

UNIwersytet Wrocławski  
Wydział Fizyki i Astronomii  
Instytut Fizyki Teoretycznej



Wiktoria Saramak

Rozwiązanie problemu przepływu  
w zamkniętej komorze metodą gazu sieciowego

Solution of the Driven-Cavity Flow  
using Lattice-Gas Cellular Automata

Opiekun  
dr Maciej Matyka

Wrocław 2016

## Streszczenie

Celem niniejszej pracy jest rozwiązanie problemu przepływu przez zamkniętą komorę za pomocą metody gazu sieciowego oraz szersze zapoznanie się z wykorzystanym modelem i jego implementacją.

Na początku omawiam modele gazów sieciowych, ich ewolucję oraz wartości makroskopowe. Przedstawiam również sposób implementacji modelu FHP-III, będący najbardziej rozbudowanym wariantem modelu FHP. Następnie poruszam problem przepływu w komorze. Przedstawiam schemat zagadnienia, wyjaśniam czym jest liczba Reynoldsa oraz uwzględniam wszystkie parametry symulacji. Na koniec wyniki zostały skonfrontowane z rozwiązaniami przepływu przez komorę metodą wielosiatkową (multigrid method) opracowane przez grupę badawczą z University of Cincinnati. Całość podsumowuję analizą i dyskusją na temat otrzymanych wyników.

W dodatku A znajdują się wyniki testów poprawności modelu FHP-III, natomiast w dodatku B fragmenty kodu przeprowadzonej symulacji.

# Abstract

The aim of this study is to solve the driven-cavity flow problem using the lattice-gas cellular automata and to get acquainted with the lattice-gas models.

In the first part, the lattice-gas cellular automata models, their evolution and macroscopic properties are presented. That leads to the most advanced FHP model and its implementation. Next, the driven-cavity problem is described in the chapter two. It contains the Reynolds number explanation and all the parameters of simulation. Finally, the last part of the study validates the correctness of the simulation where results are compared with the other study (U.Ghia, K.N. Ghia, C.T. Shin, University of Cincinnati) that uses Multigrid method for driven-cavity problem. In addition, the comparison is summed up with a short discussion about the results.

Appendix A contains the correctness tests of the FHP - III model and Appendix B contains the partial code of the simulation.

# Spis treści

|          |                                         |           |
|----------|-----------------------------------------|-----------|
| <b>1</b> | <b>Modele gazów sieciowych</b>          | <b>5</b>  |
| 1.1      | Model HPP . . . . .                     | 6         |
| 1.2      | Model FHP . . . . .                     | 7         |
| 1.3      | Implementacja modelu FHP-III . . . . .  | 8         |
| 1.4      | Wielkości makroskopowe . . . . .        | 10        |
| <b>2</b> | <b>Przepływ w zamkniętej komorze</b>    | <b>11</b> |
| 2.1      | Opis zagadnienia . . . . .              | 11        |
| 2.2      | Liczba Reynoldsa . . . . .              | 12        |
| 2.3      | Implementacja . . . . .                 | 13        |
| <b>3</b> | <b>Analiza wyników</b>                  | <b>16</b> |
| 3.1      | Wyniki . . . . .                        | 16        |
| 3.2      | Dodatkowe uwagi . . . . .               | 21        |
| <b>A</b> | <b>Testy poprawności modelu FHP-III</b> | <b>23</b> |
| A.1      | Rozchodząca się fala . . . . .          | 23        |
| A.2      | Przepływ Poiseuille’a . . . . .         | 24        |
| <b>B</b> | <b>Fragmenty kodu symulacji</b>         | <b>27</b> |

# Wstęp

W mechanice płynów wyróżniamy kilka działów, w szczególności obliczeniową dynamikę płynów, która zajmuje się rozwiązaniami numerycznymi zagadnień związanych z przepływami. Przy wykorzystaniu odpowiednich modeli i algorytmów możliwe jest przybliżone wyznaczenie rozkładu prędkości, ciśnienia i innych parametrów przepływów. Podstawową i najczęściej wykorzystywaną zależnością opisującą wpływ sił na ruch płynu są równania Naviera-Stokesa (N-S). Jest to układ cząstkowych, nieliniowych równań różniczkowych. Rozwiązanie go stanowi jeden z najważniejszych problemów matematycznych, ogłoszony przez Instytut Clay jako problem Millenium [1]. W praktyce, równania N-S dyskretyzowane są m.in. za pomocą metody elementów skończonych, metody różnic skończonych czy metody objętości skończonych. Jednym z koncepcyjnie najprostszych podejść do rozwiązania problemów transportu przepływów, jest wykorzystanie modelu gazu sieciowego (Lattice Gas Cellular Automata) [2]. Polega on na symulacji pojedynczych cząsteczek ułożonych na regularnej sieci, które poruszają się zgodnie z zasadami automatów komórkowych.

## 1 Modele gazów sieciowych

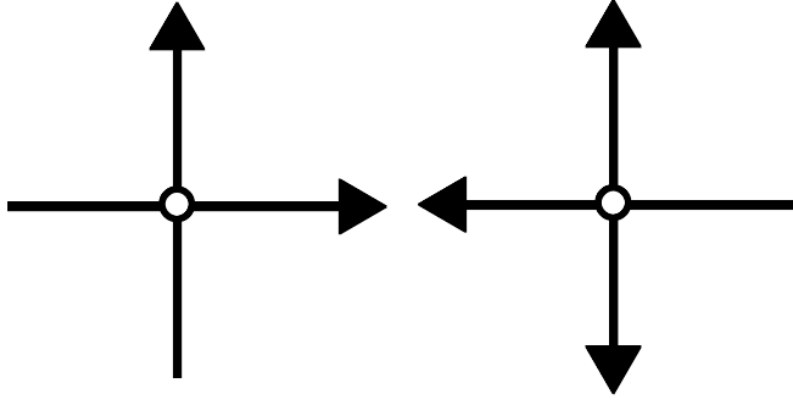
Automaty komórkowe to modele matematyczne opisane przez regularną, dyskretną siatkę komórek, gdzie każda komórka przyjmuje jeden stan ze skończonego zbioru stanów. W każdym kroku czasowym stan każdej komórki ewoluuje według ściśle określonych reguł, co następuje równolegle dla wszystkich komórek. Reguły te zależą wyłącznie od poprzedniego stanu komórki oraz skończonego zbioru jej sąsiadów. Najbardziej popularnym i jednym z pierwszych automatów komórkowych jest Gra w życie wymyślona w 1970 roku przez brytyjskiego matematyka Johna Conwaya [3]. Automaty komórkowe wykorzystywane do symulacji zagadnień hydrodynamicznych nazywamy gazami sieciowymi. Zgodnie z definicją, w gazach sieciowych wyróżniamy reguły, według których układ zmienia się w czasie. Można je ująć w postaci trzech prostych zasad:

- Z jednej komórki do następnej może przemieścić się tylko jedna cząstka.
- Każdy krok czasowy składa się z dwóch etapów: kolizji i przemieszczenia.
- Każda cząstka może poruszać się tylko w jednym kierunku z dyskretnego zbioru kierunków.

Model HPP to pierwszy model gazu sieciowego opublikowany w 1973 roku i nazwany pierwszymi literami nazwisk jego twórców (Hardy, Y. Pomeau, de Pazzis) [2]. Ten dwuwymiarowy model opisany jest na sieci kwadratowej i reprezentuje interakcję cząstek płynu. Model wykazuje jednak wysoką anizotropowość [4]. W 1986 roku U. Frisch, B. Hasslacher i Y. Pomeau zaproponowali zmianę siatki kwadratowej na trójkątną tworząc przy tym nowy model gazu sieciowego, nazwany modelem FHP [5].

## 1.1 Model HPP

Model HPP opisany jest na sieci kwadratowej, gdzie wyróżniamy cztery możliwe kierunki poruszających się cząstek. Przykładowe stany węzła przedstawia rysunek 1.

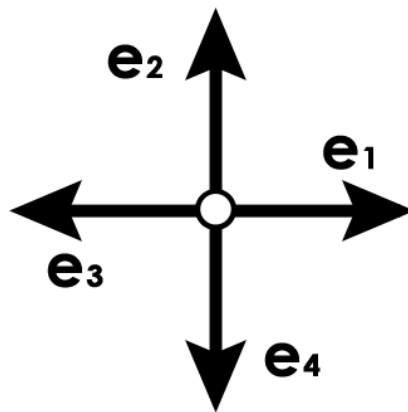


Rysunek 1: HPP - przykładowy stan węzła. Biały punkt reprezentuje węzeł, czarny punkt poruszającą się cząstkę w określonym kierunku (strzałki).

Dyskretne kierunki cząstek przedstawia się w postaci wektorów  $e$  i zapisuje w postaci:

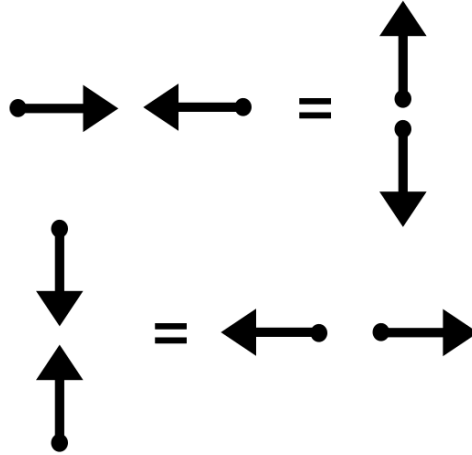
$$e_i = \left( \sin \left( \frac{\pi}{2} i \right), \cos \left( \frac{\pi}{2} i \right) \right) \quad i = 1, 2, 3, 4. \quad (1)$$

Wektory  $e$  ułożone są według schematu na rysunku 2:



Rysunek 2: HPP - wektory cząstek.

Sytuacje, w których cząstka poruszająca się np. w prawo napotka cząstkę poruszającą się w lewo nazywamy kolizją. W wyniku takich zderzeń cząstki odbijają się o 90 stopni, co w modelu HPP możliwe jest tylko w dwóch przypadkach.

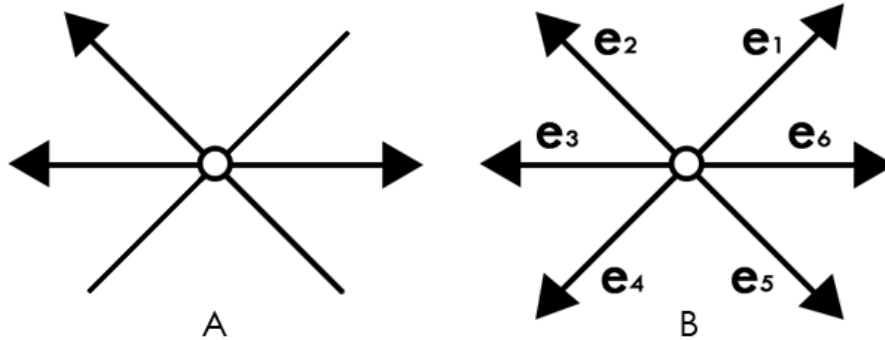


Rysunek 3: HPP - możliwe kolizje międzycząsteczkowe.

Model zachowuje zasadę zachowania pędu, masy oraz symetrii. Nie spełnia jednak kilku ważnych fizycznych własności. Na przykład brakuje tu tzw. niezmienności rotacyjnej, co powoduje, że wiry i rozchodzące się fale mają kształt kwadratowy [4].

## 1.2 Model FHP

W porównaniu do swojego poprzednika, model FHP reprezentuje interakcję między cząstkami na siatce trójkątnej. Wówczas dyskretny zbiór możliwych kierunków cząstek powiększa się o dwa dodatkowe, a co za tym idzie istnieje więcej przypadków kolizji międzycząsteczkowych. Odbicia realizowane są w formie obrotów o 60 stopni.

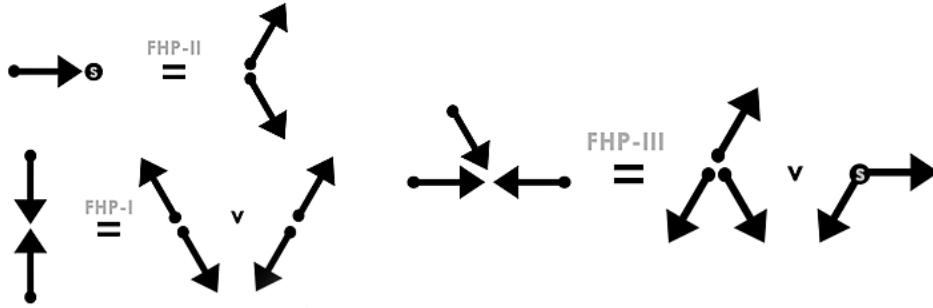


Rysunek 4: FHP - przykładowy stan węzła (A) oraz wektory cząstek (B).

Podobnie jak w modelu HPP, dyskretne kierunki cząstek można zapisać w postaci wektorów:

$$e_i = \left( \cos \left( \frac{\pi}{3} i \right), \sin \left( \frac{\pi}{3} i \right) \right) \quad i = 1, 2, \dots, 6. \quad (2)$$

W modelu FHP wyróżniamy trzy warianty: FHP-I, FHP-II i FHP-III. Pierwszy z nich nie przewiduje nieporuszającej się cząstki i wykazuje największą lepkość w stosunku do pozostałych. Warianty FHP-II i FHP-III zakładają istnienie cząstki statycznej, natomiast różnią się pomiędzy sobą ilością możliwych kolizji międzycząsteczkowych. Poniższa ilustracja przedstawia kilka ich przykładów w zależności od wariantu modelu.

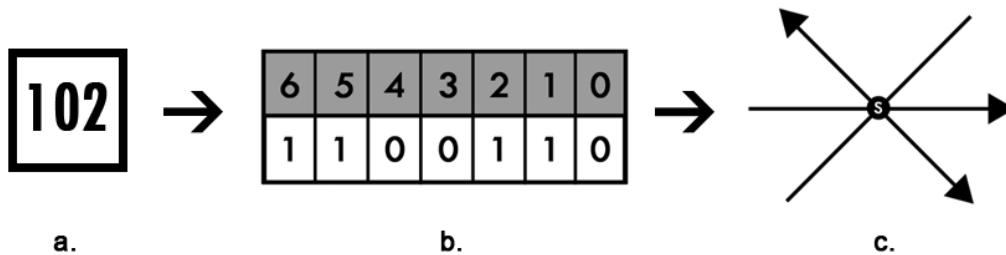


Rysunek 5: FHP - Przykłady kolizji międzycząsteczkowych

### 1.3 Implementacja modelu FHP-III

Wiele testów numerycznych udowadnia, że FHP-III wykazuje najlepsze własności na tle pozostałych wariantów modelu FHP [6]. Z tego względu w niniejszej pracy skupiono się na jego implementacji.

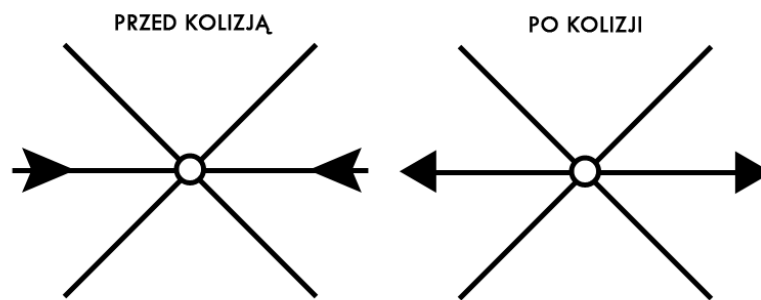
Model FHP opisany jest na sieci trójkątnej. Wyróżniamy tam regularnie ułożone węzły, gdzie w przypadku modelu FHP-III każdy z nich może przechować maksymalnie 7 cząstek. Stan każdego z węzłów można zapisać za pomocą ciągu 7 bitów gdzie każdy z nich oznacza czy cząstka zmierza z danego kierunku (1) czy nie (0). W ten sposób do reprezentacji sieci można wykorzystać dwuwymiarową tablicę, gdzie każda jej komórka przechowuje liczbę całkowitą o wartości, której reprezentacja binarna odpowiada stanowi węzła. W modelu FHP-III istnieje 128 możliwych stanów [8].



Rysunek 6: FHP-III - implementacja węzła, gdzie: a. zapis w komórce tablicy, b. zapis w postaci binarnej, c. postać węzła.

Taka implementacja umożliwia zapis interakcji między cząstkami za pomocą operatorów logicznych. Na przykład, przesunięcie bitowe w prawo lub w lewo o 3 bity umożliwia odbicie cząstki bez poślizgu [7]. Jest to jeden ze sposobów realizacji odbić ze ściankami i przejawia się odbiciem wektora prędkości cząstki o 180 stopni. Wyróżniamy też odbicie z poślizgiem, gdzie cząstka nie spowalnia na powierzchni przeszkody [8]. Innym przykładem wykorzystania operatorów logicznych, to przemieszczenie cząstek po sieci, który realizowany jest przy wykorzystaniu koniunkcji, która umożliwia sprawdzenie, czy dana cząstka znajduje się w danym węźle. Następnie, cząstka wpisywana jest w odpowiedni sąsiedni węzeł wskazywany przez wektor cząstki.

Istotne jest rozróżnienie etapów przed i po kolizji. Przed kolizją cząstki wychodzą z węzła zgodnie z wektorami sieci. Kiedy spotykają się z innymi cząstkami z sąsiednich węzłów, zmierzają do węzła i może nastąpić kolizja.



Rysunek 7: FHP-III - schemat kolizji.

Następuje wtedy zmiana węzła według ustalonych zasad kolizji. Najprościej jest przechowywać je w osobnej tablicy pod postacią liczb całkowitych, gdzie każda z nich reprezentuje stan po kolizji. Model FHP-III posiada najbardziej rozbudowany możliwy ich zbiór a w tym kolizje czterocząstkowe. To dzięki temu wykazuje najmniejszą lepkość w porównaniu do innych wariantów modelu FHP.

Biorąc pod uwagę wspomniane wcześniej reguły obowiązujące w gazach sieciowych oraz poszczególne etapy interakcji międzycząsteczkowych, symulacja modelu FHP przebiega według algorytmu:

1. *Ustaw stan początkowy sieci.*
2. *Sprawdź czy występuje kolizja. Jeżeli tak – postępuj zgodnie z regułą kolizji. Jeżeli nie – idź do punktu 3.*
3. *Przesuń cząstki zgodnie z ich wektorami prędkości.*
4. *Wróć do punktu 2.*

## 1.4 Wielkości makroskopowe

Modele HPP i FHP zachowują masę i pęd. Poniższy wzór ilustruje również zachowanie symetrii:

$$\sum_{i=1}^z e_i = 0, \quad (3)$$

gdzie  $z$  jest liczbą wektorów sieci.

Lokalna prędkość zdefiniowana jest przez sumę iloczynów wszystkich wektorów sieci i liczby  $n$  będącej informacją o obsadzeniu danego kierunku przez cząstkę. Liczba  $n$  przyjmuje wartości 1 lub 0 w zależności od obecności cząstki poruszającej się z danego kierunku sieci:

$$\vec{v} = \frac{1}{\rho} \sum_{i=1}^z e_i n_i. \quad (4)$$

Lokalna gęstość  $\rho$  odpowiada sumie liczb cząstek znajdujących się we wskazanych kierunkach przez wektory sieci  $e_i$ . W przypadku rozpatrywania większych obszarów jest to stosunek sumy cząstek poruszających się w tym obszarze do ich maksymalnej ilości:

$$\rho = \sum_{i=1}^z n_i. \quad (5)$$

Wyróżniamy również wielkość zwaną gęstością na kierunek  $d$ , będącą stosunkiem lokalnej gęstości do liczby wektorów sieci:

$$d = \frac{\rho}{z}. \quad (6)$$

Przedstawione modele gazu sieciowego charakteryzują się wysoką obecnością szumu statystycznego w rozwiązaniach. Wpływa to znacznie na wyniki obliczeń badanych zagadnień. Zwykle stosuje się metodę uśredniania wyników w czasie i lokalnej przestrzeni, co przejawia się ich istotną poprawą [12].

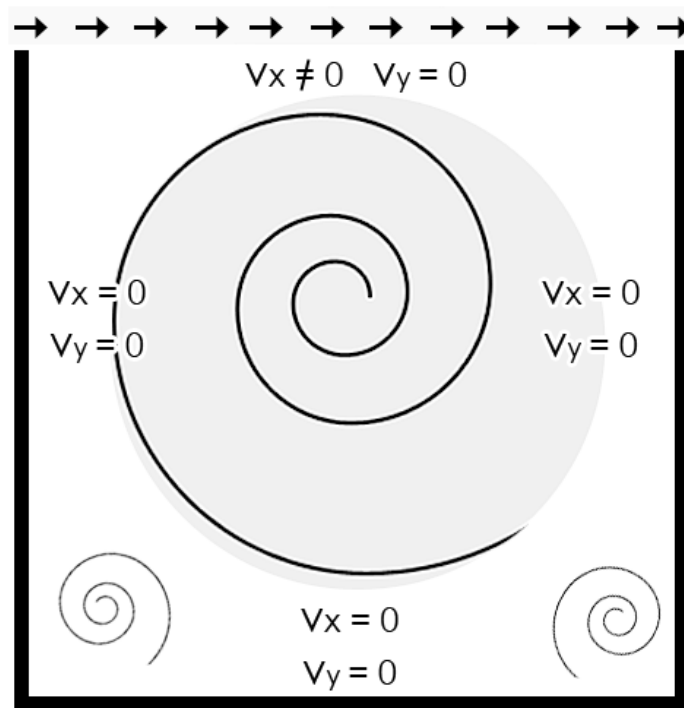
## 2 Przepływ w zamkniętej komorze

Przepływ w zamkniętej komorze, zwykle bardziej znany pod angielskim tłumaczeniem *Lid-Driven Cavity*, jest jednym z najczęściej analizowanych problemów obliczeniowej dynamiki płynów. Powód, dla którego jest on chętnie wykorzystywany to m.in. proste warunki brzegowe i geometria układu. Ponadto, przy odpowiedniej wizualizacji można uzyskać bardzo interesujące rezultaty. Obecnie pojawiają się jego różne warianty, głównie różniące się geometrią komory np. trójkąt lub trapez [9]. W ramach tej pracy wykorzystano najbardziej klasyczny przypadek komory kwadratowej.

### 2.1 Opis zagadnienia

Rozważany problem dotyczy nieściśliwego przepływu przez kwadratową komorę, gdzie górna pokrywa porusza się zadaną prędkością. Pozostałe ścianki naczynia nie poruszają się a cząstki, w wyniku zderzeń odbijają się bez poślizgu. Ruch górnej pokrywy wpływa na ruch cząstek znajdujących się w naczyniu, które zaczynają podążać za pokrywą i odbijać od ścianek, co w efekcie powoduje tworzenie się charakterystycznego wiru w komorze. Gdy płyn uzyskuje dużą prędkość, mniejsze wiry zaczynają pojawiać się w rogach komory.

Zagadnienie zostało zilustrowane na rysunku 8.



Rysunek 8: Schemat przepływu w komorze.

## 2.2 Liczba Reynoldsa

Liczbę Reynoldsa ( $Re$ ) wyrażamy jako zależność między wymiarem charakterystycznym układu, prędkością charakterystyczną płynu oraz lepkością kinematyczną, co definiuje wzór 7.

$$Re = \frac{l \cdot \vec{v}}{\nu}. \quad (7)$$

W obrębie prezentowanego zagadnienia, prędkość charakterystyczną  $\vec{v}$  rozumiemy jako prędkość górnej pokrywy. Wymiar charakterystyczny  $l$  odnosi się do wymiaru, który ma bezpośredni wpływ na stateczność ruchu płynu, natomiast lepkość kinetyczna  $\nu$  jest parametrem fizycznym określonym jednoznacznie, niezależnie od ruchu płynu. W przypadku modelu FHP-III jest to wielkość zdefiniowana według wzoru [12]:

$$\nu = \frac{1}{28} \frac{1}{(d(1-d))} \frac{1}{1-8d(1-d)/7} - \frac{1}{8}, \quad (8)$$

gdzie wielkość  $d$  jest gęstością na kierunek, określaną jako:

$$d = \frac{\rho_c}{z}, \quad (9)$$

a  $\rho_c$  rozumiemy jako gęstość w całym układzie [6] [12].

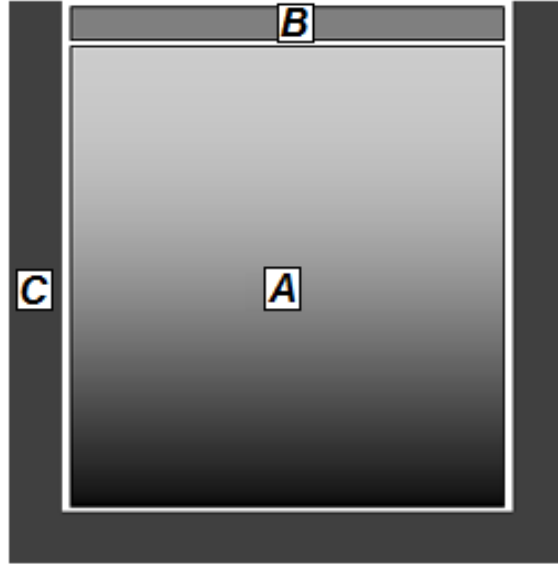
Liczba  $Re$  stanowi kryterium stateczności ruchu płynów. Jest to bezwymiarowa liczba, wyrażająca równowagę pomiędzy siłami lepkości, a siłami bezwładności. Niska prędkość i wysoka lepkość charakteryzują się małą wartością  $Re$ . Wtedy przepływ jest laminarny, ponieważ przeważają siły lepkości. Gdy wartość  $Re$  jest niższa niż 1, mówimy o przepływie Stokes'a, lub przepływie pełzającym (ang. creeping flow) [10]. Z takimi przepływami mamy do czynienia w ośrodkach porowatych. W przypadku większej wartości prędkości poruszającego się płynu, niższej lepkości i przy większym wymiarze układu, liczba Reynoldsa jest duża. Następuje wtedy przewaga sił bezwładności nad siłami lepkości. Dla bardzo wysokich  $Re$ , przepływ może stać się niestabilny, turbulentny [10].

Dla każdego przepływu istnieje pewna krytyczna wartość liczby Reynoldsa  $Re_{cr}$ , powyżej której przepływ turbulentny jest obserwowany. Jednak przekroczenie jej nie daje na to gwarancji, ponieważ nie istnieje uniwersalna wartość  $Re_{cr}$  dla dowolnego rodzaju przepływu. Zwykle zależy ona od rozpatrywanej sytuacji oraz od wartości, jakie zostały uwzględnione za charakterystyczną prędkość układu, czy jego wymiar. Zatem  $Re_{cr}$  jest raczej ogólnym przybliżeniem, opartym na doświadczeniach.

## 2.3 Implementacja

### *Układ*

Symulację przeprowadziłam na układzie o wymiarach  $129 \times 129j.s.$  (jednostek sieci), zaimplementowanym w sposób przedstawiony na rysunku 9.



Rysunek 9: Schemat układu.

Obszar B stanowił obszar poruszającego się ze stałą prędkością płynu, natomiast obszar A wypełniony był cząstkami z zadaną gęstością. W obszarze C występowały ścianki, od których wektor prędkości cząstek odbijany był o 180 stopni (wariant odbić bez poślizgu).

### *Górna pokrywa*

Górna pokrywa (obszar B) o grubości  $2j.s.$ , stanowiła rezerwuuar przepływającego płynu, którego prędkość wyznaczana była na zasadzie specjalnie dobranych parametrów. Ustaliłam siedem liczb  $p_0, p_1, p_2, \dots, p_6$  stanowiących prawdopodobieństwo wystąpienia pojedynczej cząstki z prędkością w danym kierunku, które dodawałam do górnej pokrywy. Takie rozwiązanie umożliwiło kontrolę prędkości obszaru B i wartości liczby Re.

### *Gęstość*

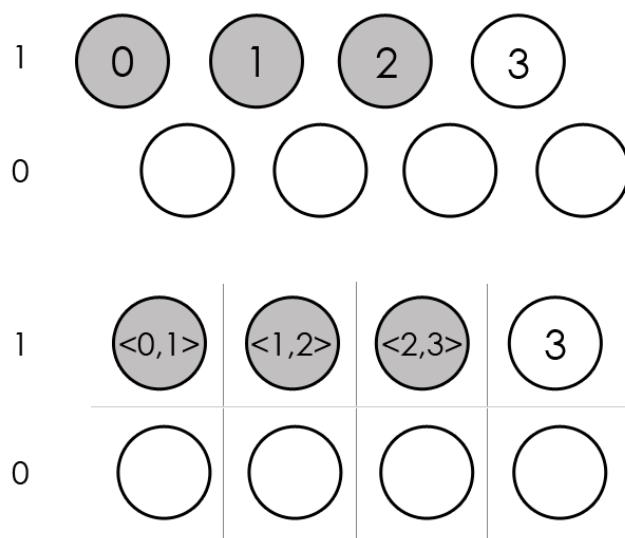
Obszar A został wypełniony pojedynczymi cząstkami, poruszającymi się w jednym z siedmiu możliwych kierunków. Zapewniło to w układzie wartość gęstości na kierunek równą  $d = 1/7$ . Ponieważ w trakcie trwania symulacji dodawano kolejne cząstki do obszaru B z takim samym prawdopodobieństwem dla wszystkich układów, gęstość rośnie i stabilizuje się przy średniej maksymalnej gęstości na kierunek  $d_{max} = 2/7$ .

## *Liczba Reynoldsa*

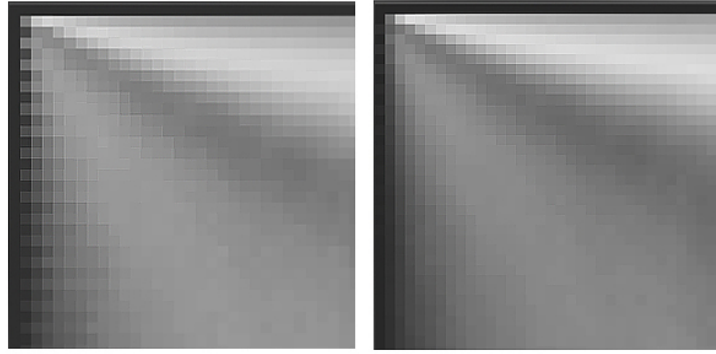
Wartość liczby Re zależy od zmiennych parametrów: gęstość układu i prędkość górnej pokrywy. Dlatego liczono ją w momencie, gdy obie te wielkości ustabilizowały się i zostały uśrednione. Każda pojedyncza symulacja stanowiła odrębny układ o innych parametrach, a co za tym idzie – inną wartością liczby Re.

## *Poprawa wyników*

Wizualizacja symulacji opartych na sieci trójkątnej zależy od sposobu ich prezentacji. Ta sieć posiada jeden wiersz przesunięty względem drugiego. Wykorzystywanie dwuwymiarowych tablic do przechowywania i odczytywania poszczególnych węzłów nie uwzględnia takiego przesunięcia. Wartości w wierszach nie znajdują się idealnie pod sobą, dlatego numerowanie wyników po indeksach tablicy i prezentacja danych zacieraają rzeczywistą geometrię układu. Wobec tego wykorzystałam metodę interpolacji - dla nieparzystych wierszy sieci każdą wartość i jej następnego sąsiada uśredniłam. Dla ostatniego węzła pozostawiłam tę samą wartość, choć można było w tym przypadku skorzystać z ekstrapolacji. Na rysunkach 10, 11 znajdują się schemat metody i uzyskany rezultat.



Rysunek 10: Schemat interpolacji.



Rysunek 11: Efekt uzyskany przed (lewo) i po (prawo) interpolacji.

Ze względu na występowanie szumu, wyniki uśredniłam w czasie, który dla każdej symulacji wynosił 200000 kroków. Jest to niezbędna poprawka w przypadku tego typu modeli. Zaleca się również uśrednianie w przestrzeni [12], jednak w obrębie tej pracy ten krok został pominięty.

### 3 Analiza wyników

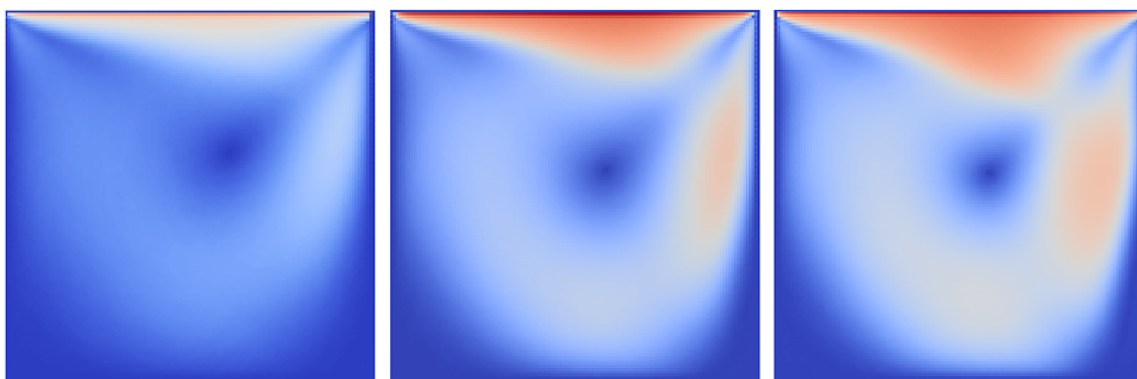
Wiele prac zajmujących się rozwiązaniem problemu przepływu w zamkniętej komorze, odwołuje się do wyników z [13]. Znajdują się tam rozwiązania przy wykorzystaniu metody wielosiatkowej (*multigrid method*). Ze względu na różnorodność i ich precyzyjną analizę, szczególnie pod względem wartości liczby Re, stanowią one dobre kryterium porównawcze. Choć metoda wielosiatkowa znacznie odbiega od wykorzystanego w niniejszej pracy modelu gazu sieciowego, nie przeszkadza to w analizie uzyskanych wyników, które stanowią rozwiązanie tych samych równań różniczkowych N-S.

#### 3.1 Wyniki

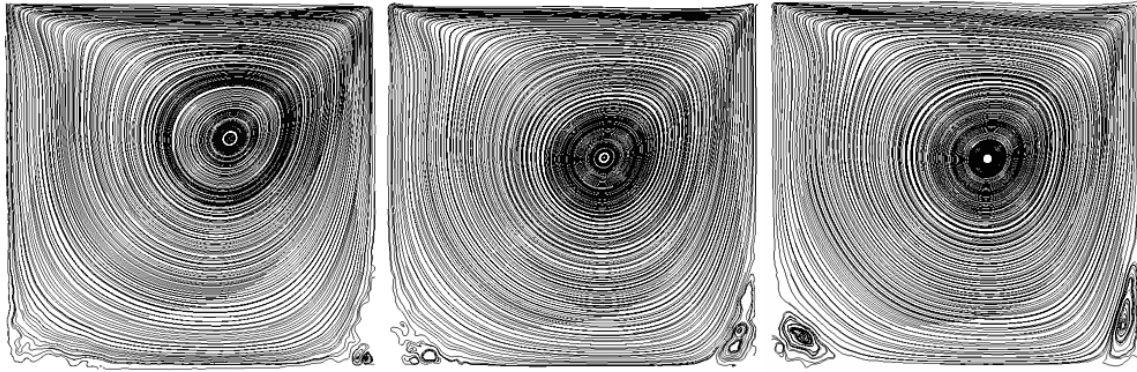
Wyniki opracowałam dla poszczególnych liczb Reynoldsa: 120, 400, oraz 1050 z pomocą programu Paraview ([www.paraview.org](http://www.paraview.org)) i porównałam z wynikami [13] dla Re 100, 400 i 1000. Wszystkie symulacje przeprowadziłam na układzie o wymiarach  $129 \times 129$  a każda z nich trwała 200000 kroków czasowych. Po około 3000 kroku układ osiągał równowagę. Określiłam ją na podstawie zmiany masy w układzie, która zaczęła utrzymywać się przy stałej wartości oraz na podstawie obserwacji prędkości cząstek. Dla pewności, po 5000 kroku nastąpił pomiar potrzebnych wartości. Na koniec wyniki zostały uśrednione w czasie.

Wszystkie węzły sieci poza obszarem górnej pokrywy i ścianek wypełniłam z taką samą gęstością. W każdym węźle umieszczałam cząstkę o losowym wektorze prędkości, co daje w chwili początkowej gęstość na kierunek równą  $d = 1/7$ . Następnie w każdym kroku symulacji dodawałam pojedyncze cząstki w górnej pokrywie w zależności od ustalonego prawdopodobieństwa. Dla każdego układu wartości były dobierane tak, by gęstość w układach była zachowana.

Program paraview umożliwia m.in. wygenerowanie linii prądu, co daje wgląd w zachowanie cząsteczek płynu w symulacji. Poniżej zestawiono rysunki dla wszystkich układów przedstawiające pole prędkości w układzie oraz linie prądu.

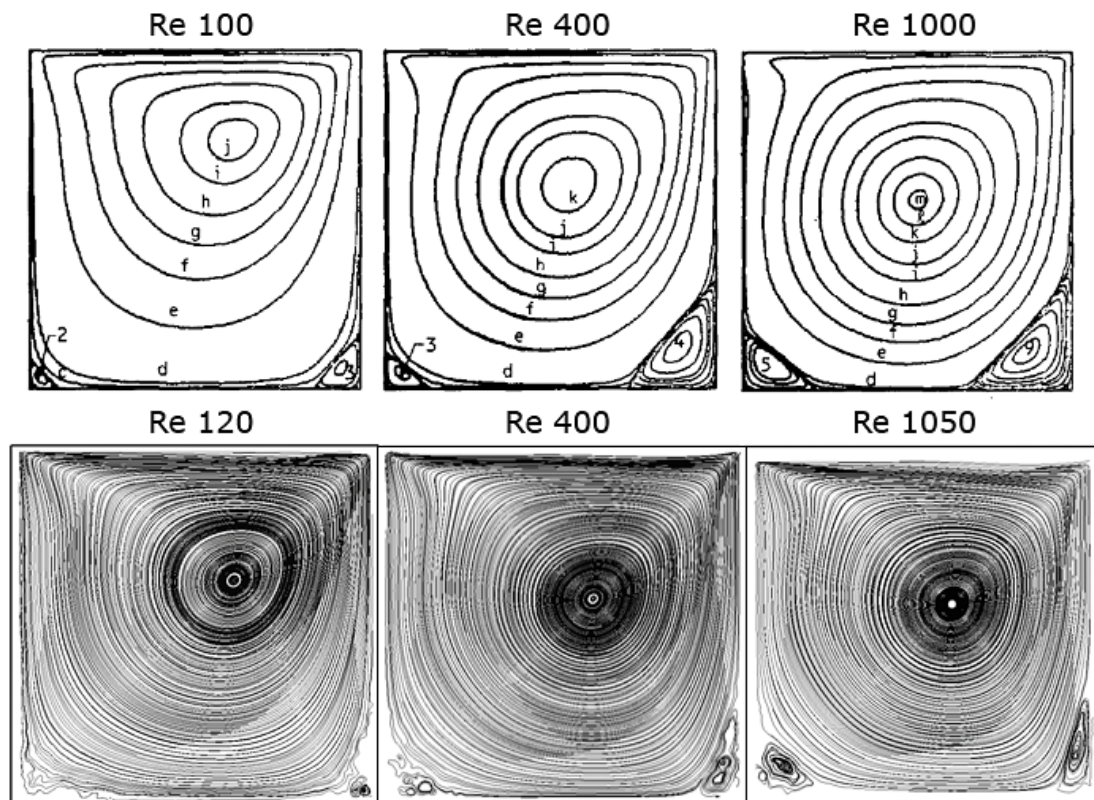


Rysunek 12: Pole prędkości dla  $L=129 \times 129$ . Od lewej Re: 120, 400, 1050.



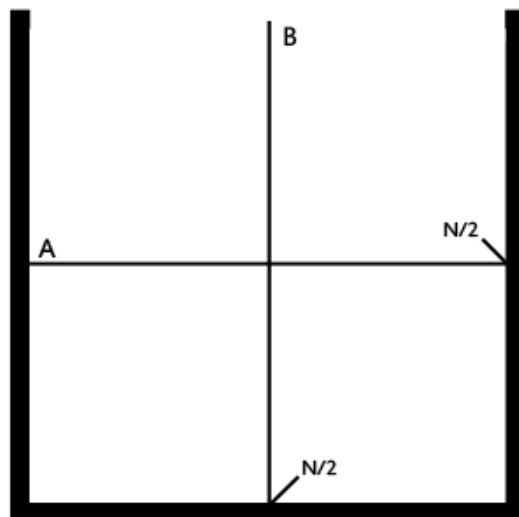
Rysunek 13: Linie prądu dla  $L=129 \times 129$ . Od lewej Re: 120, 400, 1050.

Jedną z najprostszych metod analizy wyników jest porównanie kształtu głównego wiru oraz mniejszych u dołu. Rogi są podatne na ich formowanie się przez intensywny ruch płynu i geometrię układu. Porównałam sposób i kolejność ich kształtowania się z wynikami z [13] i zestawiałam na *Rysunku 14*. Prawy dolny wir jest największy dla wartości Re równej 1050, natomiast lewy róg najdłużej się formuje, co zgadza się z przedstawionym porównaniem.



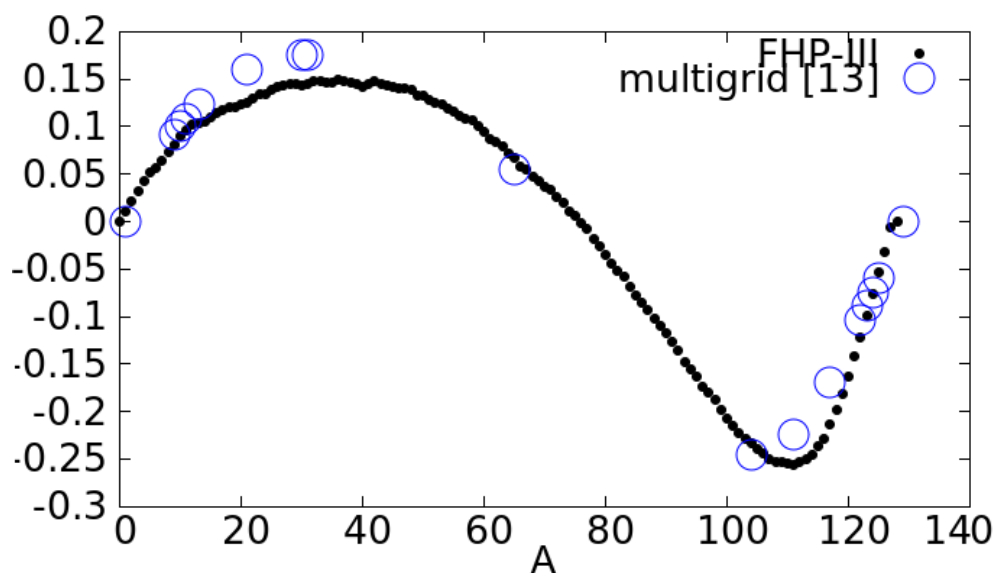
Rysunek 14: Porównanie linii prądu z [13].

Bardziej szczegółową analizę rozwiązań przepływu w komorze stanowi porównanie wykresów przedstawiających prędkości  $u, v$  w centrum geometrycznym komory, gdzie  $u$  jest składową prędkości  $v_x$ , natomiast  $v$  składową  $v_y$ . Prędkość  $v$  liczona była wzdłuż osi A, natomiast prędkość  $u$  wzdłuż osi B, jak na rysunku 15.

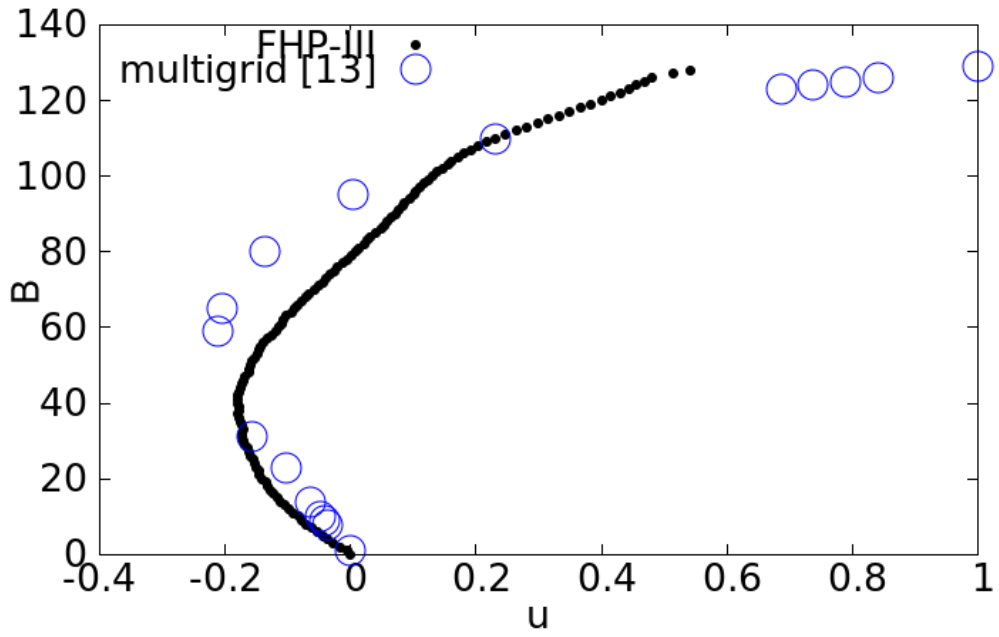


Rysunek 15: Centrum geometryczne komory: układ osi A, B wzdłuż których liczone prędkości  $u, v$ .

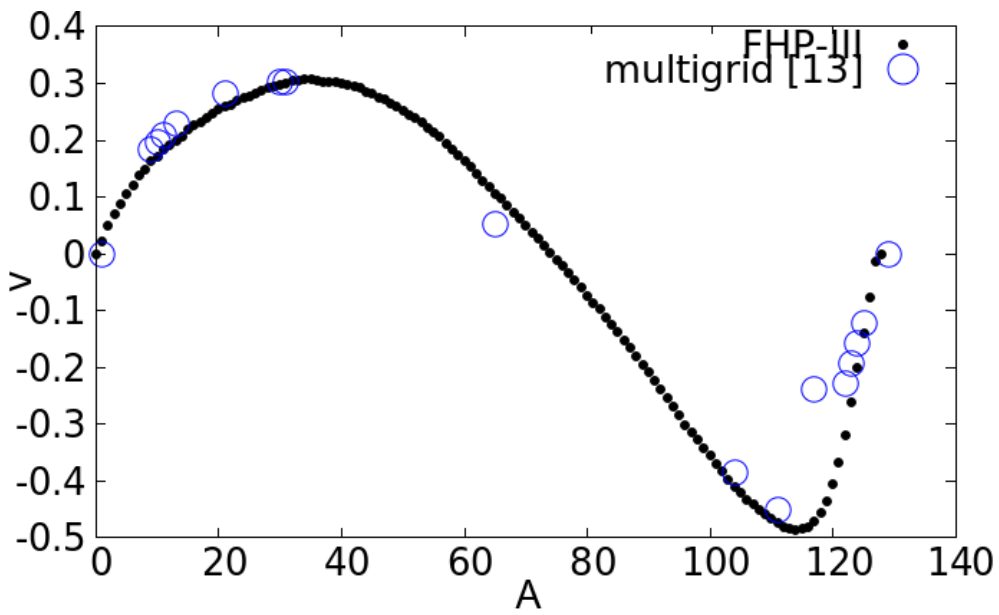
Wartości prędkości  $u$  oraz  $v$  zestawiono z [13] na wykresach zamieszczonych poniżej.



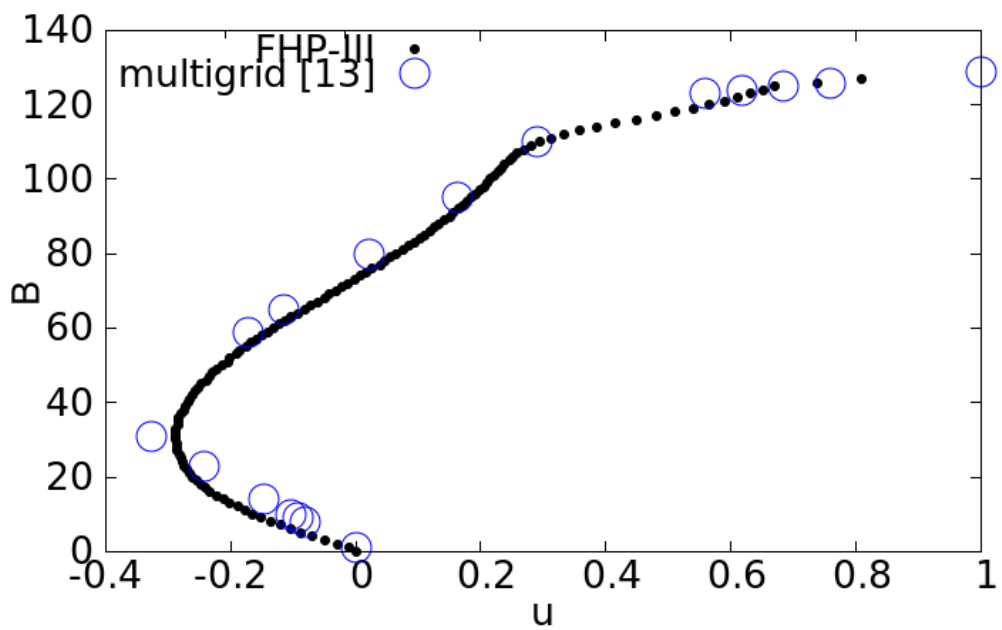
Rysunek 16: Porównanie wartości prędkości  $v$  dla  $Re\ 120$  z [13] ( $Re\ 100$ ).



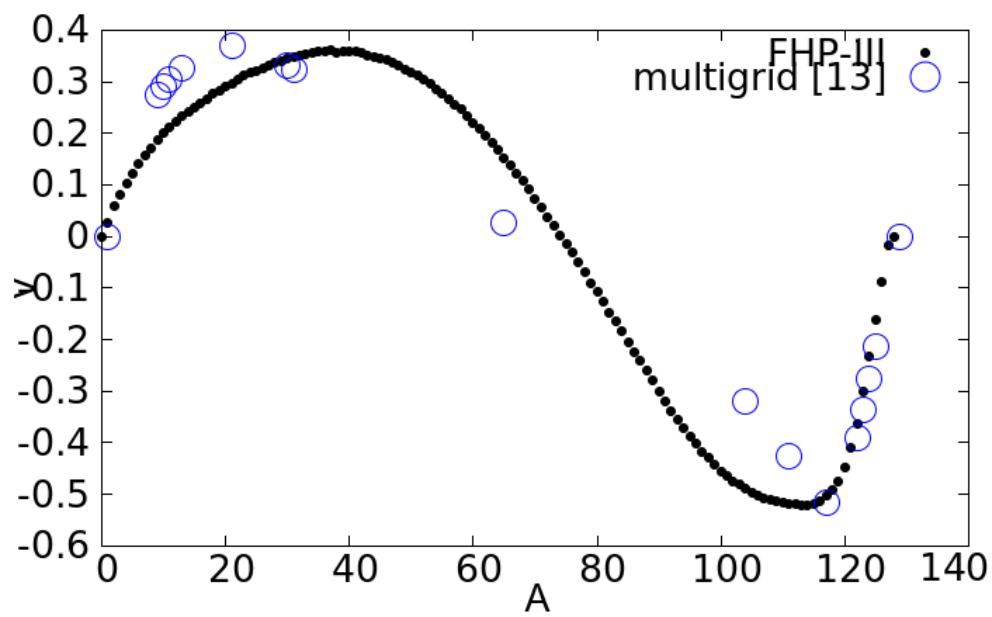
Rysunek 17: Porównanie wartości prędkości  $u$  dla  $Re = 120$  z [13] ( $Re = 100$ ).



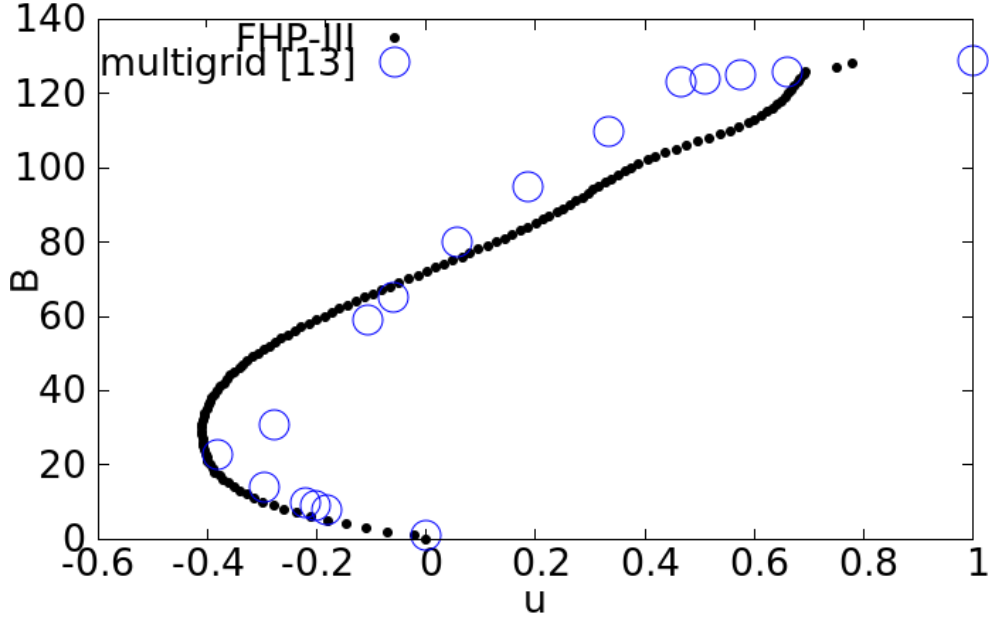
Rysunek 18: Porównanie wartości prędkości  $v$  dla  $Re = 400$  z [13] ( $Re = 400$ ).



Rysunek 19: Porównanie wartości prędkości  $u$  dla  $Re\ 400$  z [13] ( $Re\ 400$ ).



Rysunek 20: Porównanie wartości prędkości  $v$  dla  $Re\ 1050$  z [13] ( $Re\ 1000$ ).



Rysunek 21: Porównanie wartości prędkości  $u$  dla  $Re$  1050 z [13] ( $Re$  1000).

Nie trudno zauważyć różnicy między otrzymanymi wartościami a prędkościami pochodzących z [13]. Kształt wykresu zachowany jest szczególnie w przypadku wyników dla  $Re = 400$ , dla pozostałych widać pewną rozbieżność. Dla  $Re = 120$  i  $Re = 1050$  największe i najmniejsze wartości  $v$  są niemal identyczne, natomiast pozostałe wzdłuż środka układu są większe. Przede wszystkim może być to spowodowane różnicą wartości  $Re$ . Chociaż jest ona nie wielka, być może znacząco wpływa na wyniki. Ponadto, centrum geometrycznie nie koniecznie znajduje się dokładnie w środku układu. Minimalne przesunięcie osi, wzdłuż której zbierano wyniki lub uśrednienie ich w obszarze centrum mogłoby poprawić zaistniałe różnice.

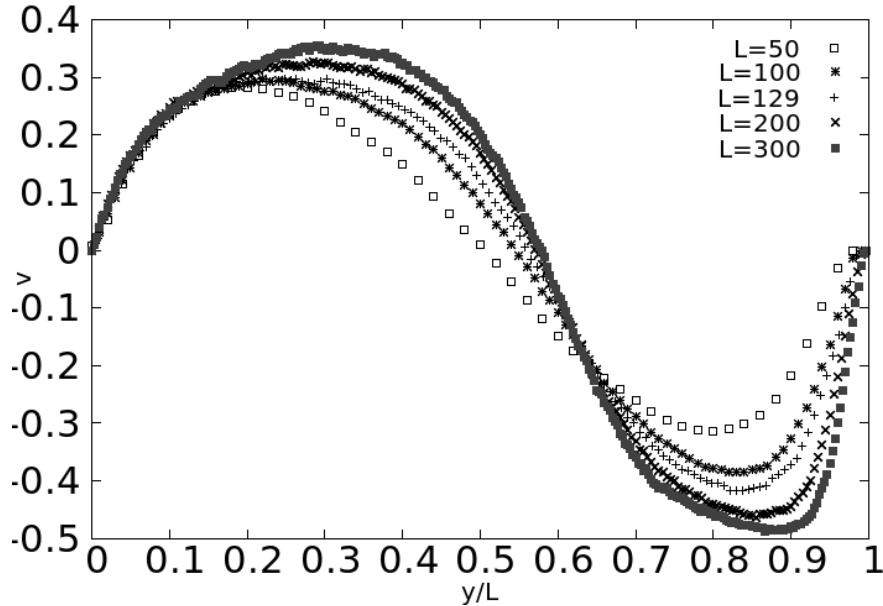
Warto również wziąć pod uwagę metody rozwiązania zagadnienia. Szum statystyczny obecny w gazach sieciowych ma duży wpływ na wyniki, dlatego dodatkowe uśrednienie w przestrzeni być może poprawiłoby kształt wykresów. Inną widoczną różnicą jest wielkość mniejszych wirów, za co prawdopodobnie odpowiada czas symulacji. Dodatkowo linie prądu, na przykład dla wyników o  $Re = 1050$ , w miejscu prawego rogu komory układają się w inny sposób niż w przypadku [13] dla  $Re = 1000$ , gdzie tworzący się tam wir jest wyraźnie większy. Stąd wartości prędkości mogą się różnić.

### 3.2 Dodatkowe uwagi

Głównym powodem różnic w prezentowanych wynikach jest niepewność co do wartości liczby Reynoldsa. Wszystkie układy miały ten sam wymiar i wypełniane były z taką samą gęstością. Jednak dodawanie pojedynczych cząstek w górnej pokrywie znacznie wpływa na lepkość kinematyczną w układzie, zgodnie ze wzorem 8. Chociaż masa stabilizowała się przy pewnej wartości, nie była dla wszystkich układów identyczna. Niewielka zmiana parametrów  $p_0 - p_6$ , czyli prawdopodobieństwa z jakim dodawano cząstki, znacznie wpływa zarówno na masę w układzie jak i na prędkość górnej pokrywy, a co za tym idzie na wartość liczby  $Re$ . Parametry doбираłam w ten sposób, by w każdym z układów dodawać te samą ilość cząstek. Jednak w celu uzyskania różnych

wartości  $Re$  takie rozwiązanie się nie sprawdza. Na przykład, dużą prędkość uzyskać najłatwiej, ponieważ wystarczy z dużym prawdopodobieństwem dodawać cząstki o dodatniej wartości prędkości. W celu uzyskania mniejszej prędkości (np. dla  $Re=100$ ) i jednocześnie zachowania masy, powinnam dodawać cząstki o ujemnej wartości prędkości z tą samą częstością. To z kolei powoduje, że uzyskana prędkość jest za mała a główny wir w komorze mało wyraźny. Być może dłuższy czas symulacji pomógłby w takiej sytuacji, jednak to również mogłoby wpływać na masę w układzie. Jak widać, prezentowany sposób przyspieszania górnej pokrywy niesie za sobą pewne problemy.

Rozmiar układu również może znacząco wpływać na jakość wyników. Nie trudno to zauważyć biorąc pod uwagę na przykład rozkład węzłów sieci w rogach dla mniejszego i większego układu. Wiry tworzące się w rogach większego układu będą zawsze mniej zniekształcone niż te w mniejszym. Najłatwiej zobrazować tę zależność na wykresie dla różnych rozmiarów sieci. Na rysunku 22 znajduje się wykres prędkości  $v$  w centrum geometrycznym (oś  $A$ ) dla układów o różnej wielkości.



Rysunek 22: Wartości prędkości  $v$  dla różnych rozmiarów układu, gdzie:  $Re \approx 650$ , czas symulacji  $t = 30000$  kroków.

Chociaż dostosowano rozmiar do [13], prawdopodobnie zwiększenie go byłoby lepszym rozwiązaniem, ponieważ wraz ze wzrostem rozmiaru  $L$  rośnie krzywizna wykresu prędkości. Wykorzystanie podobnego rozmiaru sieci miałoby więcej korzyści w przypadku porównywania wyników rozwiązania zagadnienia tą samą metodą. Z pewnością szersza analiza własności modelu FHP-III umożliwiłaby dokładne zrozumienie problemu i zmniejszenie zaistniałych niezgodności.

## A Testy poprawności modelu FHP-III

Niniejszy dodatek poświęcony został testom poprawności modelu FHP-III. Do najprostszych i najbardziej popularnych testów należy symulacja rozchodzącej się fali. Bardziej rozbudowanym zagadnieniem jest przepływ Poiseuille’a, który pozwala nie tylko na sprawdzenie poprawności implementacji modelu, ale również umożliwia zbadanie jego własności fizycznych.

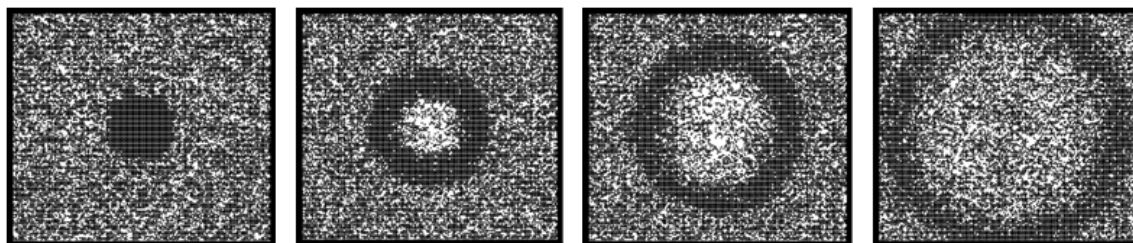
### A.1 Rozchodząca się fala

Aby osiągnąć efekt rozchodzącej się fali, wartość gęstości w układzie powinna być niewielka, natomiast w środku układu należy skoncentrować obszar o gęstości maksymalnej (okrąg). W tabeli 1 znajdują się parametry symulacji.

|                |                       |
|----------------|-----------------------|
| ROZMIAR SIECI  | $150 \times 150 j.s.$ |
| GĘSTOŚĆ UKŁADU | 0.2                   |
| GĘSTOŚĆ OKRĘGU | 1                     |
| PROMIEŃ OKRĘGU | $10 j.s.$             |
| IŁOŚĆ KROKÓW   | $t = 270$             |

Tabela 1: Parametry symulacji rozchodzącej się fali.

Układ będzie dążył do równowagi w wyniku czego zagęszczenie stopniowo będzie się rozchodzić wzdłuż wektorów sieci. W efekcie otrzymałam rozchodzącą się falę, co przedstawia rysunek 23.



Rysunek 23: Rozchodząca się fala.

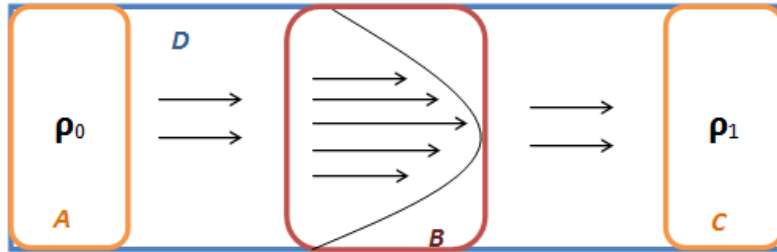
## A.2 Przepływ Poiseuille’a

Jednym z ważniejszych i prostszych rodzajów przepływu jest przepływ przez rurę lub szparę między równoległymi powierzchniami, kiedy na ściankach prędkości równe są zero (przy warunkach bez poślizgu), natomiast w środku prędkość osiąga maksimum. Taki przepływ nazywamy przepływem Poiseuille’a lub przepływem przez kanał płaski. Stanowi on jedno z nielicznych zagadnień hydrodynamicznych z pełnym rozwiązaniem analitycznym. Opisuje laminarny, jednokierunkowy przepływ cieczy przez wąskie naczynie, który nie zmienia się w czasie. Przy pominięciu sił masowych podtrzymywany jest przez stały gradient ciśnienia i dokonuje się w stałe określonym kierunku. W wyniku kontaktu płynu ze ściankami naczynia wytwarza się charakterystyczny profil prędkości. Prędkość płynu jest mniejsza przy ściankach kanału niż pomiędzy nimi, więc profil ma charakter paraboliczny zgodnie ze wzorem 10 [7]:

$$v(y) = -\frac{1}{2\mu} \frac{\Delta p}{\Delta x} y(h - y), \quad (10)$$

gdzie  $\mu$  to lepkość kinematyczna,  $\frac{\Delta p}{\Delta x}$  to gradient ciśnienia a  $h$  wysokość kanału.

Zgodnie z założeniami zagadnienia, utworzyłam prostokątny układ i wypełniłam cząstkami z prawdopodobieństwem równym  $p = 0.5$ . W celu wytworzenia stałego gradientu ciśnienia wybrałam dwa obszary w układzie: prawy i lewy, w których utrzymywałam gęstości przy wybranych wartościach. Niewielki obszar pomiędzy nimi przeznaczyłam do obliczeń. Uproszczony schemat symulacji przedstawia rysunek 24.



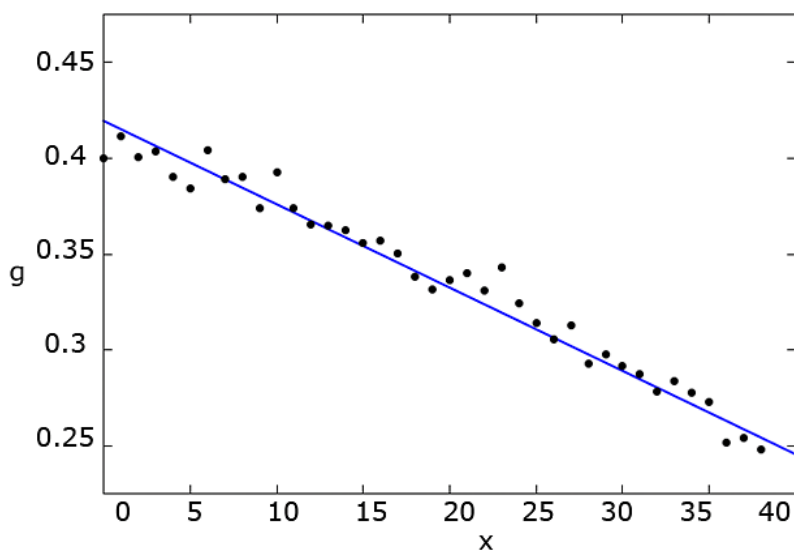
Rysunek 24: Schemat przepływu przez kanał płaski.

Obszary A i C stanowią rezerwuary ciśnień o wybranych gęstościach, w obszarze B obliczałam profil prędkości, natomiast obszar D obejmuje cały układ. Parametry schematu znajdują się w tabeli 2.

|                            |                     |
|----------------------------|---------------------|
| ROZMIAR SIECI (OBSZAR D)   | $200 \times 10j.s.$ |
| GĘSTOŚĆ $\rho_0$ OBSZARU A | 0.4                 |
| GĘSTOŚĆ $\rho_1$ OBSZARU C | 0.25                |
| ROZMIAR OBSZARÓW A I C     | $5 \times 10j.s.$   |
| ROZMIAR OBSZARU B          | $5 \times 10j.s.$   |
| CZAS TRWANIA               | 5000 kroków         |

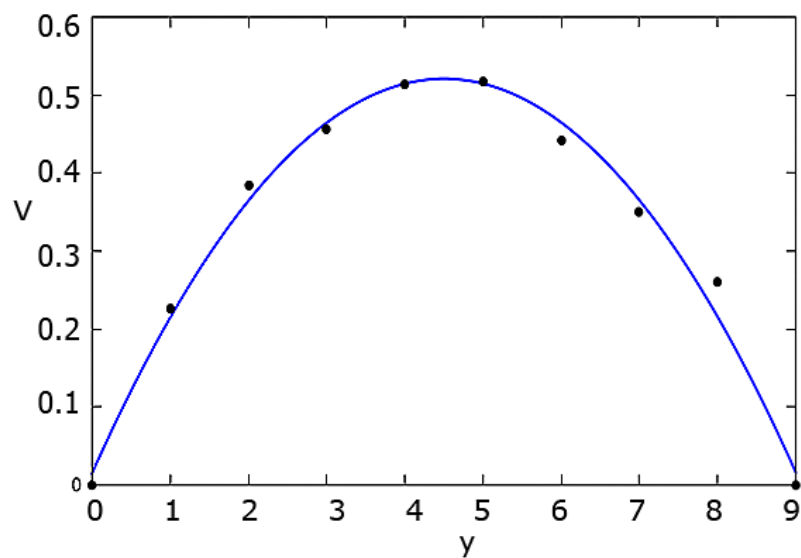
Tabela 2: Parametry symulacji przepływu Poiseuille’a.

Otrzymałam następujące rezultaty:



Rysunek 25: Gęstość w układzie.

Gęstość oscyluje na początku wokół wybranej wartości  $\rho_0$  i stopniowo spada do gęstości  $\rho_1$ . Wykres ma charakter liniowy, co oznacza, że gradient ciśnienia jest stały. Ostatnim etapem symulacji jest wyznaczenie profilu prędkości. W wybranym obszarze B, w każdym węźle sieci i wzdłuż osi OY liczyłam prędkość zgodnie z wcześniej zamieszczonym wzorem, a następnie uśredniłam wyniki. Rysunek 26 przedstawia otrzymane wartości oraz dopasowaną do nich parabolę dla lepszej czytelności.



Rysunek 26: Profil prędkości w przepływie Poiseuille.

Prędkości w wybranym obszarze układają się na wykresie zgodnie z oczekiwaniami, co pozwoliło otrzymać prawidłowy profil Poiseuille'a. Punkty niemal pokrywają się z pomocniczą parabolą, zatem test poprawności wykonałam z powodzeniem.

## B Fragmenty kodu symulacji

W niniejszym dodatku znajdują się wybrane fragmenty kodu napisane w języku C++ i nie stanowią one samodzielnej symulacji. Każdy z fragmentów został krótko objaśniony. Do implementacji modelu FHP-III w większości wykorzystałam następujące źródła: [7] , [12], [8].

Tablica wszystkich możliwych kolizji w modelu FHP-III z uwzględnieniem sytuacji kiedy cząstka może odbić się na dwa różne sposoby:

```
1 unsigned char zasadyKolizji[][] = {
2   {0, 1, 2, 3, 4, 66, 6, 7, 8, 36, 68, 38, 12, 74,
3    14, 15, 16, 96, 9, 98, 72, 42, 13, 102, 24, 52,
4    84, 45, 28, 90, 30, 110, 32, 33, 65, 35, 18, 19,
5    69, 39, 80, 81, 21, 83, 26, 54, 77, 87, 48, 49,
6    41, 51, 104, 105, 27, 107, 56, 57, 116, 117, 60,
7    122, 93, 63, 64, 34, 5, 67, 10, 11, 70, 71, 20,
8    100, 22, 23, 76, 86, 78, 79, 40, 50, 73, 101, 44,
9    106, 46, 47, 88, 58, 108, 109, 92, 62, 94, 95, 17,
10   97, 37, 99, 82, 43, 75, 103, 25, 114, 85, 55, 29,
11   118, 31, 111, 112, 113, 53, 115, 89, 59, 91, 119,
12   120, 121, 61, 123, 124, 125, 126, 127},
13   {0, 1, 2, 3, 4, 66, 6, 7, 8, 18, 68, 69, 12, 22,
14    14, 15, 16, 96, 36, 37, 72, 42, 74, 75, 24, 104,
15    44, 54, 28, 108, 30, 110, 32, 33, 65, 35, 9, 98,
16    11, 39, 80, 50, 21, 101, 84, 27, 86, 87, 48, 49,
17    81, 51, 25, 114, 45, 107, 56, 57, 89, 117, 60, 122,
18    93, 63, 64, 34, 5, 67, 10, 38, 70, 71, 20, 82, 13,
19    102, 76, 46, 78, 79, 40, 41, 100, 43, 26, 106, 77,
20    47, 88, 116, 29, 118, 92, 62, 94, 95, 17, 97, 19,
21    99, 73, 83, 23, 103, 52, 53, 85, 55, 90, 91, 31,
22    111, 112, 113, 105, 115, 58, 59, 109, 119, 120,
23    121, 61, 123, 124, 125, 126, 127}
24 };
```

Tablica przechowująca masę każdego z możliwych stanów węzła:

```
1 int masaWezla[ ] = {0, 1, 1, 2, 1, 2, 2,
2   3, 1, 2, 2, 3, 2, 3, 3, 4, 1, 2, 2, 3, 2, 3, 3,
3   4, 2, 3, 3, 4, 3, 4, 4, 5, 1, 2, 2, 3, 2, 3, 3,
4   4, 2, 3, 3, 4, 3, 4, 4, 5, 2, 3, 3, 4, 3, 4, 4,
5   5, 3, 4, 4, 5, 4, 5, 5, 6, 1, 2, 2, 3, 2, 3, 3,
6   4, 2, 3, 3, 4, 3, 4, 4, 5, 2, 3, 3, 4, 3, 4, 4,
7   5, 3, 4, 4, 5, 4, 5, 5, 6, 2, 3, 3, 4, 3, 4, 4,
8   5, 3, 4, 4, 5, 4, 5, 5, 6, 3, 4, 4, 5, 4, 5, 5,
9   6, 4, 5, 5, 6, 5, 6, 6, 7};
```

Proces kolizji z przeszkodą lub cząstkami:

```
1 for(int i=0; i<NX; i++){
2   for (int j=0; j<NY; j++){
3
4     if( przeszkody[i][j] == 0){
5       p = losujLiczbe(); //funkcja losujaca
6       wezel = (int)siec[i][j]; // wartosc wezla
7       stanPoKolizji[i][j] = zasadyKolizji[p][wezel]; //zapis po
      kolizji w tymczasowej tablicy
8     }
```

```

9     else if( przeszkody[i][j] == 1){
10         stanPoKolizji[i][j] |= ((1 & siec[i][j]) << 3);
11         stanPoKolizji[i][j] |= ((2 & siec[i][j]) << 3);
12         stanPoKolizji[i][j] |= ((4 & siec[i][j]) << 3);
13         stanPoKolizji[i][j] |= ((8 & siec[i][j]) >> 3);
14         stanPoKolizji[i][j] |= ((16 & siec[i][j]) >> 3);
15         stanPoKolizji[i][j] |= ((32 & siec[i][j]) >> 3);
16     }
17 }
18 }
19 }
20     odwroc(siec, stanPoKolizji);    //przepisanie wyników

```

Proces przesuwania cząstek po sieci zgodnie z ich wektorami prędkości, gdzie funkcja *warunekPrzejscia()* zawiera warunki brzegowe.

```

1  for(int i=0; i<NX; i++){
2      for (int j=0; j<NY; j++){
3
4          b = ( (j+2) - 1 ) % 2;    // parametr wskazujący sąsiadów
           poszczególnego węzła
5
6          wsx = (i+ b);
7          wsy = (j+1 ) ;
8          if(warunekPrzejscia(wsx,wsy))
9              stanPoPrzesunieciu[wsx][wsy] |= (1 & siec[i][j]);
10
11         wsx = (i+1);
12         wsy=j;
13         if(warunekPrzejscia(wsx,wsy))
14             stanPoPrzesunieciu[wsx][wsy] |= (2 & siec[i][j]);
15
16         wsx = (i+b);
17         wsy = (j-1);
18         if(warunekPrzejscia(wsx,wsy))
19             stanPoPrzesunieciu[wsx][wsy] |= (4 & siec[i][j]);
20
21         wsx = (i-1+b);
22         wsy = (j-1 ) ;
23         if(warunekPrzejscia(wsx,wsy))
24             stanPoPrzesunieciu[wsx][wsy] |= (8 & siec[i][j]);
25
26         wsx = (i-1);
27         wsy=j;
28         if(warunekPrzejscia(wsx,wsy))
29             stanPoPrzesunieciu[wsx][wsy] |= (16 & siec[i][j]);
30
31         wsx = (i-1+b);
32         wsy = (j+1 ) ;
33         if(warunekPrzejscia(wsx,wsy))
34             stanPoPrzesunieciu[wsx][wsy] |= (32 & siec[i][j]);
35
36         stanPoPrzesunieciu[i][j] |= ( 64 & siec[i][j] );
37     }
38 }
39 }
40     odwroc(siec, stanPoPrzesunieciu);

```

Funkcje zwracające prędkości  $v_x$  (u),  $v_y$  (v) dla danego węzła.

```

1 double predkoscU(unsigned char siec[ ][ ], int x, int y, int masa){
2
3 double sum = 0.0;
4 double u;
5 unsigned char wynik;
6 // wartosci predkosci u dla kazdego z wektorow sieci
7 double ex[z]={0.5, 1.0, 0.5, -0.5, -1.0, -0.5, 0.0};
8
9 if( masa == 0){
10     return sum;
11 }
12 else if( masa != 0){
13     for( int i=0; i<z; i++){
14         wynik = (unsigned char) pow(2, i) & siec[x][y];
15         if( (int)wynik != 0){
16             sum += ex[i];
17         }
18     }
19     u = (double)sum / (double)masa;
20     return u;
21 }
22
23 }

```

```

1 double predkoscV(unsigned char siec[ ][ ], int x, int y, int masa){
2
3 double sum=0.0;
4 double v;
5 unsigned char wynik;
6 //wartosci predkosci v dla kazdego z wektorow sieci
7 double ey[z] = {0.866, 0.0, -0.866, -0.866, 0.0, 0.866, 0.0};
8
9 if( masa == 0){
10     return sum;
11 }
12 else if(masa != 0 ){
13     for( int i=0; i < z; i++){
14         wynik = (unsigned char) pow(2,i) & siec[x][y];
15         if( (int)wynik != 0){
16             sum += ey[i];
17         }
18     }
19     v = (double)sum / (double)masa;
20     return v;
21 }
22
23 }

```

Funkcja dodająca cząstki do górnej pokrywy w zależności od ustalonych parametrów  $p0$ - $p7$ .

```

1 void gornaPokrywa( unsigned char siec[ ][ ]){
2
3 double p[z]={0.0,0.5,0.5,0.3,0.1,0.1,0.0}; //przykładowe wartosci
   parametrow p0-p6
4

```

```

5 for( int i=1; i<NX-1; i++){
6   for(int j=NY-B; j<NY; j++){ //zmienna B oznacza grubosc górnej
    pokrywy
7
8     for (int k=0; k < z; k++){
9       if( losowa() < p[k] ){ //funkcja losowa() zwraca liczbe z
        przedzialu <0,1)
10        siec[i][j] |= (unsigned char) pow(2,k);
11      }
12    }
13  }
14 }
15
16 }

```

Przebieg interpolacji wyników. W tablicach *sredniaU*, *sredniaV* znajdują się uśrednione w czasie wartości prędkości *u* i *v*.

```

1 for( int j=0; j<NY; j++){
2   for( int i=0; i<NX; i++){
3
4     if(j % 2 == 0 ){
5       wynikiU[i][j] = sredniaU[i][j];
6       wynikiV[i][j] = sredniaV[i][j];
7     }
8     else if ( j % 2 != 0){
9       wynikiU[i][j] = interpolacja(sredniaU, i, j);
10      wynikiV[i][j] = interpolacja(sredniaV, i, j);
11    }
12  }
13 }

```

gdzie funkcja *interpolacja()* wygląda następująco:

```

1 double interpolacja(double dane[ ][ ], int x, int y){
2   if( x == NX-1){
3     return dane[x][y];
4   }
5   else{
6     return ((dane[x][y] + dane[x+1][y])/2.0);
7   }
8 }

```

## Literatura

- [1] Clay Mathematics Institute Millennium Prize Problems, <http://www.claymath.org/millennium/NavierStokesEquations/navierstokes.pdf>.
- [2] Dieter A. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models*.
- [3] Gardner, Martin, *Mathematical Games – The fantastic combinations of John Conway's new solitaire game "life"*. Scientific American 223. pp. 120–123. **1970**.
- [4] Sauro Succi, *The Lattice Boltzmann Equation, for Fluid Dynamics and Beyond*, Oxford Science Publications, (Chapter 2).
- [5] U. Frish, B. Hasslaher, Y. Pomeau, *Lattice Gas Automata for the Navier-Stokes Equation*, Physical Review Letters vol. 56.
- [6] Bastien Chopard, Michel Droz, *Cellular Automata Modelling of Physical Systems*, University of Geneva.
- [7] S. Szczecina, *Własności Hydrodynamiczne Modelu Gazu Sieciowego FHP-III*, Uniwersytet Wrocławski, Wydział Fizyki i Astronomii, Instytut Fizyki Teoretycznej, **2008**.
- [8] S. Szkoda, *Implementacja Modelu FHP w Technologii NVIDIA CUDA*, Uniwersytet Wrocławski, Wydział Fizyki i Astronomii, Instytut Fizyki Teoretycznej.
- [9] Ercan Erturk and Orhan Gokcol, *Fine Grid Numerical Solutions of Triangular Cavity Flow*, The European Physical Journal - Applied Physics, textbf2007.
- [10] Michael C. Sukop, Daniel T. Thorne, Jr., *Lattice Boltzmann Modeling*, Springer, **(2006)**.
- [11] M. Krafczyk, E. Rank, *A Parallelized Lattice-Gas Solver for Transient Navier-Stokes-Flow: Implementation and Simulation Results*, FB Bauwesen, University of Dortmund, Germany. **1995**.
- [12] Kang Jin, *The Lattice Gas Model and Lattice Boltzmann Model On Hexagonal Grids*, Auburn University. **2005..**
- [13] U. GHIA, K. N. GHIA, AND C. T. SHIN, *High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method*, University of Cincinnati, Cincinnati, Ohio. **January 15, 1982**.