

УДК 519.63, 532.5

MSC 76M27

Viscous fluid flow modeling with the lattice Boltzmann method on graphics processors using WebGL API

S.O. Dovgiy¹, A.O. Ostapenko², G.G. Bulanchuk²¹Institute of Telecommunication and Global Information Space, Kyiv, Ukraine²Pryazovskyi State Technical University, Mariupol, Ukraine*E-mail: ostapt5@gmail.com*

This work is dedicated to the modeling methodology of a viscous fluid flows with the lattice Boltzmann method on graphic processors based on the technology of images rendering in web browsers WebGL. A two-dimensional nine-velocity LBM model (D2Q9) with a collision integral in a Bhatnagar-Gross-Kruk approximation form is shown. The possibilities of calculation acceleration using WebGL technology is described, namely features of using textures to contain values of some physical quantities in numerical algorithms and using framebuffer to storage the textures, influence of the texture parameters on the numerical algorithms, features of shaders programming. The questions of shader programs using for carrying out stages of physical modeling were considered. The proposed methodology was used to develop an original web program for modeling of classical test problems. Simulations of the Poiseuille flow in a plane channel and the flow around a circular cylinder in a plane channel were performed. The obtained results were compared with the results of calculations performed in the original verified modeling program based on the lattice Boltzmann method and in the Comsol Multiphysics package with the finite element method. Comparisons of the values of the velocity magnitude showed the consistency of the obtained results with the data of other numerical experiments. The analysis of computational speed in comparison with modeling using the optimized algorithm of a method with use of the technology of parallel calculations on CPU OpenMP in the original program is carried out. It is shown that the acceleration of calculations depends on the number of cells of the calculation grid. The results of the fluid flow modeling around a circular cylinder at $Re = 1000$ are demonstrated, which are obtained 30 times faster than with the calculations obtained with optimized lattice Boltzmann method and OpenMP technology.

Key words: viscous flows modeling, lattice Boltzmann method, technologies of parallel computations, graphic processors, textures.

Моделювання в'язких течій методом граткових рівнянь Больцмана на графічних процесорах з використанням WebGL API

С.О. Довгий¹, А.О. Остапенко², Г.Г. Буланчук²¹Інститут телекомунікацій і глобального інформаційного простору НАН України, Київ, Україна²ДВНЗ «Приазовський державний технічний університет», Маріуполь, Україна*E-mail: ostapt5@gmail.com*

Робота присвячена методології моделювання течій в'язкої рідини методом граткових рівнянь Больцмана на графічних процесорах на основі технології рендерингу зображень у веб-браузерах WebGL. Використовувалась двовимірна дев'ятишвидкісна модель методу LBM (D2Q9) із інтегралом зіткнення частинок у наближенні Бхатнагара-Гросса-Крука. Розглянуті можливості апаратного прискорення обчислень на основі технології WebGL: особливості реалізації обчислювальних алгоритмів із використанням текстур для зберігання даних фізичних величин, фреймбуферів для зберігання текстур, вплив текстурних параметрів на обчислення та особливості програмування шейдерів. Розглянуті також питання робо-

ти шейдерних програм для проведення етапів фізичного моделювання. Описана методологія була використана при розробці оригінальної веб-програми з моделювання класичних тестових задач. Проведено моделювання течії Пуазейля у плоскому каналі та обтікання кругового циліндра. Отримані результати порівнювалися із результатами обчислень, що проводилися за допомогою оригінальної верифікованої програми з моделювання методом ґраткових рівнянь Больцмана на CPU та в пакеті Comsol Multiphysics методом скінченних елементів. Порівняння значень модуля швидкості показали узгодженість отриманих результатів із даними інших чисельних експериментів. Був проведений аналіз швидкості обчислень у порівнянні з моделюванням за допомогою оптимізованого алгоритму методу LBM із використанням технології паралельних обчислень на CPU OpenMP. Показано, що прискорення обчислень залежить від кількості комірок розрахункової сітки. Швидкість розрахунків на GPU для кругового циліндра при $Re = 1000$ в 30 разів вища ніж на CPU.

Ключові слова: моделювання в'язких течій, метод ґраткових рівнянь Больцмана, технології паралельних обчислень, графічні процесори, текстури.

1. Вступ

Метод ґраткових рівнянь Больцмана (далі LBM, від англ. Lattice Boltzmann Method) є сучасним підходом до моделювання в'язких течій, що є альтернативою традиційним кінцево-різницевою методам [1-3]. Замість того, щоб розв'язувати рівняння Нав'є-Стокса, динаміка рідини моделюється як динаміка крупних частинок. Такий підхід був вперше розроблений Давидовим Ю. М. та Білоцерківським О. М. в 1965 році і відомий як метод крупних частинок [4]. Згодом, на основі клітинно-автоматної гідродинаміки (LGA, від англ. Lattice Gas Cellular Automata [1, 5]) він набув свого розвитку у методі LBM, в якому крупні частинки описуються статистично за допомогою функції розподілу частинок за координатами та швидкостями, що є розв'язком рівняння Больцмана.

На сьогодні метод ґраткових рівнянь Больцмана вже сміливо конкурує із іншими чисельними методами і дозволяє моделювати багатофазні та багатокомпонентні течії, течії з вільними границями та течії у пористих середовищах [6-8]. Найбільший доробок у його розвиток внесли такі світові вчені як S. Succi, F. Higuera, Y. L. He, Q. Li, Q. Chen, J. Latt, Q. Liu, K. H. Luo, A. Perumal, V. S. Kumar, A. K. Dass, D. Wolf – Gladrow, A. Л. Куперштох, Г. В. Кривовичев, Д. А. Бікулов, Д. С. Сенін. В Україні застосування методу до моделювання мікротечій викладено в роботах Тирінова И. І., Авраменка А. А., Басок Б. І. та Давиденка Б. В [9]. Така зростаюча популярність методу обумовлена простотою у його програмуванні, застосовністю до різних прикладних задач та використанню технологій паралельних обчислень.

Використання технологій паралельних обчислень є на сьогодні популярним напрямком у наукових дослідженнях. Однією з перших робіт із моделювання методом LBM на GPU була робота [10]. У цьому дослідженні та у його продовженні [11] автори описали моделювання задач газової динаміки з використанням текстури для зберігання даних та їх подальшої обробки. Проте переважна більшість робіт із цієї тематики присвячені використанню технології CUDA чи кластерів для апаратного прискорення обчислень [12-14].

Зважаючи на появу нових технологій, у даній роботі запропонований інших підхід – використання сучасного API WebGL для прискорення фізичного моделювання течій в'язкої рідини. Розглянута методологія такої реалізації, яка, окрім проведення обчислень паралельно на графічних процесорах, дозволяє створити про-

грамну реалізацію у браузері. Таким чином, такий застосунок є доступним будь-якому користувачу і не потребує жодних додаткових засобів, бібліотек тощо.

2. Метод ґраткових рівнянь Больцмана: D2Q9 модель та алгоритм

У роботі розглядаються плоскі ізотермічні течії в'язкої рідини. Обчислювальна область розбивається квадратними комітками зі стороною d . Вводиться ґраткова швидкість частинок c – фіксована величина, що визначає швидкість переміщення частинок у ґратковому просторі і обчислюється як $c = d / \Delta t$, де Δt – крок по часу. Для задання моделі решітки вводиться сукупність векторів можливих напрямків переміщення частинок $\vec{e}_k, k = \overline{0, n}$. Найбільш поширеною двовимірною моделлю є модель з дев'ятьма векторами можливих напрямків D2Q9. Кожному вектору \vec{e}_k ставиться у відповідність швидкість переміщення частинки у відповідному напрямку $\vec{V}_k = c \cdot \vec{e}_k$. Таким чином, у рамках D2Q9 моделі методу LBM у кожній комітці розрахункової сітки з координатами $\vec{r} = (x, y)$ та в момент часу t буде перебувати дев'ять значень функції розподілу частинок $f_k(\vec{r}, t), k = \overline{0, 8}$, які визначають ймовірність крупних частинок мати одну з дев'яти можливих швидкостей \vec{V}_k .

Система дискретних кінетичних рівнянь, що описує динаміку ансамблю крупних частинок із урахуванням відсутності зовнішніх сил та моделі зіткнень частинок у наближенні Бхатнагара-Гроса-Крука (BGK) має вид [1-3, 15]:

$$\underbrace{f_k(\vec{r} + \vec{V}_k \Delta t, t + \Delta t)}_{\text{перенесення}} = f_k(\vec{r}, t) - \underbrace{\frac{1}{\tau} [f_k(\vec{r}, t) - f_k^{eq}(\vec{r}, t)]}_{\text{релаксація}}, \quad (1)$$

де τ – безрозмірний параметр релаксації,

$f_k^{eq}(\vec{r}, t)$ – наближення локальної рівноважної функції розподілу Максвелла-Больцмана.

В'язкість рідини ν визначається як [16]:

$$\nu = \frac{1}{3} c^2 \Delta t \left(\tau - \frac{1}{2} \right) \quad (3)$$

Перехід від опису системи за допомогою функцій розподілу частинок $f_k(\vec{r}, t)$ до таких макроскопічних параметрів рідини як густина ρ , швидкість \vec{u} і тиск p здійснюється за формулами [1-3, 15]:

$$\rho(\vec{r}, t) = \sum_{k=0}^8 f_k(\vec{r}, t); \quad \vec{u}(\vec{r}, t) = \frac{1}{\rho(\vec{r}, t)} \sum_{k=0}^8 \vec{V}_k f_k(\vec{r}, t); \quad p(\vec{r}, t) = c_s^2 \rho(\vec{r}, t) \quad (4)$$

3. Методологія використання WebGL API до моделювання LBM

Класичне моделювання передбачає використання двовимірних масивів для зберігання даних поля швидкості чи густини та тривимірних масивів для зберігання дев'яти значень функцій розподілу у кожній комітці розрахункової сітки. Очевидно, таке представлення даних є дискретним, бо складається з комірок, в кожній з яких записано певне значення фізичної величини. Для моделювання на основі WebGL API для запису та зберігання даних були використані текстури. В загаль-

ному уявленні текстури представляють собою зображення, що формуються як сукупність пікселів, тобто існує відповідність текстурної координати певному кольору у цій точці.

Таким чином, текстуру можна використовувати замість масиву даних і в одній текстурі зберігати чотири значення певної фізичної величини, які у внутрішньому представленні маскуються під колір пікселя у форматі RGBA. При цьому рендерінг зображення, тобто процес формування кольорів та отримання нових значень кольору зображення, при такому підході є етапом обчислення при фізичному моделюванні.

WebGL, як інструмент рендерингу зображень, передбачає роботу шейдерної програми для створення зображень. Шейдерна програма включає в себе два шейдери, що написані на мові GLSL - вершинний та фрагментний. Вершинний шейдер оперує координатами простору відображення, а фрагментний шейдер задає колір у кожній точці простору відображення у форматі RGBA.

Послідовність фізичного моделювання для такого підходу наступна:

1. Створення об'єктів FBO (framebuffer object), що зберігають інформацію про текстуру цього об'єкта. Для моделювання методом LBM необхідно створити одну текстуру для зберігання значень густини та двох значень швидкості в точці та три текстури для зберігання дев'яти значень функції розподілу. Для зберігання координат тіл обтікання теж була використана текстура.

2. Написання шейдерів. Шейдерна програма складається зі стандартного вершинного шейдера, що отримує вершинні координати площини відображення $\vec{x} = (x, y); x, y \in [-1; 1]$ та перетворює їх у текстурні координати $\vec{t} = (t_x, t_y); t_x, t_y \in [0; 1]$ відповідно до формули $\vec{t} = 0,5\vec{x} + 0,5$.

Фрагментний шейдер отримує інформацію із інших текстур та змінних та формує вихідне значення кольору пікселя текстурної координати за певною чисельною схемою. Таким чином, робота фрагментного шейдеру дорівнює етапу обчислення, що програмно записується для однієї комірки точки простору.

3. Створення шейдерної програми: додавання до програми вершинного та фрагментного шейдерів та отримання змінних із шейдера. Робота шейдерної програми з рендерингу зображення є етап запису нових даних у текстуру.

4. Написання алгоритму фізичного моделювання з використанням шейдерних програм.

4. Результати

Незважаючи на те, що описаний у п.2. алгоритм D2Q9-BGK моделі методу LBM неодноразово проходив верифікацію авторами на низці прикладних задач, актуальними є питання перевірки запропонованої методології моделювання. Цікавим є питання: чи зберігається точність моделювання при використанні текстур для зберігання та обробки даних і наскільки збільшується швидкість розрахунків у порівнянні з раніше описаним оптимізованим алгоритмом, що реалізований з використанням технології паралельних обчислень на центральному процесорі OpenMP.

Були розглянуті класичні тестові задачі з моделювання течії Пуазейля та обтікання кругового циліндра у плоскому каналі.

Отримані результати порівнювались як з результатами класичного моделювання методом LBM, так і з результатами, отриманими методом скінчених елементів (FEM) у пакеті Comsol Multiphysics та показали точність моделювання з використанням WebGL та фізичну відповідність результатів. На рис. 1 зліва представлені графіки розподілу модуля швидкості течії Пуазейля, отримані методами LBM та FEM при $Re = 200$. Причому результати методом LBM, що отримані на CPU та GPU співпадають. На рис. 1 справа зображена діаграма модуля швидкості при обтіканні кругового циліндра при $Re = 200$ ($t = 100$), що отримана методом LBM на GPU.

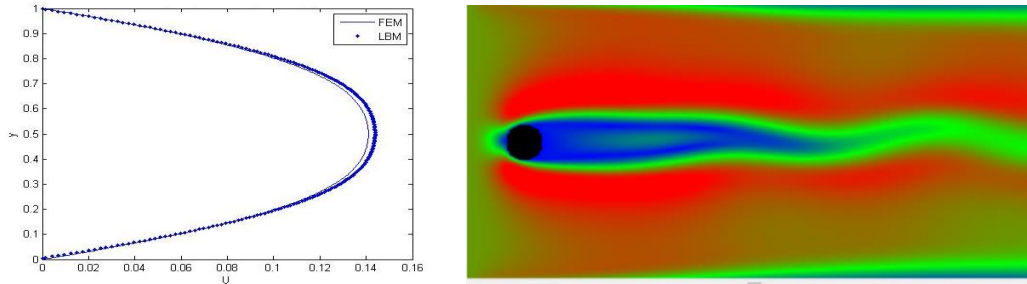


Рис. 1. Порівняння графіків розподілу модуля швидкості течії Пуазейля (зліва) та діаграма модуля швидкості при обтіканні кругового циліндра (справа)

Була проведена оцінка прискорення обчислень на графічних процесорах на прикладі моделювання обтікання кругового циліндра. Проведено моделювання в області, що має розмір 3×1 . Моделювання проводилось до моменту часу $t = 100$. Чисельний метод в усіх випадках мав фіксовану ґраткову швидкість $c = 1$. Таблиця 1 містить порівняння швидкості обчислень при $Re = 100$ для різної кількості комірок на одиницю довжини N .

Таблиця 1. Порівняння часу моделювання методом LBM із використанням різних технологій при $Re = 100$.

$N =$	200	300	400	500
OpenMP	2200с (~37мин)	7100с (~2 год)	18100с (~2 год)	36000с (~10 год)
WebGL	340с (~6 хв)	505с (~9 хв)	690с (~12 хв)	1230с (~21 хв)
Прискорення	6,5 р	14 р	26,2 р	29,3 р

При моделюванні методом LBM зі сталою ґратковою швидкістю частинок $c = 1$, крок по простору (розмір комірки) дорівнює кроку по часу $d = \Delta t$. Таким чином, незалежно від інших чисельних параметрів, обчислення на однакових за розміром розрахункових сітках займає приблизно однаковий час. Наприклад, при $Re = 200$ та $N = 200$ час моделювання з використанням OpenMP становить 40 хв., а WebGL – 5,7 хв., як і для $Re = 100$.

Однак моделювання із більшими числами Рейнольдса вимагає істотного подібнення розрахункової сітки. Так, при $Re = 1000$ кількість комірок повинно бути

щонайменше $N \geq 500$. Таке моделювання проводити без WebGL не доцільно, оскільки займає від 10 годин для $N = 500$. Аналогічне обчислення з використанням описаної методики з використанням технології WebGL займає 22 хвилини. Рисунок 2 ілюструє порівняння діаграм модуля швидкості, що були отримані методом LBM та FEM.

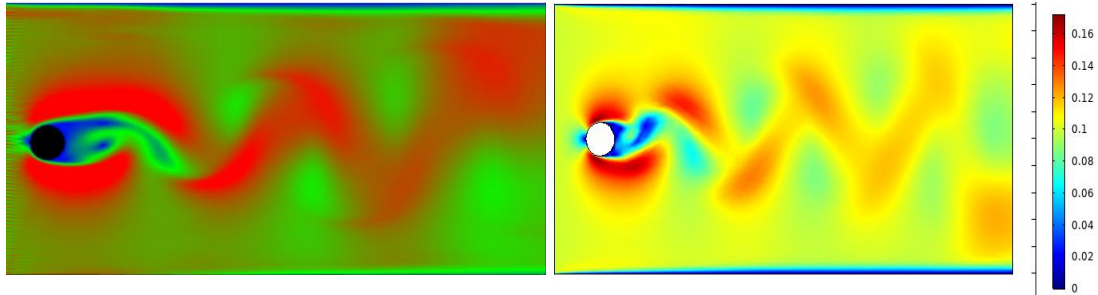


Рис. 2. Кольорова діаграма модуля швидкості при обтіканні кругового циліндра при $Re = 1000$ методом LBM з WebGL (зліва) та FEM (справа)

Варто зазначити, що моделювання було проведено на комп'ютері з двоядерним процесором Intel Core i3-5005U та дискретною відеокартою GeForce 920M. Використання більш потужної техніки потенційно пришвидшить розрахунки на основі обох технологій паралельних обчислень: OpenMP та WebGL.

5. Висновки

У роботі викладено методологію моделювання в'язких течій методом ґраткових рівнянь Больцмана на графічних процесорах з використанням технології WebGL. Можна зробити такі висновки:

1. Текстуру можна використовувати замість масиву даних і в одній текстурі зберігати чотири значення певної фізичної величини, які у внутрішньому представленні маскуються під колір пікселя у форматі RGBA. При цьому рендеринг зображення, тобто процес формування кольорів та отримання нових значень кольору зображення, при такому підході є етапом обчислення при фізичному моделюванні.

2. Дані, що зберігаються у текстурі повинні мати тип float. Використання half_float призводить до нестійкості розв'язків.

3. Швидкість обчислень зростає до 30 разів зі збільшенням кількості комірок розрахункової області, що дозволяє істотно пришвидшити фізичне моделювання.

4. Запропонована реалізація методу, окрім проведення обчислень паралельно на графічних процесорах, дозволяє створити програмну реалізацію у браузері. Отриманий застосунок є доступним будь-якому користувачу і не потребує жодних додаткових засобів, бібліотек тощо.

У подальшому планується більш детально розглянути текстурні параметри, такі як `gl.NEAREST` та `gl.LINEAR` і дослідити їх вплив на чисельний розв'язок. Припускається, що використання `gl.LINEAR` для отримання значення фізичної величини з текстури, як лінійну апроксимацію сусідніх значень, можна використати замість процедури регуляризації для згладження дрібних флуктуації у розв'язках при числах Рейнольдса $10^2 \leq Re \leq 10^5$.

ЛІТЕРАТУРА

1. Wolf-Gladrow D. Lattice-Gas Cellular Automata and Lattice Boltzmann Models – An Introduction. Bremerhaven: Alfred Wegener Institute for Polar and Marine, 2005. 311 p.
2. Succi S., Benzi R., Higuera F. The lattice Boltzmann equation: a new tool for computational fluid-dynamics. *Physica D*. 1991. 47. P. 219–230.
3. Succi S. The lattice Boltzmann equation for fluids and beyond. Oxford: Oxford University Press, 2001. 290 p.
4. Белоцерковский О.М., Давыдов Ю.М. Метод крупных частиц в газовой динамике. М.: Наука, 1982. 392 с.
5. Бандман О.Л. Клеточно-автоматные модели естественных процессов и их реализация на современных компьютерах. *Прикладная дискретная математика*. 2017. № 35. С. 102–121.
6. Guo Z., Zhao T.S. Lattice Boltzmann model for incompressible flows through porous media. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* 2002. Vol. 66. P. 036304-1–036304-9.
7. Derksen J.J. The lattice-Boltzmann method for multiphase fluid flow simulations and Euler-Lagrange large-eddy simulations. *Multiphase Reacting Flows: Modelling and Simulation*. 2006. Vol. 492. P. 181–228.
8. Куперштох А.Л. Метод решеточных уравнений Больцмана для моделирования двухфазных систем типа жидкость-пар. *Современная наука*. 2010. Т. 4, № 2. С. 56–63.
9. Тыринов А.И., Авраменко А.А., Басок Б.И., Давиденко Б.В. Моделирование микротечений методом решеток Больцмана. *Промышленная теплотехника*. 2011. Т. 33, № 2. С. 11–18.
10. Li W., Wei X., Kaufman A. Implementation lattice Boltzmann computation on graphic hardware. *Visual Computer*. 2003. 19. P. 444–456.
11. Wei X., Muller K., Li W., Kaufman A. The Lattice-Boltzmann Method for Simulation Gaseous Phenomena. *IEEE transactions on visualization and computer graphics*. 2004. V. 10, no. 2. P. 164–176.
12. Бикулов Д.А., Сенин Д.С., Демин Д.С., Дмитриев А.В., Грачев Н.Е. Реализация метода решеточных уравнений Больцмана для расчетов на GPU-кластере. *Вычислительные методы и программирование*. 2012. Т. 13. С. 13–19.
13. Куперштох А.Л. Реализация метода решеточных уравнений Больцмана на многопроцессорных графических ускорителях для 3d моделирования двухфазных систем типа жидкость-пар. *Современная наука*. 2011. Т. 7, № 2. С. 112–118.
14. Obrecht C., Kuznik F., Tourancheau B., Roux J. Multi-GPU implementation of the lattice Boltzmann method. *Computers and Mathematics with Applications*. 2013. 65. P. 252–261.
15. Succi M.C., Thorne D.T. Lattice Boltzmann modeling. An introduction for geophysics and engineeres. Miama: Springer, 2006. 173 p.

Надійшла 20.05.2021.

Моделирование вязких течений методом решеточных уравнений Больцмана на графических процессорах с использованием WebGL API

С.А. Долгий¹, А.А. Остапенко², Г.Г. Буланчук²

¹ Институт телекоммуникаций и глобального информационного пространства НАН Украины, Киев, Украина

² ГВУЗ «Приазовский государственный технический университет», Мариуполь, Украина
E-mail: ostapt5@gmail.com

Работа посвящена методологии моделирования течений вязкой жидкости методом решеточных уравнений Больцмана на графических процессорах на основе технологии рендеринга изображений в веб-браузерах WebGL. Описана двумерная девятискоростная модель метода LBM (D2Q9) с интегралом столкновения частиц в приближении Бхатнагара-Гросса-Крука. Рассмотрены возможности аппаратного ускорения вычислений на основе технологии WebGL, а именно особенности реализации вычислительных алгоритмов с использованием текстур для хранения данных физических величин, фреймбуфер для хранения текстур, влияния текстурных параметров на вычисления и особенности программирования шейдеров. Рассмотрены вопросы работы шейдерных программ для проведения этапов физического моделирования. Описанная методология была использована при разработке оригинальной веб-приложения по моделированию классических тестовых задач. Проведено моделирование течения Пуазейля и обтекание кругового цилиндра в плоском канале. Полученные результаты сравнивались с результатами вычислений, полученные с помощью оригинальной верифицированной программы по моделированию методом решеточных уравнений Больцмана на CPU и в пакете Comsol Multiphysics методом конечных элементов. Сравнение значений модуля скорости показали согласованность полученных результатов с данными других численных экспериментов. Был проведен анализ скорости вычислений по сравнению с моделированием с помощью оптимизированного алгоритма метода LBM с использованием технологии параллельных вычислений на CPU OpenMP. Показано, что ускорение вычислений зависит от количества ячеек расчетной сетки. Скорость расчетов для кругового цилиндра при $Re = 1000$ в 30 раз выше, чем вычисления на CPU.

Ключевые слова: моделирование вязких течений, метод решеточных уравнений Больцмана, технологии параллельных вычислений, графические процессоры, текстуры.