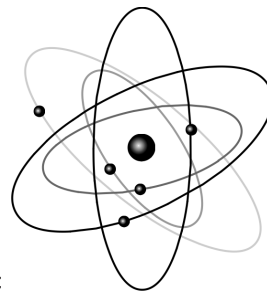




AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Fizyki i Informatyki Stosowanej



Praca inżynierska

Filip Koperski

kierunek studiów: **informatyka stosowana**

Opiekun: **prof. dr hab. inż. Bartłomiej Szafran**

Kraków, styczeń 2016

Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

.....
(czytelny podpis)

Merytoryczna ocena pracy przez opiekuna:

Merytoryczna ocena pracy przez recenzenta:

Spis treści

| | | |
|----------|------------------------------------------------------------------------|-----------|
| 1 | Wstęp | 8 |
| 1.1 | Motywacja | 8 |
| 1.2 | Mój wkład – założenia | 8 |
| 2 | Teoria | 9 |
| 2.1 | Równanie Naviera-Stokesa | 9 |
| 2.2 | Metoda SPH | 10 |
| 2.2.1 | Jądra wygładzające | 11 |
| 3 | Modelowanie płynu | 13 |
| 3.1 | Gęstość i ciśnienie cząsteczki | 13 |
| 3.2 | Siły | 13 |
| 3.2.1 | Ciśnienie | 13 |
| 3.2.2 | Lepkość | 14 |
| 3.2.3 | Napięcie powierzchniowe | 14 |
| 3.2.4 | Siły zewnętrzne | 15 |
| 3.3 | Obsługa kolizji cząstek ze ścianami | 16 |
| 3.4 | Symulacja | 16 |
| 3.4.1 | Schemat Verleta | 16 |
| 4 | Wizualizacja płynu | 17 |
| 4.1 | Znalezienie cząsteczek powierzchniowych | 17 |
| 4.1.1 | Metoda odległości do punktu środka masy | 17 |
| 4.1.2 | Metoda wykorzystująca gradient pola kolorowego | 17 |
| 4.2 | Konstrukcja mapy odległości | 18 |
| 4.3 | Renderowanie - GPU raycasting | 18 |
| 5 | Implementacja | 19 |
| 5.1 | Algorytm | 20 |
| 5.1.1 | Schemat blokowy aplikacji | 20 |
| 5.1.2 | Pseudokod algorytmu symulacji | 21 |
| 5.2 | Optymalizacje | 21 |
| 5.2.1 | Wyszukiwanie sąsiadów | 22 |
| 5.3 | Całkowanie numeryczne | 23 |
| 5.3.1 | Krok czasowy | 23 |
| 5.4 | Parametry symulacji | 24 |
| 5.4.1 | Objętość symulowanego płynu oraz masa pojedynczej cząsteczki | 24 |
| 5.4.2 | Promień odcięcia jądra wygładzającego | 24 |

| | | |
|----------|----------------------------------------------|-----------|
| 5.4.3 | Gęstość spoczynkowa | 25 |
| 5.4.4 | Lepkość | 25 |
| 5.4.5 | Napięcie powierzchniowe | 25 |
| 6 | Wyniki | 26 |
| 6.1 | Prezentacja symulacji «foto» | 26 |
| 6.2 | Analiza wydajności | 26 |
| 6.3 | Analiza stabilności numerycznej | 26 |
| 7 | Podsumowanie, wnioski i dalszy rozwój | 27 |
| | Bibliografia | 28 |

Spis rysunków

| | | |
|---|----------------------------------------------------------|----|
| 1 | test | 7 |
| 2 | Diagram | 20 |
| 3 | Diagram indeksowania i szeregowania cząsteczek | 23 |

Spis tabel

Spis kodów źródłowych

| | | |
|---|----------------------------|---|
| 1 | Painter.cpp;9:18 | 7 |
|---|----------------------------|---|

TEST



Rysunek 1: test

```
1 void Painter::paint(const Tetromino& t)
2 {
3     const auto& VBO = t.getVBO();
4
5     glBindVertexArray(VAO);
6     glBindBuffer(GL_ARRAY_BUFFER, VBO);
7     glDrawArrays(GL_TRIANGLES, 0, 6);
8     //glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, 0);
9     glBindVertexArray(0);
10 }
```

Listing 1: Painter.cpp;9:18

1 Wstęp

1.1 Motywacja

Symulacja płynów jest stosunkowo złożonym i skomplikowanym problemem. W ostatnim dziesięcioleciu staje się jednak coraz bardziej popularnym tematem opracowań naukowych. Obliczeniowa mechanika płynów (ang. **C**omputational **F**luid **D**ynamics) wykorzystuje metody numeryczne oraz stale rosnącą moc obliczeniową komputerów do rozwijania tej dyscypliny. Powodem złożoności zachowania płynów jest skomplikowany mechanizm działania różnych zjawisk składających się na mechanikę płynów, takich jak turbulentność, laminarność czy napięcie powierzchniowe płynu.

Symulacje płynu można przeprowadzać w dwojaki sposób: offline i online. Metody offline zapewniają największą dokładność obliczeń. Oprogramowania stosujące te techniki wykorzystywane są m.in. w przemyśle motoryzacyjnym, lotniczym i kosmicznym gdzie precyzja kalkulacji jest bardzo istotna. Metody dokładne sprowadzają się do rozwiązania równań Naviera-Stokesa.

W ramach tej pracy wykonano symulacje zachowania płynów w oparciu o metodę online – SPH (Smoothed Particle Hydrodynamics), która wykorzystuje podejście znane z dynamiki molekularnej i używa wariantu dynamiki Newtonowskiej z parametrami, które mają symulować zachowanie cieczy realnej. To podejście sprawdza się w symulacjach w czasie rzeczywistym lub sytuacjach gdzie wymagana jest interakcja z płynem. Metody online wykorzystywane są również we wspomnianych powyżej gałęziach przemysłu, jednak są przydatne jedynie na etapie projektowania. Poza tym metody online znajdują zastosowanie w symulacjach medycznych lub grach wideo.

1.2 Mój wkład – założenia

W niniejszej pracy moim celem było zaprezentowanie metody zdolnej do symulacji płynu w czasie rzeczywistym. Głównym założeniem pracy była możliwość interakcji z płynem w stosunkowo dużej skali przy zachowaniu satysfakcjonującej wydajności. Mój wybór, przy sugestii promotora, padł na metodę, która zdobywa w ostatnim dziesięcioleciu najwięcej rozgłosu: Smoothed Particle Hydrodynamics (pol. wygładzonej hydrodynamiki cząstek).

Osobnym krokiem w ramach symulacji płynu jest jego wizualizacja, na którą składają się m.in. elementy: wydobywania powierzchni płynu oraz jej renderowania. Ten fragment często pochłania najwięcej czasu symulacji i kluczowym jest znalezienie efektywnego rozwiązania. W niniejszej pracy został przedstawiony jeden z wielu opracowanych sposobów - jednak zdaniem autora również jeden z najciekawszych - na rozwiązanie tego problemu.

2 Teoria

Metoda SPH zastosowana przy symulacji płynów służy m.in. do dyskretyzacji równania Naviera-Stokesa.

2.1 Równanie Naviera-Stokesa

Izotermiczny płyn opisuje się przy pomocy 3 parametrów: prędkości v , gęstości ρ i ciśnienia p . W celu opisu wykorzystać można również dwa podejścia charakteryzujące ośrodek: Eulera i Lagrange’a. W tym pierwszym płyn umieszcza się na siatce i jego parametry są obliczane w poszczególnych komórkach tej siatki. Za pomocą wspomnianych trzech parametrów opisuje się postępowanie płynu w czasie za pomocą dwóch równań. Pierwsze równanie opisuje zasadę zachowania masy:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

Drugie równanie czyli równanie Naviera-Stokesa opisuje zasadę zachowania pędu (drugą zasadę dynamiki Newtona) w nieściśliwym płynie.

RÓWNANIE NAVIERA-STOKESA:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{F} \quad (2)$$

gdzie \mathbf{F} oznacza tzw. siły zewnętrzne oddziałujące na płyn a μ to współczynnik lepkości dynamicznej płynu. Z tego równania wyprowadzimy wyrażenia na siły lepkości i ciśnienia występujące w symulowanym płynie.

Przy opisie Lagrange’a wykorzystuje się cząsteczki. Każda cząsteczka reprezentuje pewną stałą objętość płynu przypisaną do tej cząsteczki i opisaną wspomnianymi powyżej trzema parametrami. Suma wszystkich cząsteczek w całości definiuje płyn. Takie założenia znacząco ułatwiają równania opisujące płyn. Liczba cząsteczek podczas symulacji jest stała co gwarantuje spełnienie zasady zachowania masy. Równanie (1) jest zawsze spełnione. Dodatkowo wyrażenie z lewej strony równania (2): $\left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right)$ jest równoważne pochodnej materialnej $\frac{D\mathbf{v}}{Dt}$ [Wik15a]. Z definicji opisu Lagrange’a cząsteczka (tj. układ odniesienia) wobec płynu pozostaje nieruchoma - porusza się razem z tym płynem. Dzięki temu stwierdzeniu wyrażenie upraszcza się do pochodnej $\frac{d\mathbf{v}}{dt}$.

Z prawej strony równania (2) pozostają trzy składniki. Wyrażenie (∇p) opisuje ciśnienie, $(\mu \nabla^2 \mathbf{u})$ opisuje lepkość a (\mathbf{F}) to pozostałe siły zewnętrzne (np. grawitacja). Równanie (2) możemy ostatecznie zapisać jako:

$$\mathbf{f}_i = -\nabla p_i + \mu \nabla^2 \mathbf{u}_i + \rho_i \mathbf{g} \quad (3)$$

Zgodnie z drugą zasadą dynamiki wartość działających sił jest proporcjonalna do zmiany pędu w czasie. Zakładając stałość masy «maso-gęstość» podczas ruchu przyspieszenie każdej z

cząsteczek można znaleźć następująco:

$$\mathbf{a}_i = \frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{f}_i}{\rho_i} \quad (4)$$

Tak obliczone przyspieszenie posłuży nam do znalezienia pozycji i prędkości w chwili czasowej dla każdej cząsteczki. Metody całkowania zostaną opisane w rozdziale (3.4).

2.2 Metoda SPH

Początkowo metoda SPH została stworzona niezależnie przez R.A. Gingolda i J.J. Monaghana (1977) oraz J.B. Lucy (1977) do symulacji problemów astrofizycznych jednak jest na tyle ogólna, że z powodzeniem znalazła zastosowanie w innych typach modelowań, m.in. w symulacji płynów. Po raz pierwszy SPH do symulacji płynów wykorzystał [MCG03]. Od tamtego czasu ewoluowała w stronę bardziej skomplikowanych algorytmów m.in.: WCSPH, PCISPH, IISPH, FLIP.

SPH jest metodą interpolacji w układach, w których używa się opisu Lagrange’a, tj. takich gdzie układ odniesienia porusza się razem z cieczą (każda cząsteczka posiada swój własny układ odniesienia). Przy wykorzystaniu SPH pole, które jest zdefiniowane w konkretnych punktach (w położeniach gdzie znajdują się cząsteczki) może zostać opisane dla dowolnego miejsca w przestrzeni. W tym celu wykorzystuje się następujące równanie:

RÓWNANIE INTERPOLACYJNE SPH:

$$A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \quad (5)$$

gdzie:

- A_S - interpolacja pola w pozycji \mathbf{r}
- j - cząsteczki po których następuje iteracja
- m_j - masa j -tej cząsteczki
- \mathbf{r}_j - pozycja j -tej cząsteczki
- ρ_j - gęstość j -tej cząsteczki
- A_j - wartość pola w punkcie \mathbf{r}_j
- $W(\mathbf{r} - \mathbf{r}_j, h)$ - jądro wygładzające dla j -tej cząsteczki (tj. jej odległości od pozycji \mathbf{r} : $\|\mathbf{r} - \mathbf{r}_j\|$)

Przy wykorzystaniu powyższego równania da się opisać poszczególne składniki równania Navier’a-Stokes’a (3) opisane w rozdziale (2.1). Opis ten znajduje się w następnym rozdziale.

Istotną właściwością SPH jest łatwość wyliczania pochodnych przestrzennych w postaci gradientu i laplasjanu pola skalarne. Gradient pola $A_S(\mathbf{r})$ w notacji SPH zapisuje się jako:

$$\nabla A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (6)$$

lub w precyzyjniejszej formie gradient pola skalarnego można zapisać jako «ref kelager 3.11»:

$$\nabla A_S(\mathbf{r}) = \rho \sum_j m_j \left(\frac{A}{\rho^2} + \frac{A_j}{\rho_j^2} \right) \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (7)$$

Laplasjan pola $A_S(\mathbf{r})$ zapisuje się jako:

$$\nabla^2 A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \quad (8)$$

2.2.1 Jądra wygładzające

Funkcja $W(\mathbf{r}, h)$ to jądro wygładzające (ang. smoothing kernel) o promieniu h . Taka funkcja musi spełniać dwa następujące warunki:

- normalizacja:

$$\int_{\Omega} W(\mathbf{r}, h) d\mathbf{r} = 1$$

- zbieżność do delty Diraca przy h dążącym do 0:

$$\lim_{h \rightarrow 0} W(\mathbf{r}, h) = \delta(\mathbf{r})$$

Oprócz powyższych warunków funkcja jądra wygładzającego musi być dodatnia:

$$W(\mathbf{r}, h) \geq 0$$

aby mogła pełnić rolę uśredniającą. Kiedy dodamy jeszcze jeden warunek funkcji jądra - symetrię:

$$W(\mathbf{r}, h) = W(-\mathbf{r}, h)$$

wtedy interpolacja jest drugiego stopnia precyzji, to znaczy że błąd przybliżenia sumowania w równaniu (5) jest równy $O(h^2)$ lub lepszy. Jądro zanika dla promienia większego od \mathbf{r} :

$$W(\mathbf{r}, h) = 0, \|\mathbf{r}\| > h$$

Dla zastosowań SPH, w oparciu o powyższe założenia, projektuje się różne funkcje jąder wygładzających. Od nich w dużym stopniu zależy stabilność, dokładność i wydajność metody. Za idealną funkcję jądra uważa się krzywą Gaussa. Jej obliczenie jest jednak skomplikowane (przez zawartą funkcję eksponent) dlatego w praktyce stosuje się prostsze funkcje przypominające tą krzywą.

Do większości obliczeń korzysta się z jądra wygładzającego opisanego stosunkowo tanim obliczeniowo funkcją wielomianową W_{poly6} :

$$W_{poly6}(\mathbf{r}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - \|\mathbf{r}\|^2)^3 & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (9)$$

$$\nabla W_{poly6}(\mathbf{r}, h) = \frac{945}{32\pi h^9} \begin{cases} \mathbf{r} (h^2 - \|\mathbf{r}\|^2)^2 & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (10)$$

$$\nabla^2 W_{poly6}(\mathbf{r}, h) = \frac{945}{32\pi h^9} \begin{cases} (h^2 - \|\mathbf{r}\|^2) \left(\|\mathbf{r}\|^2 - \frac{3}{4}(h^2 \|\mathbf{r}\|^2) \right) & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (11)$$

Powyższe jądro wykorzystywane jest przy interpolacji gęstości, ciśnienia oraz pola kolorowego. Przy obliczaniu gradientu ciśnienia oraz siły lepkości, z racji natury oddziaływania tych sił, konieczne jest zastosowanie innych jąder wygładzających.

W przypadku zastosowania jądra W_{poly6} przy obliczaniu siły ciśnienia cząsteczki mają tendencję do «zlepiania, gromadzenia» się. Dla małych odległości pomiędzy cząsteczkami gradient tego jądra wykorzystywany do obliczania ciśnienia pomiędzy cząsteczkami dąży do zera i siła odpychająca cząsteczki zanika. Ten problem rozwiązuje się korzystając z innego jądra wygładzającego - $W_{spiky}(\mathbf{r}, h)$:

$$W_{spiky}(\mathbf{r}, h) = \frac{15}{\pi h^6} \begin{cases} (h - \|\mathbf{r}\|)^3 & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (12)$$

$$\nabla W_{spiky}(\mathbf{r}, h) = -\frac{45}{\pi h^6} \begin{cases} \frac{\mathbf{r}}{\|\mathbf{r}\|} (h - \|\mathbf{r}\|)^2 & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (13)$$

Gradient tego jądra pozostaje niezerowy dla argumentów w pobliżu zera, co generuje odpowiednie siły odpychające. Spełnia również pozostałe warunki konieczne dla jąder wygładzających.

Do obliczenia siły lepkości konieczne jest wprowadzenie jeszcze jednego jądra wygładzającego. Lepkość jest zjawiskiem, które - spowodowane tarcie - prowadzi do rozproszenia energii kinetycznej płynu w postaci ciepła. Laplasjan takiego jądra musi być dodatni aby zjawisko lepkości było poprawnie odwzorowywane. Stosuje się tutaj jądro wygładzające $W_{viscosity}(\mathbf{r}, h)$ opisane następującym wzorem:

$$W_{viscosity}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \begin{cases} \left(-\frac{\|\mathbf{r}\|^3}{2h^3} + \frac{\|\mathbf{r}\|^2}{h^2} + \frac{h}{2\|\mathbf{r}\|} - 1 \right) & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (14)$$

$$\nabla W_{viscosity}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \begin{cases} \mathbf{r} \left(-\frac{3\|\mathbf{r}\|}{2h^3} + \frac{2}{h^2} - \frac{h}{2\|\mathbf{r}\|^3} \right) & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (15)$$

$$\nabla^2 W_{viscosity}(\mathbf{r}, h) = \frac{45}{\pi h^6} \begin{cases} (h - \|\mathbf{r}\|) & 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \|\mathbf{r}\| > h \end{cases} \quad (16)$$

3 Modelowanie płynu

Dzięki metodzie SPH można modelować poszczególne elementy równania Naviera-Stokesa, wyróżnione w rozdziale (2) opisane w punkcie (2.1).

Organizacja tego rozdziału odpowiada kolejności obliczania elementów składających się na implementację metody SPH. W pierwszym kroku obliczana jest wartość gęstości dla wszystkich cząsteczek. Następnie znajduwane są siły z jakimi oddziałują na siebie cząsteczki. Wypadkowa suma sił służy do obliczenia przyspieszenia każdej z cząsteczek. Dalej następuje wykrycie i odpowiedź na ewentualne kolizje cząsteczek ze ścianami. Ostatnim krokiem jest całkowanie gdzie symulacja postępuje, zgodnie z obliczonym wcześniej przyspieszeniem.

3.1 Gęstość i ciśnienie cząsteczki

Zamin zostaną obliczone siły pomiędzy cząsteczkami, konieczne jest znalezienie ich masy oraz gęstości. Masa każdej cząsteczki jest identyczna oraz stała przez cały czas symulacji. Wartość gęstości natomiast jest zmienna i trzeba ją obliczać na początku każdej iteracji symulacji. Poprzez podstawienie do wzoru (5) otrzymujemy wzór na gęstość w punkcie \mathbf{r} :

$$\rho_i(\mathbf{r}_i) = \sum_j m_j \frac{\rho_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) = \sum_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (17)$$

Następnie za pomocą równania gazu doskonałego może zostać obliczone ciśnienie. W symulacji wykorzystuje się zmodyfikowaną formę równania gazu doskonałego:

$$p_i = k(\rho_i - \rho_0) \quad (18)$$

gdzie k to stała gazowa zależna od temperatury, stała ρ_0 oznacza tzw. gęstość spoczynkową, a ρ_i to gęstość obliczona równaniem (17). «słowo komentarza» «w innych pracach gęstość jest czasem nazywana mass-density»

3.2 Siły

Przy pomocy notacji SPH wyprowadzone zostały równania na składniki równania Naviera-Stokesa.

3.2.1 Ciśnienie

«zależy od gradientu ciśnienia»

$$\mathbf{F}_i^{cisl} = -\nabla p(\mathbf{r}_i) = -\sum_j \frac{m_j}{\rho_j} p_j \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (19)$$

Niestety tak przedstawiona siła ciśnienia nie jest symetryczna. Takie oddziaływanie cząsteczek widoczne jest w przypadku interakcji dwóch cząsteczek: przy liczeniu siły ciśnienia

pierwsza cząsteczka i wykorzystuje jedynie wartość ciśnienia drugiej cząsteczki p_j , i na odwrót (kiedy wartości parametrów p_i i p_j są różne). W ten sposób łamana jest III zasada dynamiki Newtona - siły wzajemnego oddziaływania cząsteczek nie mają takich samych wartości. Aby rozwiązać problem asymetryczności w tym miejscu stosuje się alternatywny wzór SPH (7) na gradient pola:

$$\mathbf{F}_i^{cisl} = -\nabla p(\mathbf{r}_i) = -\rho_j \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (20)$$

3.2.2 Lepkość

«zależy od pola prędkości»

$$\mathbf{F}_i^{lep} = \mu \nabla^2 \mathbf{v}(\mathbf{r}_i) = \mu \sum_j \frac{m_j}{\rho_j} \mathbf{v}_j \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (21)$$

Podobnie jak w przypadku siły ciśnienia, tak przedstawiona siła lepkości jest asymetryczna. Korzystając jednak z faktu, iż lepkość zależy nie tyle od bezwzględnej prędkości płynu ale od różnic prędkości w płynie, naturalnym sposobem symetryzacji jest podstawienie w miejsce prędkości \mathbf{v}_j różnic prędkości dwóch cząsteczek:

$$\mathbf{F}_i^{lep} = \mu \nabla^2 \mathbf{v}(\mathbf{r}_i) = \mu \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_i - \mathbf{v}_j) \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (22)$$

3.2.3 Napięcie powierzchniowe

Napięcie powierzchniowe nie jest uwzględnione w równaniach Naviera-Stokesa, które dotyczą objętości cieczy, a nie jej powierzchni. Efekt napięcia powierzchniowego wprowadzany jest jako warunek brzegowy. Cząsteczki w płynie przyciągają się ze swoimi sąsiadami poprzez wzajemne oddziaływania. Podczas gdy wewnątrz płynu te oddziaływania się równoważą, na powierzchni płynu - gdzie występuje kontakt z innym ośrodkiem (np. powietrzem) - równowaga sił nie jest zachowana. W tym miejscu do gry wchodzi dodatkowa siła nazywana właśnie napięciem powierzchniowym. Powoduje ona dociskanie cząsteczek ‘brzegowych’ w kierunku normalnym do powierzchni płynu, o zwrocie ‘do’ tej powierzchni. Jednocześnie siła zmierza do minimalizacji zakrzywienia powierzchni. Im większe jest zakrzywienie powierzchni tym większa jest ta siła. Prowadzi to do ‘wyplaszczania’ powierzchni płynu.

Zgodnie z powyższym paragrafem napięcie powierzchniowe zostanie obliczone jedynie dla cząsteczek będących na powierzchni płynu. Znaleźć je można poprzez tzw. pole kolorowe (ang. color field). Pole to przyjmuje wartość 1 wewnątrz objętości płynu (tj. wszędzie tam gdzie znajdują się cząsteczki) i 0 w pozostałych miejscach.:

$$c_S(\mathbf{r}_i) = \sum_j \frac{m_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (23)$$

Obliczając gradient tego pola uzyskuje się informację o «normalnych/wektorach prostopadłych do» powierzchni płynu, skierowanych do jego wnętrza:

$$\mathbf{n}_i = \nabla c_S(\mathbf{r}_i) \quad (24)$$

Dywergencja gradientu pola to pole skalarne oznaczające długość wektorów normalnych powierzchni. Długości poszczególnych wektorów oznaczają wielkość zakrzywienia powierzchni w danym miejscu (tj. w miejscu cząsteczki i):

$$\kappa_i = -\frac{\nabla^2 c_S(\mathbf{r}_i)}{\|\mathbf{n}_i\|} \quad (25)$$

Minus w powyższym równaniu pozwala na uzyskiwanie dodatnich wartości zakrzywienia dla powierzchni wypukłych.

Łącząc poszczególne elementy otrzymujemy wzór na siłę działającą na jednostkę powierzchni płynu (ang. surface traction):

$$\mathbf{t} = \sigma \kappa \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad (26)$$

Występująca w nim stała σ oznacza współczynnik napięcia powierzchniowego charakterystyczny dla substancji tworzącej płyn.

Aby obliczana siła była aplikowana tylko do cząsteczek znajdujących się na powierzchni wzór (26) mnoży się przez znormalizowane pole skalarne $\delta = \|\mathbf{n}\|$, które maleje wraz z oddalaniem się od powierzchni płynu (jest niezerowe jedynie dla cząsteczek na powierzchni). Ostatecznie wzór na siłę napięcia powierzchniowego wygląda następująco:

$$\mathbf{F}_i^{pow} = \delta_i \mathbf{t}_i = \sigma \kappa_i \mathbf{n}_i = -\sigma \nabla^2 c_S(\mathbf{r}_i) \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|} \quad (27)$$

Jak zostało wspomniane powyżej, wartość $\|\mathbf{n}\|$ jest mała dla cząstek ‘wewnątrz’ płynu. Dzielenie przez tę wartość w powyższym równaniu wprowadza się potencjalną niestabilność numeryczną. Rozwiązaniem tego problemu jest obliczanie tego równania jedynie dla wartości $\|\mathbf{n}\|$ przekraczających pewien stały próg l :

$$\|\mathbf{n}_i\| \geq l \quad (28)$$

gdzie $l > 0$.

3.2.4 Siły zewnętrzne

Do symulacji można swobodnie dodawać kolejne siły, które wpływają na zachowanie się płynu. Może być to siła wywoływana przez interakcje użytkownika lub siła symulująca środowisko w jakim znajduje się płyn, np. siła grawitacji.

3.3 Obsługa kolizji cząstek ze ścianami

Naczynie, w którym umieszcza się płyn definiuje się poprzez określenie jego ścianek (np. w postaci trójkątów) wraz z ich wektorami normalnymi. W ten sposób można bez trudu przenieść do środowiska symulacji bryłę powstałą w programach do tworzenia trójwymiarowej grafiki komputerowej.

Obsługę kolizji cząstek ze ściankami naczynia można podzielić na dwa etapy. Pierwszym jest ustalenie czy dana cząsteczka będzie kolidować ze ścianką. Decyduje o tym odległość cząsteczki od ścianki - jeśli jest ona mniejsza od promienia jądra wygładzającego wynik detekcji jest pozytywny. [Po pozytywnym wyniku detekcji kolizji]

Drugim etapem jest obsługa zderzenia cząsteczki ze ścianką. Aby uniemożliwić penetrację aplikowana jest siła sprężysta proporcjonalna co do wartości do odległości cząsteczki od ścianki, w kierunku prostopadłym do tej ścianki. Na charakterystykę tej siły wpływają dwa parametry: sztywność oraz sprężystość ścianek. Oba muszą być tak ustalone aby nie dopuścić penetracji ścianek przez cząsteczki.

Taki sposób obsługi kolizji powoduje niestety gwałtowne zmiany ciśnienia co negatywnie wpływa na stabilność symulacji [Ihm+10]. W praktyce, dla każdej symulacji parametry ścianek muszą być dobierane osobno.

3.4 Symulacja

3.4.1 Schemat Verleta

Do całkowania równania (4) wykorzystywany jest tzw. prędkościowy schemat Verleta [Wik15b]. Kolejne położenia cząstek są obliczane następującym wzorem:

$$\mathbf{r}_{t+\Delta t} = 2\mathbf{r}_t - \mathbf{r}_{t-\Delta t} + a_t \Delta t^2 \quad (29)$$

gdzie $[\mathbf{r}_{t-\Delta t}, \mathbf{r}_t, \mathbf{r}_{t+\Delta t}]$ to położenia cząstki w kolejnych trzech chwilach czasowych, Δt to wlekość kroku czasowego a a_t to przyspieszenie wynikające z wypadkowych sił (opisanych w sekcji (3.2)) działających na cząstkę w danej chwili czasowej. Prędkość cząstki wylicza się na podstawie położenia w dwóch następnych chwilach czasowych:

$$\mathbf{v}_{t+\Delta t} = \frac{\mathbf{r}_{t+\Delta t} - \mathbf{r}_t}{\Delta t} \quad (30)$$

4 Wizualizacja płynu

Wizualizacja płynu opiera się na metodzie opisanej w pracy [Gos+10]. W tym podejściu powierzchnia płynu reprezentowana jest przez pole wokseli. Każdy woksel zawiera dystans do najbliższej powierzchni. W ten sposób reprezentuje tzw. mapę odległości (ang. distance field). «foto DF w 2d» Głównymi elementami opisywanej metody wizualizacji są: znalezienie cząsteczek na powierzchni płynu, konstrukcja mapy odległości oraz przeprowadzenie raycastingu.

4.1 Znalezienie cząsteczek powierzchniowych

Cząsteczki, które nie leżą na powierzchni płynu nie są potrzebne do jego wizualizacji i mogą zostać pominięte. Znacznie zmniejsza to koszt generacji mapy odległości za czym idzie przyspieszenie procesu wizualizacji.

Znalezienie cząsteczek leżących na powierzchni można przeprowadzić różnymi metodami. Jedna z nich wykorzystuje gradient pola kolorowego (ang. color field) wykorzystywane przy obliczaniu napięcia powierzchniowego płynu [Kel06]. Kolejna opiera się na obliczaniu odległości danej cząsteczki do punktu środka masy (ang. center of mass). Poniżej zostały opisane obie wymienione metody.

4.1.1 Metoda odległości do punktu środka masy

Cząsteczka i jest zaliczana do cząsteczek powierzchniowych jeżeli jej dystans do środka masy \mathbf{r}_{CM_i} jej sąsiedztwa jest większy niż pewna określona wartość. Środek masy można obliczyć poprzez zsumowanie pozycji wszystkich sąsiadujących cząsteczek (w układzie odniesienia i -tej cząsteczki) ważone przez masę cząsteczek:

$$\mathbf{r}_{CM_i} = \frac{\sum_j m_j \mathbf{r}_j}{\sum_j m_j} \quad (31)$$

Okazuje się jednak, że powyższy warunek jest niewystarczający w miejscach gdzie występuje niewiele cząsteczek. Dodatkowym warunkiem jest automatyczne zaliczenie cząsteczki do powierzchniowych gdy liczba cząsteczek w jej sąsiedztwie jest poniżej pewnego progu.

Jeżeli cząsteczka spełnia powyższe warunki, wpisywana jest do odpowiedniej tablicy. W przeciwnym wypadku jest po prostu pomijana.

4.1.2 Metoda wykorzystująca gradient pola kolorowego

Pole kolorowe pochodzi od funkcji (23), która definiuje kształt symulowanego płynu. Korzystając z gradientu tego pola można zdefiniować normalne powierzchni płynu. Gradient ten rośnie najbardziej w kierunku ‘na zewnątrz’ płynu. Cząsteczki powierzchniowe to te, w których miejscu długość normalnych do powierzchni przekracza pewien próg.

$$|\mathbf{n}| > t_{normal} \quad (32)$$

Gradient pola kolorowego obliczany jest już przy modelowaniu siły napięcia powierzchniowego (3.2.3). Z tego powodu klasyfikację cząstek można przeprowadzić już na tamtym etapie, bez dodatkowych obliczeń.

4.2 Konstrukcja mapy odległości

Mapa odległości zbudowana jest z jednorodnej trójwymiarowej siatki wokseli. Gęstość tej siatki ma istotne znaczenie dla wydajności jak i jakości odwzorowania płynu. Siatka o większej ilości wokseli pozwala na zachowanie większej ilości szczegółów ale wymaga więcej czasu na zbudowanie - czas ten rośnie w tempie $O(n^3)$. Wpływ na szybkość konstrukcji mapy ma również ilość cząsteczek powierzchniowych.

Podejście proponowane przez [Gos+10] wygląda następująco: siatka wokseli inicjowana jest wartością maksymalną r_{max} . Dana cząsteczka i otaczana jest sześcianem o boku długości r . Dla każdego woksela znajdującego się wewnątrz tego sześcianu obliczany jest dystans d do cząsteczki i . Wartość woksela określana jest według następującego wzoru:

$$d_v = \min(d, d_v^{old}) \quad (33)$$

Jeśli d jest mniejsze niż r_{min} , to wokselowi zostaje przypisana wartość r_{min} .

4.3 Renderowanie - GPU raycasting

W ostatnim kroku wizualizacji płynu dane zawarte w mapie odległości (przechowywanej w pamięci karty graficznej jako tekstura 3D) poddawane są procesowi raycastingu [Had+05]. Raycasting polega na przeprowadzeniu promienia przez renderowany obszar dla każdego fragmentu (piksela) obrazu; Dla każdego piksela wykonywany jest mały program - fragment shader - który inicjuje promień zberający krok po kroku próbki z danej objętości. Początek i koniec promienia muszą być znane przed uruchomieniem shadera. Dane te uzyskuje się poprzez uprzednie stworzenie bryły ograniczającej objętość renderowanego płynu. W najprostrzym przypadku bryłą tą jest prostopadłościan. Bryłę tą poddaje się procesowi rasteryzacji (w dwóch przejściach [ang. rendering pass] renderując raz przednią a raz tylną ścianę) do dwóch tekstur. Obliczając różnicę kolorów pomiędzy tymi dwoma teksturami można uzyskać kierunek promienia dla każdego piksela ograniczonego tą bryłą.

«foto bryła»

Poprzez konstrukcję bryły lepiej dopasowanej do objętości płynu można uniknąć wysyłania dużej części promieni, które nigdy nie przetną tej objętości. Istotnie zwiększa to wydajność raycastingu. Technika ta została opisana w [Haf15].

Częstotliwość próbkowania mapy odległości wzdłuż promienia ma wpływ na prędkość procesu raycastingu jak i na jakość uzyskiwanego obrazu.

5 Implementacja

Przedstawioną metodę zaimplementowano w języku C++, przy użyciu elementów języka dodanych przy standaryzacji w wersji C++11/14. Wykorzystano również następujące biblioteki:

- glm - (OpenGL Mathematics) biblioteka zawierająca funkcje i klasy wykorzystywane przy operacjach matematycznych. W zamyśle naśladuje zapis OpenGL Shading Language (GLSL).
- OpenGL - (ang. Open Graphics Library) API służące do tworzenia grafiki.
- GLEW - (OpenGL Extension Wrangler) biblioteka pomocna przy ładowaniu odpowiednich elementów biblioteki OpenGL.
- GLFW - biblioteka zawiera zestaw funkcji ułatwiających m.in. tworzenie okna dla aplikacji wykorzystujących OpenGL i obsługę urządzeń wejściowych (klawiatury, myszki, joysticka).

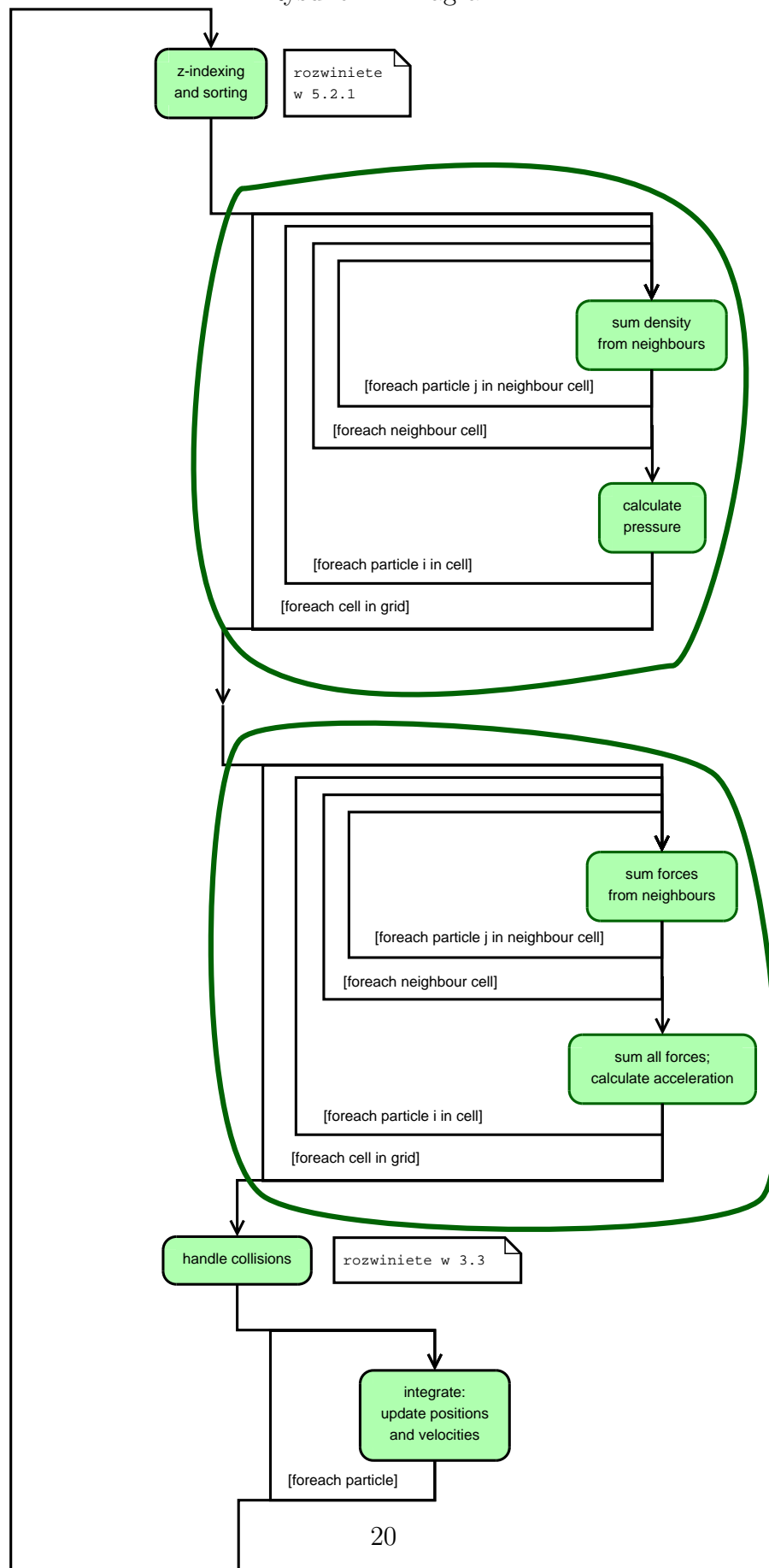
Kod źródłowy aplikacji dostępny jest w publicznym repozytorium kodu pod adresem:

<https://github.com/filu005/SPH>.

5.1 Algorytm

5.1.1 Schemat blokowy aplikacji

Rysunek 2: Diagram



5.1.2 Pseudokod algorytmu symulacji

```

stwórz cząsteczki;
wyczyść siatkę sąsiadów;
posortuj cząsteczki według z-indeksów;
umieść cząsteczki na siatce;
forall komórki w siatce do ▷ obliczenie gęstości
    foreach cząsteczki  $i$  w komórce do
        foreach sąsiadującej komórki do
            foreach cząsteczki  $j$  w sąsiadującej komórce do
                 $\rho_i(\mathbf{r}_i) \leftarrow \rho_i(\mathbf{r}_i) + m_j W(\mathbf{r}_i - \mathbf{r}_j, h)$ 
             $p_i \leftarrow k((\frac{\rho_i}{\rho_0})^7 - 1)$ 
forall komórki w siatce do ▷ obliczenie sił
    foreach cząsteczki  $i$  w komórce do
        foreach sąsiadującej komórki do
            foreach cząsteczki  $j$  w sąsiadującej komórce do
                 $\mathbf{F}_i^{cisl} \xleftarrow{i \neq j} \mathbf{F}_i^{cisl} - \rho_j m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$ 
                 $\mathbf{F}_i^{lep} \leftarrow \mathbf{F}_i^{lep} + \mu \frac{m_j}{\rho_j} (\mathbf{v}_i - \mathbf{v}_j) \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h)$ 
                 $\nabla c_S(\mathbf{r}_i) \leftarrow \nabla c_S(\mathbf{r}_i) + \frac{m_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$ 
                 $\nabla^2 c_S(\mathbf{r}_i) \leftarrow \nabla^2 c_S(\mathbf{r}_i) + \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h)$ 
            if  $\|\nabla c_S(\mathbf{r}_i)\| \geq l$  then
                 $\mathbf{F}_i^{pow} \leftarrow -\sigma \nabla^2 c_S(\mathbf{r}_i) \frac{\nabla c_S(\mathbf{r}_i)}{\|\nabla c_S(\mathbf{r}_i)\|}$ 
             $\mathbf{F}_i^{total} \leftarrow \mathbf{F}_i^{cisl} + \mathbf{F}_i^{lep} + \mathbf{F}_i^{pow} + \mathbf{F}_i^{zew}$ 
             $\mathbf{a}_i \leftarrow \frac{\mathbf{F}_i^{total}}{g}$ 
foreach cząsteczka do ▷ obsługa kolizji
    foreach ścianka naczynia do
        oblicz dystans  $d$  cząsteczki do ścianki;
        if  $d > \text{promień jądra wygładzającego } h$  then
            odepchnij cząsteczkę;
foreach cząsteczka  $i$  do ▷ całkowanie
     $\mathbf{x}_i(t + \Delta t) \leftarrow \mathbf{x}_i(t) + \mathbf{v}_i(t) \Delta t + \frac{1}{2} \mathbf{a}_i(t) \Delta t^2$ 
     $\mathbf{v}_i(t + \Delta t) \leftarrow \frac{1}{\Delta t} (\mathbf{x}_i(t + \Delta t) - \mathbf{x}_i(t))$ 

```

5.2 Optymalizacje

Wydajność aplikacji poprawiać można na wiele sposobów. Najwięcej czasu pochłaniają obliczenia matematyczne związane z symulacją. Tutaj z pomocą przychodzą metody wyszukiwania

sąsiadów cząstki koniecznych do obliczenia fizycznych właściwości. Kolejnym ‘wąskim gardłem’ jest czas dostępu do pamięci RAM więc istotna jest lokalność danych i porządkowanie ich w pamięci podręcznej. Odrębnym tematem jest optymalizacja renderowania powierzchni płynu. «zrównoleglanie obliczeń i GPGPU»

5.2.1 Wyszukiwanie sąsiadów

Stworzona na potrzeby tej pracy implementacja struktury do optymalizacji poszukiwań sąsiednich cząsteczek, wraz ze sposobem ich sortowania, została opisana w [Ihm13].

Obliczając parametr symulacji dla każdej cząsteczki przy użyciu SPH iteruje się po wszystkich pozostałych cząsteczkach. Jako, że dla każdej cząsteczki musimy poznać jej parametry, złożoność obliczeniowa iteracji dla n cząsteczek wynosi $O(n^2)$. Chcąc symulować płyn opisany liczbą cząsteczek rzędu 10’000 w czasie rzeczywistym, taka złożoność obliczeniowa jest nie-satysfakcjonująca. Aby zredukować złożoność obliczeniową stosuje się metody wyszukiwania sąsiadów.

Wykorzystywane w symulacji jądra wygładzające posiadają tzw. promień odcięcia, który ogranicza zasięg w jakim poszukiwani są sąsiedzi brani pod uwagę przy obliczaniu parametrów pojedynczej cząstki. Wiedząc, że cząstka wykorzystuje jedynie małą część ze wszystkich cząstek do wyliczenia swoich parametrów, można znacznie skrócić czas obliczeń poprzez zacieśnienie zakresu poszukiwań sąsiadów. W takim wypadku asymptotyczna złożoność obliczeniowa redukowana jest z $O(n^2)$ do $O(mn)$, gdzie m oznacza średnią liczbę sąsiadów cząstki. Ta liczba często nie przekracza wartości 30-40.

Jednym z typów struktur, które pozwalają poszeregować cząsteczki na grupy sąsiadów jest jednorodna siatka (ang. uniform grid). Cały obszar, w którym znajdują się cząsteczki - i mogą się znaleźć podczas przebiegu symulacji - dzielony jest na komórki. Obszar ten nie zmienia się przez cały czas trwania symulacji więc istotne jest aby cząsteczki nie miały szansy z niego ‘ucieć’. W przeciwnym wypadku algorytm nie będzie w stanie poprawnie znaleźć sąsiadów dla ‘uciekierów’. Przy obliczaniu parametrów cząsteczek znajdujących się w komórce X iteruje się tylko po tych cząsteczkach, które znajdują się w 26 komórkach sąsiadujących z komórką X . Rozmiar komórek również jest stały i jest równy najbliższej potęgce dwójki większej od promienia odcięcia jądra wygładzającego. Przyjęcie takiego rozmiaru komórki zapewnia, że sąsiadujące komórki zawierają wszystkie cząsteczki wymagane do przeprowadzenia poprawnych obliczeń.

Każdej komórce może zostać przypisany indeks, który oblicza się dla cząsteczki według poniższego wzoru, na podstawie jej położenia:

$$c_{idx} = k + l * K + m * K * L \quad (34)$$

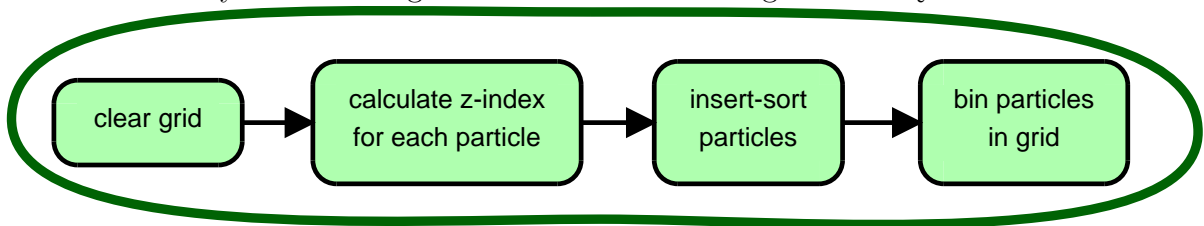
gdzie K i L to rozmiary obszaru symulacji w kierunkach x i y , a $[k, l, m]$ to położenie komórki w globalnym układzie współrzędnych. Wartości K , L i M (rozmiar obszaru symulacji w kierunku z), aby zapobiec wprowadzaniu niedokładności związanych z zaokrągleniami liczb przy dzieleniu, są ustawiane jako wielokrotności liczby 2. «Implementacja funkcji w dodatku/listingu xxx».

Aby zachować optymalne ułożenie danych w pamięci podręcznej, tablica przechowująca cząsteczki jest sortowana ze względu na indeks komórek w której znajdują się cząsteczki - czyli pośrednio ze względu na ich położenie w obszarze symulacji. Dzięki temu cząsteczki znajdujące się w jednej komórce ułożone są blisko siebie również w pamięci. Nie oznacza to jednak, że cząsteczki z sąsiadujących komórek - które również biorą udział w obliczeniach - będą znajdowały się również niedaleko w pamięci. Aby zwiększyć wykorzystanie danych znajdujących się w pamięci podręcznej (ang. cache-hit) wykorzystywane jest kodowanie Mortona/Krzywa-Z (ang. Morton code/Z-Curve) [Wik15c]. Indeksowanie komórek i sortowanie cząsteczek według tej krzywej pozwala na ułożenie cząsteczek znajdujących się w sąsiadujących komórkach bliżej siebie w pamięci.

Posortowanie cząsteczek pozwala również na uproszczenie struktury komórki, która przy takich założeniach przechowuje jedynie **adres pierwszej cząsteczki** w posortowanej tablicy i **liczbę cząsteczek** znajdujących się w danej komórce.

Cały opisany powyżej proces optymalizacji wyszukiwania sąsiadów przedstawiony jest na poniższym schemacie:

Rysunek 3: Diagram indeksowania i szeregowania cząsteczek



5.3 Całkowanie numeryczne

5.3.1 Krok czasowy

5.4 Parametry symulacji

Płyn symulowany metodą SPH można «formułować» nadając wartości kilku stałych opisujących ten płyn. Niektórych wartości parametrów nie można jednak przepisać bezpośrednio z tablic stałych fizycznych, ponieważ w «obszarze» na jakim przeprowadzana jest symulacja nie mają takiego sensu jak w rzeczywistym świecie i nie «powodują/inicjują» spodziewanych efektów. Ich wielkości muszą być także dostrajane z myślą o stabilności symulacji, na którą mają duży wpływ. Powyższe kwestie sprawiają, że ustawianie stałych płynu - przynajmniej na początkowych etapach implementacji symulatora - jest trudne i odbywa się metodą prób i błędów. Poniżej umieszczone zostały przykładowe wartości tych parametrów, które dobrze sprawdzają się dla symulacji płynów skonstruowanym algorytmem oraz wskazówki jak dobierać te wielkości.

5.4.1 Objętość symulowanego płynu oraz masa pojedynczej cząsteczki

Masa pojedynczej cząsteczki wpływa na wielkość sił powstających przy symulacji. Ten parametr jest «dosyć» czuły i istotnie wpływa na charakterystykę symulacji jak i jej stabilność. Jego dopasowanie powinno przebiegać równolegle z korektą przede wszystkim kroku czasowego oraz promienia odcięcia jądra wygładzającego.

5.4.2 Promień odcięcia jądra wygładzającego

Promień odcięcia h jest wartością mającą duży wpływ na stabilność i wydajność symulacji oraz jakość odwzorowania płynu. Określa on ‘zasięg widzenia’ pojedynczej cząsteczki, w którym to szuka swoich sąsiadów - cząsteczek wykorzystywanych do obliczeń parametrów tej cząsteczki. Im więcej sąsiadów, tym zoptymalizowany algorytm ich przeszukiwania (5.2.1) osiąga gorszą złożoność, która z powrotem dąży do $O(n^2)$. Mylnym jest pojęcie o zwiększaniu dokładności symulacji wraz ze zwiększeniem promienia odcięcia. Od pewnego momentu zwiększanie tego parametru powoduje nienaturalne zachowanie się płynu. Siła ciśnienia, która jest odpowiedzialna za utrzymywanie odpowiednich odległości pomiędzy cząsteczkami staje się zbyt duża i cząsteczki zlepiają się nienaturalnie.

Promień odcięcia może być również zbyt mały. W takim wypadku zbyt mała liczba cząstek jest brana pod uwagę przy obliczaniu parametrów i symulacja znowu staje się nienaturalna. Dodatkowo mały promień negatywnie wpływa na stabilność symulacji.

Ustawianie tego parametru, spowodowane przez jego duży wpływ na charakter symulacji, jest trudne. Często sprowadza się do wielokrotnego testowania ustawień tego parametru w parze z innymi, mającymi wpływ na stabilność i wizualną jakość symulacji. W tej implementacji zdecydowano się na dopasowanie promienia odcięcia do wielkości oczka siatki przechowującej sąsiadów cząstki (patrz rozdział (5.2.1)) tak aby te dwie wielkości były jak najbardziej zbliżone. Zostało zbadane, że taki stosunek tych wielkości jest optymalny ze względu na wydajność

przeszukiwania sąsiadów [Ihm+14].

5.4.3 Gęstość spoczynkowa

Równanie (18) wykorzystuje gęstość spoczynkową do obliczenia ciśnienia dla pojedynczej cząsteczki. Mała wartość tego parametru powoduje, że siła ciśnienia pomiędzy cząsteczkami wzajemnie je odpycha. Odpowiednio go korygując powodujemy zmniejszenie się tej siły odpychającej. W pewnym momencie siła osiąga wartości ujemne i cząsteczki zaczynają się przyciągać. Zbyt duża wartość prowadzi do szybkiego wzrostu wartości ciśnienia co powoduje eksplozję numeryczną.

5.4.4 Lepkość

Stała występująca w równaniu (22) na siłę lepkości dostosowywana jest ze względu na pożądany charakter symulowanego płynu oraz efekt ‘wygładzający’, który zapobiega nagłym wzrostom prędkości cząsteczek i stabilizuje symulację.

5.4.5 Napięcie powierzchniowe

W równaniu (27) występuje stała σ modulująca siłę przyciągania do siebie cząstek znajdujących się na powierzchni płynu. Jej większa wartość powoduje zwiększenie się tej siły. Wartość progu l funkcjonującego w równaniu (28) ma jedynie ustrzec przed dzieleniem przez bardzo małe wartości co negatywnie wpływa na stabilność symulacji. Wartość progu powinna być niska aby zachować naturalność działania siły napięcia powierzchniowego.

6 Wyniki

6.1 Prezentacja symulacji «foto»

6.2 Analiza wydajności

6.3 Analiza stabilności numerycznej

7 Podsumowanie, wnioski i dalszy rozwój

W tej pracy przedstawiona została metoda symulacji płynów Smoothed Particle Hydrodynamics. Została ona wykorzystana do dyskretyzacji równania Naviera-Stokesa dla mechaniki płynów. W symulacji użyto opisu płynu metodą Lagrange’a wykorzystującego cząsteczki. Przedstawione zostały oddziaływania charakteryzujące mechanikę cieczy: ciśnienie, lepkość, napięcie powierzchniowe.

Do przyśpieszenia obliczeń wykorzystano mechanizm wyszukiwania sąsiadów bazujący na jednorodnej siatce. Siatka ta dzieli obszar symulacji i jednocześnie wyznacza możliwych sąsiadów poszczególnych cząsteczek. Aby zoptymalizować dostęp do pamięci (zmniejszyć ilość błędnych odniesień do pamięci cache, ang. cache miss), uwzględniono sortowanie cząsteczek według krzywej mortona (ang. Z-curve).

Zaimplementowano sposób wizualizacji symulowanych cząsteczek bazujący na trójwymiarowej mapie odległości (ang. distance field) opisującej kształt powierzchni płynu. Po zbudowaniu mapy przeprowadzany jest na karcie graficznej proces ray castingu.

Opisana metoda pozwala na przeprowadzenie symulacji i wizualizacji kilku tysięcy cząstek do użytku w czasie rzeczywistym (dla 4000 cząstek jedna iteracja trwa 160 ms na testowej konfiguracji). Istnieje duże pole do optymalizacji, głównie w zakresie poprawienia lokalności danych. Duży wzrost wydajności - przy odpowiedniej implementacji o rząd wielkości - przyniesie wykorzystanie GPU do przetwarzania symulacji.

Zbadano jaki wpływ na stabilność symulacji mają parametry: lepkość i krok czasowy.

Porównano jakościowe oraz wydajnościowe wyniki wizualizacji w zależności od gęstości próbkowania płynu na mapie odległości.

Bibliografia

- [Gos+10] Prashant Goswami et al. “Interactive SPH Simulation and Rendering on the GPU”. In: (2010).
- [Had+05] Markus Hadwiger et al. “Real-Time Ray-Casting and Advanced Shading of Discrete Isosurfaces”. In: (2005).
- [Haf15] Andreas Haferburg. *Object Order Empty Space Skipping in OpenGL 4*. [Online; dostęp 30 Października 2015]. 2015. URL: <http://haferburg.github.io/>.
- [Ihm+10] Markus Ihmsen et al. “Boundary handling and adaptive time-stepping for PCI-SPH”. In: (2010).
- [Ihm+14] Markus Ihmsen et al. “SPH Fluids in Computer Graphics – State of the Art Report”. In: (2014).
- [Ihm13] Markus Ihmsen. “Particle-based Simulation of Large Bodies of Water with Bubbles, Spray and Foam”. In: (2013).
- [Kel06] Micky Kelager. “Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics”. In: (2006).
- [MCG03] Matthias Müller, David Charypar, and Markus Gross. “Particle-Based Fluid Simulation for Interactive Applications”. In: (2003).
- [Wik15a] Wikipedia. *Operator Stokesa* — *Wikipedia, The Free Encyclopedia*. [Online; dostęp 19 Listopada 2015]. 2015. URL: https://pl.wikipedia.org/wiki/Operator_Stokesa.
- [Wik15b] Wikipedia. *Verlet integration* — *Wikipedia, The Free Encyclopedia*. [Online; dostęp 22 Września 2015]. 2015. URL: https://en.wikipedia.org/wiki/Verlet_integration.
- [Wik15c] Wikipedia. *Z-order curve* — *Wikipedia, The Free Encyclopedia*. [Online; dostęp 7 Kwietnia 2015]. 2015. URL: https://en.wikipedia.org/wiki/Z-order_curve.