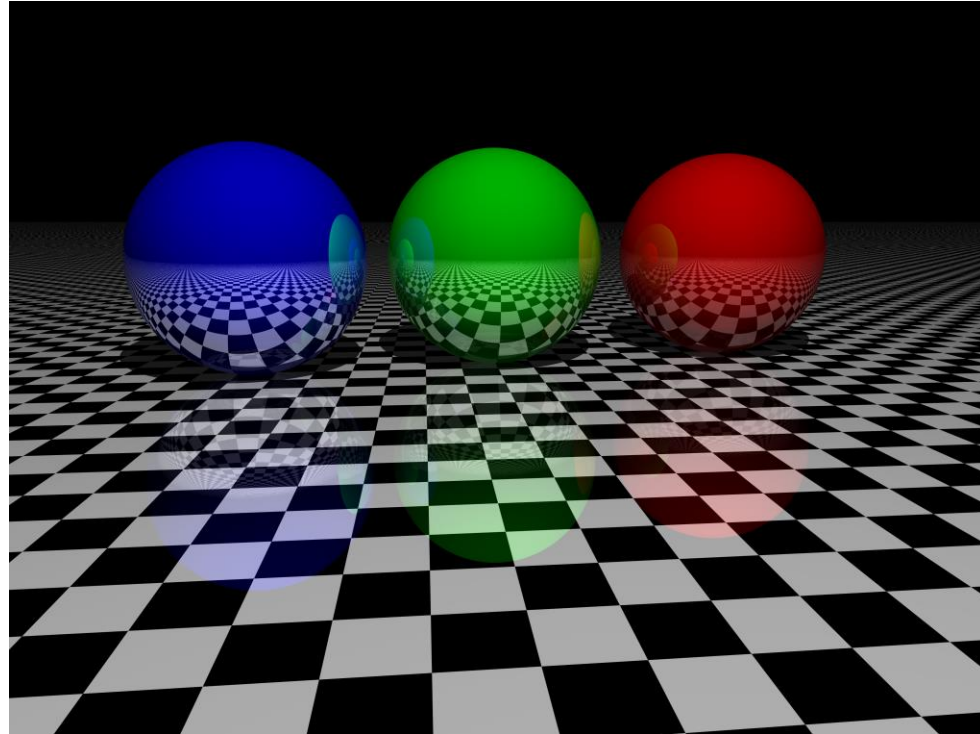
- Під час другого етапу запускаються потоки для визначення швидкості та позиції частинок після поточного кроку симуляції
- Потік зчитує позицію та швидкість поточної частинки та сусідніх частинок а також густину і записує в результат оновлену швидкість та позицію поточної частинки

Етап 3. Перерозподіл

- На попередньому етапі змінилася позиція частинок, проте їх фізичне розташування в просторовій сітці не змінилося. На поточному етапі розташування частинок в просторовій сітці не є правильним і потребує перерозподілу
- На цьому етапі будуть запуснені потоки для кожної комірки для перебудови просторової сітки
- Потік зчитує інформацію частинок в сусідніх клітинках і запише в результат лише ті частинки, які належать поточній комірці

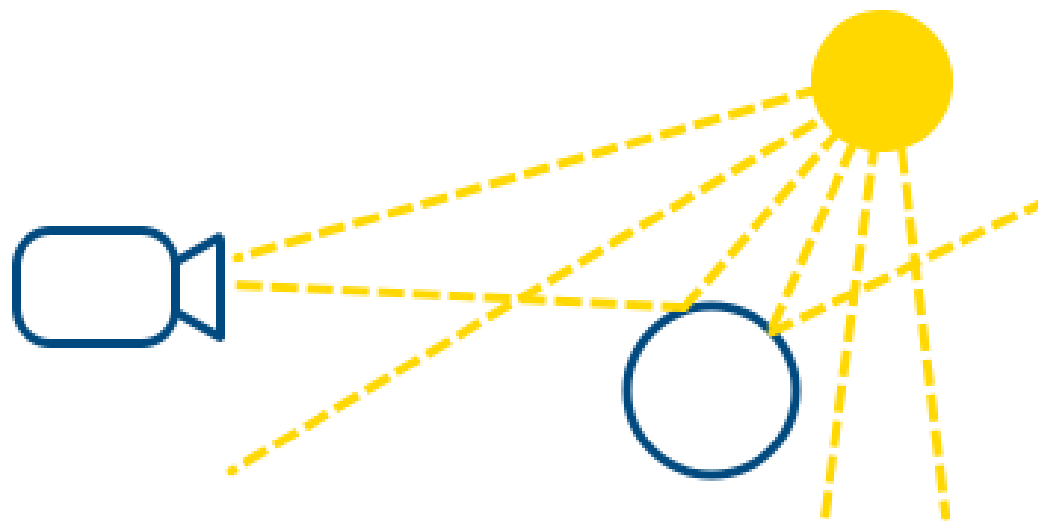
Візуалізація. Трасування променів

- Для візуалізації частинок в тривимірному просторі була вибрана технологія трасування променів



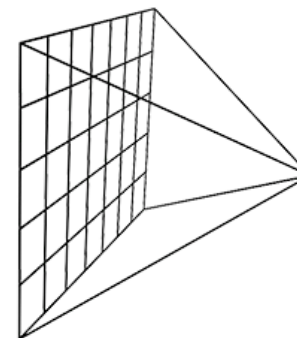
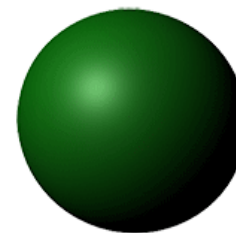
Трасування променів. Основні положення

- **Камерою** будемо називати об'єкт в просторі, з позиції якого буде відбуватися візуалізація.
- **Промінь** починає рух від джерела світла, реагує з **об'єктами сцени** і може закінчити його в позиції камери

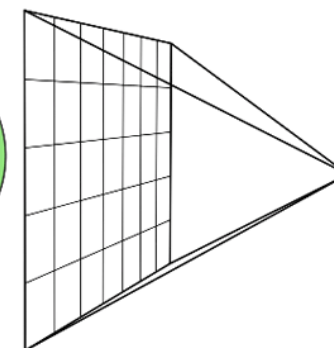
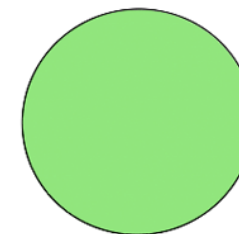


Трасування променів. Основні положення

- **Трасування** променю відбувається з кінця (позиція камери) до початку (джерело світла). Метою трасування є дізнатися **звідки** цей промінь прийшов і яку **інтенсивність** він має в кінці.
- Трасування променів запускається для кожного пікселя екрану. **Яскравість** пікселю буде відповідати **інтенсивності** променю.



© www.scratchapixel.com

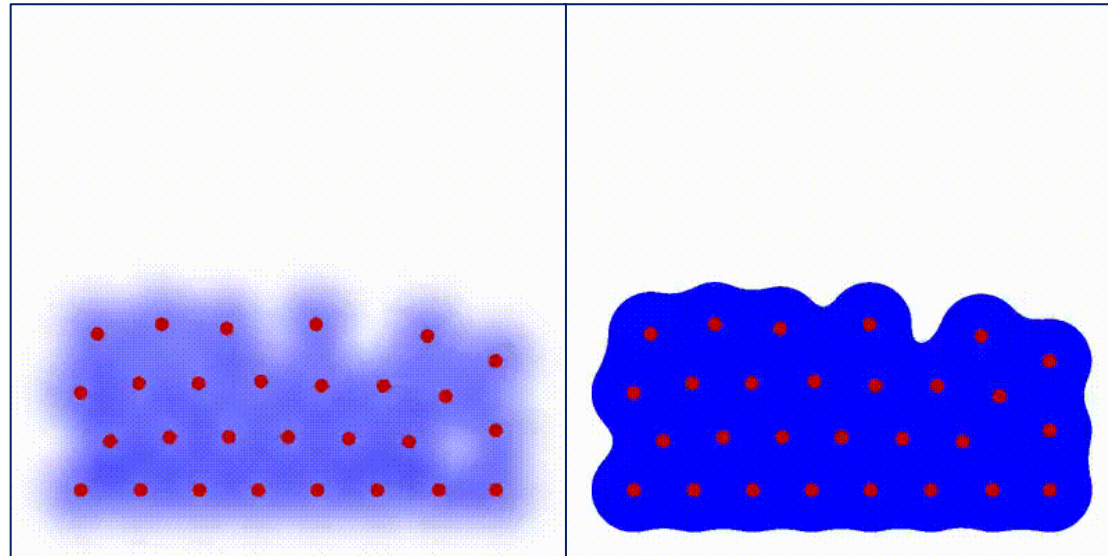


Трасування променів. Функція перетину

- Технологія трасування променів може бути використана для візуалізації будь-якої фігури, для якої визначена функція перетину.
- **Функція перетину** за позицією та напрямком променю визначає точку перетину цього променю з фігурою та нормаль поверхні фігури в цій точці перетину

Трасування променів. Функція перетину

- Для візуалізації гладкої поверхні рідини ми визначимо функцію перетину променю на основі значень густини.
- Виберемо деяке мінімальне значення густини. Якщо значення густини в точці простору більше за мінімальне, то ця точка належить фігурі.



Трасування променів. Функція перетину

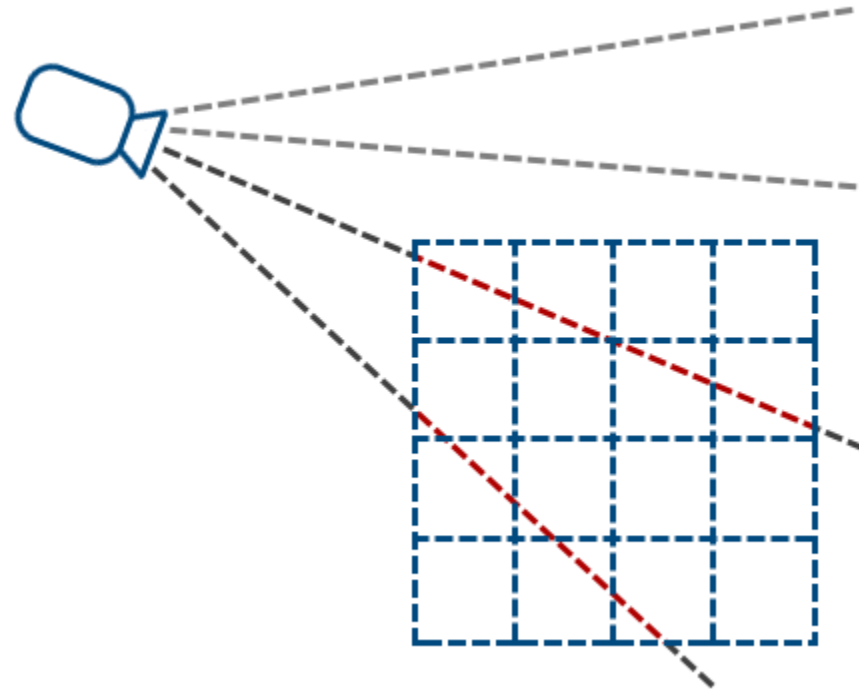
- Функцію перетину можна визначити **числовим методом**.
Наприклад послідовно з деяким **кроком** перевіряти точки, через які проходить промінь доки перетин з фігурою не буде виявлений.

Не належить
фігурі



Трасування променів. Оптимізації

- Не обраховуємо промені, які **не перетинають** виділену для симуляції **область простору**
- Якщо промінь перетинає цю область, то рухаємо промінь в найближчу позицію до цієї області

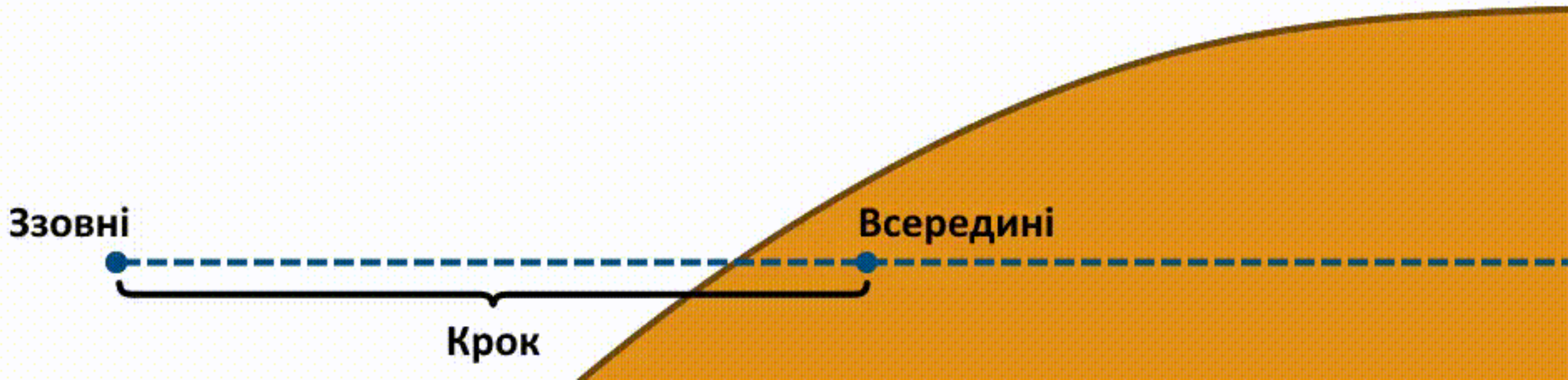


Трасування променів. Оптимізації

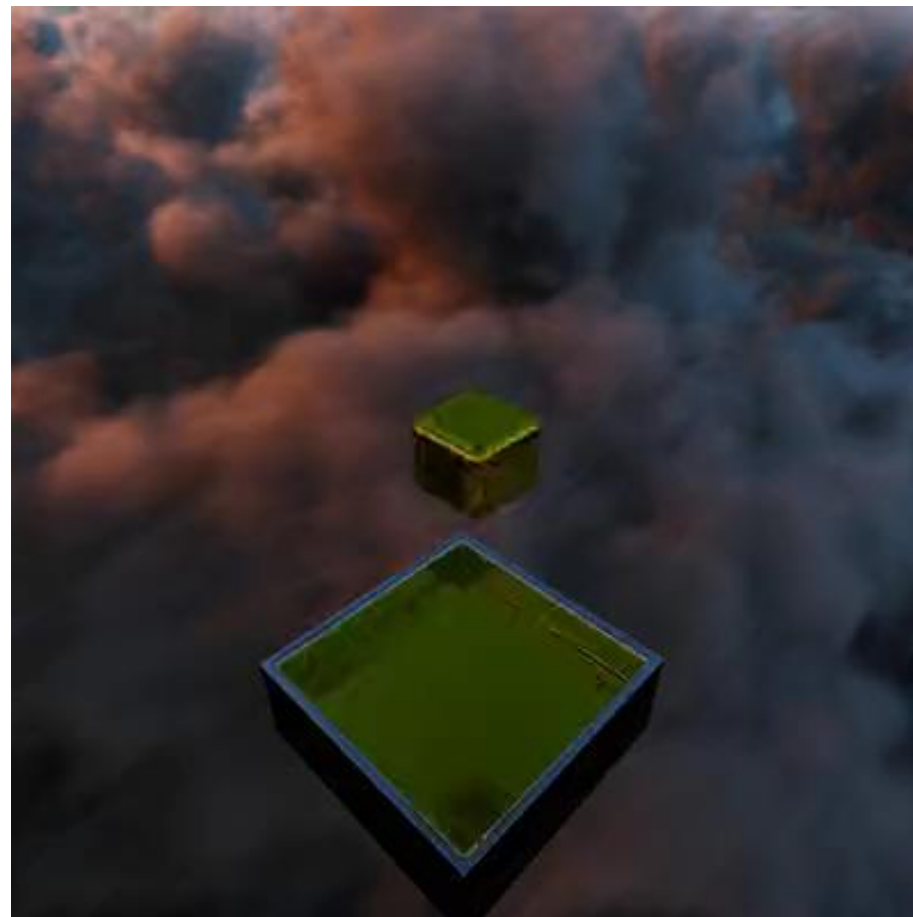
- Якщо промінь проходить через комірки, де немає частинок, то довжину кроку можна збільшити

Трасування променів. Оптимізації

- Коли в певній точці було виявлено належність фігурі можна використати **бінарний пошук** точної точки перетину променя з фігурою. Це підвищує якість отриманого зображення



Результат роботи трасування променів



Висновки

- Було проведено дослідження фізики рідини і розроблено власний програмний інструмент для моделювання рідини на GPU з використанням технології Nvidia CUDA. При розробці було досліджено апаратну будову відеокарти і принципи роботи з її пам'яттю, що дозволило досягти максимальної швидкості і продуктивності обчислень.

Висновки

- Також було розроблено новий ефективний спосіб візуалізації рідини на основі технології трасування променів, що використовує значення густини і не потребує попередньої побудови поверхні з використанням алгоритму крокуючих кубів.

Висновки

- Багатопоточне програмування під GPU достатньо легко масштабується, тому використані в роботі алгоритми можна адаптувати для моделювання рідини на суперкомп'ютерах та потужних відеокартах майбутнього.