

С. О. ДОВГИЙ, Г. Г. БУЛАНЧУК, О. М. БУЛАНЧУК, В. В. ЛИСТОПАДОВА

ТОЧНІСТЬ РОЗРАХУНКІВ ПОЛЯ ШВИДКОСТЕЙ ВІД СИСТЕМИ ВИХРОВИХ ВІДРІЗКІВ ПРИ ВИКОРИСТАННІ ГРАФІЧНИХ ПРОЦЕСОРІВ

У роботі досліджувалась можливість використання графічних процесорів для розрахунку поля швидкостей від системи вихрових відрізків в ідеальній рідині при тривимірному моделюванні. Дане дослідження є важливим з огляду на велику кількість методів, що використовують дискретні вихрові структури при моделюванні тривимірних течій. При моделюванні міських забудов вихровими методами генерується велика кількість вихрових відрізків. При цьому складність алгоритму для розрахунку таких систем пропорційна квадрату кількості елементів. Це вимагає значних обчислювальних ресурсів, особливо на тривалому проміжку часу або при великій кількості об'єктів обтікання. Графічні процесори вже давно активно використовуються для пришвидшення такого роду розрахунків, зокрема і для моделювання вихрових структур. Проте, слід зауважити, що при цьому максимальна швидкість моделювання досягається при використанні змінних типу float і це призводить до збільшення похибок обчислень. Тому при використанні GPU у методах, де треба обчислювати поле швидкостей від сукупності вихрових елементів, наприклад, точкових вихорів у методі дискретних вихорів чи вихрових відрізків у методі дискретних вихрових рамок, ми повинні бути впевнені, що не втрачаємо точність обчислень.

Розглядалися дві системи вихрових відрізків: відрізки, що розміщені вздовж прямої, та відрізки, що сходять із пластинки при відриві потоку. Основна увага приділяється дослідженню точності розрахунків, формуванню структур даних та їх передачі на графічний процесор. Результати вказують на те, що правильна організація передачі даних із центрального процесора на графічний і навпаки робить можливим збільшити швидкість обчислень без суттєвої втрати точності розрахунків.

Ключові слова: OpenGL, вихрові відрізки, GPU, обчислювальні шейдери, ідеальна рідина, вихрова пелена.

С. А. ДОВГИЙ, Г. Г. БУЛАНЧУК, О. Н. БУЛАНЧУК, В. В. ЛИСТОПАДОВА

ТОЧНОСТЬ РАСЧЕТОВ ПОЛЯ СКОРОСТЕЙ ОТ СИСТЕМЫ ВИХРЕВЫХ ОТРЕЗКОВ ПРИ ИСПОЛЬЗОВАНИИ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

В работе исследовалась возможность использования графических процессоров для расчета поля скоростей от системы вихревых отрезков в идеальной жидкости при трехмерном моделировании. Данное исследование важно, учитывая большое количество методов, использующих дискретные вихревые структуры при моделировании трехмерных течений. При моделировании городских застроек вихревыми методами генерируется огромное количество вихревых отрезков. При этом сложность алгоритма для расчета таких систем пропорциональна квадрату количества элементов. Это требует значительных вычислительных ресурсов, особенно на длительном промежутке времени или при большом количестве объектов обтекания. Графические процессоры уже давно активно используются для ускорения такого рода расчетов, в том числе и для моделирования вихревых структур. Однако следует отметить, что при этом максимальная скорость моделирования достигается при использовании переменных типа float и это приводит к увеличению погрешностей вычислений. Поэтому при использовании GPU в методах, где нужно вычислять поле скоростей от совокупности вихревых элементов, например, точечных вихрей в методе дискретных вихрей или вихревых отрезков в методе дискретных вихревых рамок, мы должны быть уверены, что не теряем точность вычислений.

Рассматривались две системы вихревых отрезков: отрезки, расположенные по прямой, и отрезки, сходящие с пластинки при отрыве потока. Основное внимание уделяется исследованию точности расчетов, формированию структур данных и их передаче на графический процессор. Результаты указывают на то, что правильная организация передачи данных с центрального процессора на графический и наоборот позволяет увеличить скорость вычислений без существенной потери точности расчетов.

Ключевые слова: OpenGL, вихревые отрезки, GPU, вычислительные шейдеры, идеальная жидкость, вихревая пелена

S. O. DOVGIY, G. G. BULANCHUK, O. M. BULANCHUK, V. V. LYSTOPADOVA

ACCURACY OF CALCULATIONS OF THE VELOCITY FIELD FROM THE SYSTEM OF VORTEX SEGMENTS USING GRAPHIC PROCESSORS

The paper explored the possibility of using graphics processors to calculate the velocity field from a system of vortex segments in the ideal fluid three-dimensional simulation. This study is important in view of the large number of methods that use discrete vortex structures in modeling of three-dimensional flows. When simulating numerically urban buildings by vortex methods, a huge number of vortex segments are generated. At the same time, the algorithm complexity for calculating such systems is proportional to the square of the number of elements. This requires significant computing resources, especially over a long time or with a large number of fluid-affected objects. Graphics processors have long been actively used to speed up this kind of calculations, in particular, for vortex structures modeling. However, it should be noted that the maximum simulation speed is achieved when using float type variables and this leads to an increase in calculation errors. Therefore, when using the GPU in methods where it is necessary to calculate the velocity field from a set of vortex elements, for example, point vortices in the method of discrete vortices or vortex segments in the method of discrete vortex frames, we must be sure that we do not lose the accuracy of calculations.

Two systems of vortex segments were considered: segments located along a straight line and segments leaving the plate during flow separation. The main attention is paid to the study of the accuracy of calculations, the formation of data structures and their transfer to the graphics processor. The results indicate that the correct organization of data transfer from the central processor to the graphic processor and vice versa allows to significantly improve the speed of calculations without substantial loss of calculations accuracy.

Key words: OpenGL, vortex segments, GPU, computer shaders, ideal fluid, vortex sheet.

Вступ. Останнім часом з'являється багато публікацій по моделюванню на GPU різних фізичних процесів, зокрема течій рідини. Як свідчать останні дослідження [1] – [6], графічні процесори є ефективним інструментом для моделювання руху вихрових структур. Вони забезпечують швидке виконання великих обчислювальних задач навіть для тривимірного випадку, що дозволяє досліджувати складні явища в реальному часі.

Хоча використання графічних процесорів для обчислень може бути дуже ефективним, однак це пов'язано з деякими проблемами і труднощами. Графічні процесори мають обмежену кількість пам'яті, доступну для обчислень. Це може бути проблемою при розрахунках з великою кількістю даних або при складних моделях. Графічні

процесори мають свою власну архітектуру та набір інструкцій, які відрізняються від стандартних процесорів і це може створювати проблеми при розробці та оптимізації алгоритмів. Розробка *ефективних алгоритмів* для розрахунку на графічних процесорах може бути важкою через проблеми з синхронізацією та координацією даних між центральним процесором та графічним. Для ефективної роботи з графічними процесорами потрібно розділяти обчислення на блоки. Однак розмір блоку може мати вплив на продуктивність та ефективність розрахунків. Нерідко програмне забезпечення не відповідає можливостям апаратного забезпечення, тобто не використовує його повністю, що призводить до зниження продуктивності. Отже, використання графічних процесорів для обчислень має свою специфіку і вимагає ретельного підходу в програмуванні та оптимізації алгоритмів.

Для розрахунків з використання графічних процесорів використовуються різні програмні засоби: *Cuda*, *OpenCL*, *OpenGL*. У даній роботі розглядається використання *відкритої бібліотеки OpenTK*, яка була перероблена на мову C#. Ця бібліотека дає можливість використовувати *комп'ютерні шейдери* [7], на основі яких і були проведені дані розрахунки. Для програмування комп'ютерних шейдерів необхідна відеокарта з підтримкою OpenGL 4.3 та вище. Шейдери були створені на *Ci-подібній мові GLSL*. Однією із переваг такого підходу є те, що дані для вихрових відрізків, які використовуються для розрахунку швидкості, використовуються також і для візуалізації вихрової пелени.

Графічні процесори забезпечують збільшення швидкості обчислень та прискорення візуалізації, оскільки в силу своєї архітектури містять тисячі скалярних процесорів і дані, які потрібні для візуалізації, вже знаходяться у пам'яті GPU. Однак при цьому є необхідність використовувати додатковий час на копіювання даних із основної пам'яті в пам'ять GPU. Для того, щоб цей розрахунок був достатньо швидким, при копіюванні даних із основної пам'яті на GPU треба дані із *tiny double* перетворювати у *tin float*. Крім того, при передачі даних на графічний процесор треба поділяти їх на блоки. Ці особливості використання GPU і досліджуються в даній роботі.

Постановка задачі. У тривимірному випадку при моделюванні обтікання одного будинку одиничного нормованого розміру *методом дискретних вихрових рамок* [8] з однієї будівлі за одиницю часу сходять пелена, що містить близько 1500 вихрових відрізків, які її моделюють. У випадку житлових масивів, що містять десятки чи сотні будинків, кількість вихрових відрізків стає дуже великою і при розрахунку швидкості від них складність алгоритму має порядок квадрату кількості вихрових відрізків, тобто становить $O(n^2)$. Тому швидкість моделювання значно сповільнюється і *актуальною задачею є прискорення розрахунків*, яке може бути здійснено з використанням *багатоядерних процесорів* або *графічних процесорів на відеокартах*. Але при використанні графічних процесорів, доступних по ціні, найбільша швидкість досягається при використанні одинарного типу даних із плаваючою точкою (4 байти), що веде до втрати точності при розрахунках. При цьому виникає питання, чи не буде ця втрата досить значною, щоб істотно спотворити результати розрахунків.

Розглядається задача *тривимірного моделювання вихрового руху*, що генерується системою вихрових відрізків, які знаходяться в ідеальній рідині. Швидкість від кожного відрізка може бути обчислена за *формулою Біо – Савара*:

$$\vec{v} = \frac{\Gamma}{4\pi} \int_L \frac{d\vec{r} \times \vec{r}}{r^3}.$$

При чисельних розрахунках, вихрові лінії розбиваються на вихрові відрізки. Формула для обчислення швидкості в точці з радіус-вектором \vec{r} від вихрового відрізка має вигляд:

$$\vec{v} = \frac{\Gamma}{4\pi} \frac{[d\vec{l} \times \vec{R}]}{|d\vec{l} \times \vec{R}|^2} \left(\frac{d\vec{l} \cdot \vec{R}}{R} - \frac{d\vec{l} \cdot (\vec{R} - d\vec{l})}{|\vec{R} - d\vec{l}|} \right);$$

$$\vec{R} = \vec{r} - \vec{r}_0;$$

$$R = |\vec{R}|,$$

де \vec{r}_0 – радіус-вектор початку відрізка; $d\vec{l}$ – вектор, напрямлений з початку відрізка до його кінця. При цьому для уникнення ділення на нуль робилась перевірка, щоб виконувалась умова $[d\vec{l} \times \vec{R}] \neq 0$. Якщо це було не так, швидкість покладалася рівною нулю.

Результати досліджень. Розглядалися дві системи вихрових відрізків у тривимірному просторі: відрізки, що розміщені вздовж прямої та відрізки, що сходять із пластинки при відриві потоку.

Був розроблений алгоритм програмування, який виглядає наступним чином:

1. При розрахунках на центральному процесорі створюються *структури даних* для вихрових відрізків. У нашому випадку такою структурою була структура, що містила: початок відрізка, вектор $d\vec{l}$ і циркуляцію відрізка (усі дані мали подвійну точність).

2. Для розрахунків на GPU необхідно було створити дві структури даних (усі компоненти яких мають тип

float): одна в оперативній пам'яті комп'ютера, а інша – у пам'яті графічного процесора. Оскільки дані копіювались із оперативної пам'яті у пам'ять GPU, то мала бути встановлена відповідність між цими структурами даних. Структура в оперативній пам'яті обов'язково повинна мати *mun struct*. Окрім того, з метою подальшого використання цієї структури для відображення пелени, замість векторного елемента \vec{d} задавався кінець вихрового відрізка. Оскільки розмір структур має бути кратним 16 байтам (а отримали 28 байт), то треба було задати ще одну надлишкову змінну типу float. У цю змінну було поміщено *радіус Ренкіна*, який використовується для обмеження швидкості при наближенні до вихрового відрізка [8]. Слід зазначити, що в даному дослідженні він не використовувався, оскільки при даних обчисленнях немає перетину відрізків.

При копіюванні на графічний процесор було встановлено, що змінні в структурі, що знаходиться в оперативній пам'яті, не можуть бути задані у будь-якому порядку. Компоновка даних всередині структури має бути такою, щоб розмір блока змінних дорівнював 16 байт. Наприклад, якщо в оперативній пам'яті структура задана у вигляді:

```
public struct VortexSegmentStruct
{
    public Vector3f r0; // вектор початку вихрового відрізка
    public float gamma; // циркуляція вихрового відрізка
    public Vector3f r1; // вектор кінця вихрового відрізка
    public float RankineRadius;
}
то на графічному процесорі відповідна структура повинна мати вигляд:
struct VortexSegment
{
    vec3 r0; // вектор початку вихрового відрізка
    float gamma; // циркуляція вихрового відрізка
    vec3 r1; // вектор кінця вихрового відрізка
    float rankineRadius;
};
```

де Vector3f, vec3 – структури даних, що містять три числа типу float (разом 12 байт). Якщо спочатку вектори r0 та r1 слідуватимуть один за одним, а потім – циркуляція і радіус Ренкіна, то дані не будуть коректно скопійовані у пам'ять відеокарти.

Розрахунки проводились наступним чином:

1. На CPU створювались структури даних для системи вихрових відрізків із подвійною точністю.
2. Задавалась система тестових точок для розрахунку в них швидкостей від системи відрізків.
3. У тестових точках проводився розрахунок швидкості на CPU від системи вихрових відрізків.
4. Із структур даних на *кроці 1* створювались структури вихрових відрізків звичайної точності.
5. Із координат тестових точок на *кроці 2* створювались координати тестових точок звичайної точності.
6. Структури даних із *кроків 4* та *5* копіювались у пам'ять графічного процесора.
7. У пам'яті графічного процесора створювався буфер для швидкості, що розраховувалась.
8. Проводився розрахунок швидкості на GPU і дані записувались у буфер швидкості.
9. Дані із буфера швидкості на GPU копіювались в оперативну пам'ять.
10. Проводилось порівняння отриманих результатів.

При розрахунках змінювалась кількість вихрових відрізків та відстань, на якій знаходились тестові точки від вихрових відрізків. При цьому обчислювалась відносна похибка швидкості в кожній тестовій точці порівняно з розрахунками на центральному процесорі і знаходилося її максимальне значення.

На рис. 1 зображена залежність максимальної відносної похибки обчислення швидкості на GPU в залежності від відстані d до системи вихрових відрізків. Тестові точки розміщувались на відрізку та чверті кола симетрично відносно осі y (зображена половина точок) на відстані d від системи вихрових відрізків. Система вихрових відрізків знаходилась на осі Ox на інтервалі $[-0.5, 0.5]$. Розглядалася різна кількість однакових вихрових відрізків: 4, 8, 16. Циркуляція кожного вихрового відрізка дорівнювала π . Відстань d приймала 160 різних значень з кроком 0.00125. Кількість тестових точок становила: 2656 точок на відрізку і 1440 – на дузі кола. Як бачимо, при зменшенні відстані d до системи вихрових відрізків похибка збільшується і на відстані 0.0125 має порядок $3,2 \cdot 10^{-3}$ для 16 вихрових відрізків. Розрахунки показали, що при $d > 0.2$ похибка змінюється мало і її порядок становить 10^{-5} . Видно, що при збільшенні кількості вихрових відрізків і наближенні до них похибка збільшується.

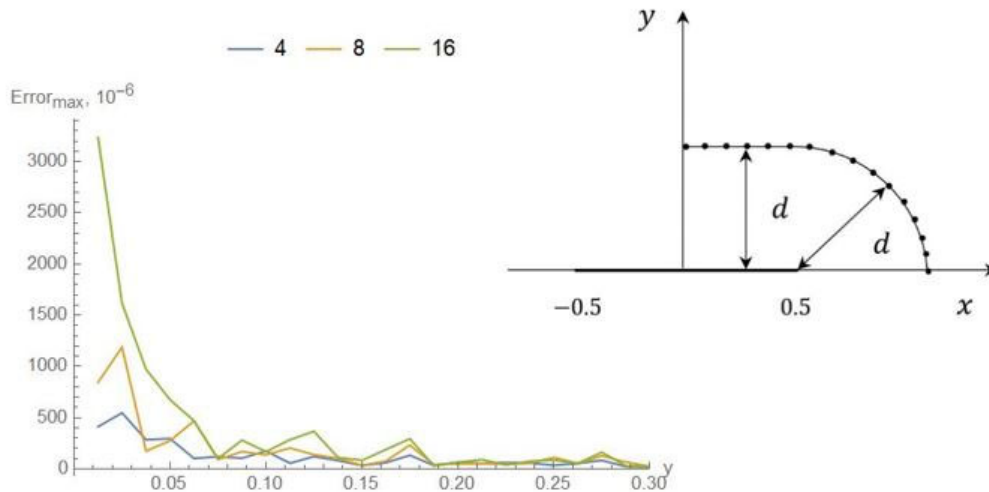


Рис. 1 – Залежність максимальної похибки в тестових точках від відстані $d = y$ до системи вихрових відрізків.

Цікавим було з'ясувати, в яких же точках похибка буде максимальною. Було встановлено, що найбільша похибка спостерігається в точках, які знаходяться майже на одній осі із системою вихрових відрізків. На рис. 2 показана залежність полярного кута φ тестових точок, в яких похибка максимальна, від відстані до вихрових відрізків. Видно, що найбільша похибка спостерігається в точках, які знаходяться майже на одній осі із системою вихрових відрізків.

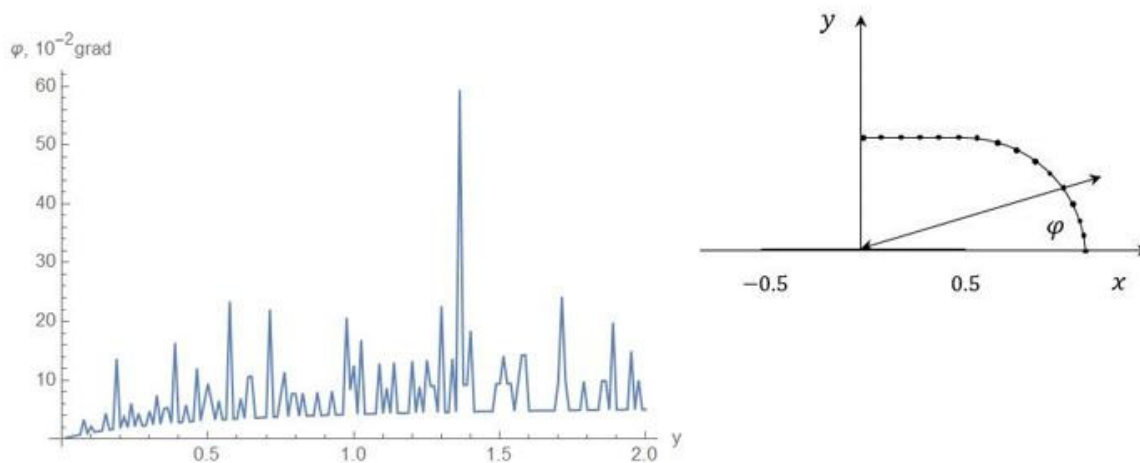


Рис. 2 – Залежність полярного кута φ точки з максимальною похибкою від відстані $d = y$ до системи із 16 вихрових відрізків.

Була досліджена похибка обчислень на GPU при моделюванні обтікання пластини методом дискретних вихрових рамок. На рис. 3 зображена залежність максимальної відносної похибки швидкості в тестових точках при обтіканні пластини (сторонами пластини є відрізки $x = [-0.5, 0.5]$, $y = [-0.5, 0.5]$). Швидкість потоку нормувалась і становила по модулю $v_\infty = 1$. Площина із тестовими точками була перпендикулярна потоку, паралельна пластинці і знаходиться від неї на відстані $d = |z|$ у вихровому сліді (координата z є від'ємною). Час розрахунку становив $t = 0.5$ (з пластинки зійшло 5 рядів рамок). Кількість вихрових відрізків на пластинці та пелені сумарно становила 420. Сторони квадрата, в якому були розміщені тестові точки, було задано в інтервалах $x \in [-2, 2]$, $y \in [-2, 2]$.

Різним кольором зображені результати для різної кількості точок на тестовій площині: 256, 1024 і 4096. Із рис. 3 видно, що максимальна похибка має порядок 10^{-5} в області пелені і при віддаленні тестових точок від пластинки зменшується.

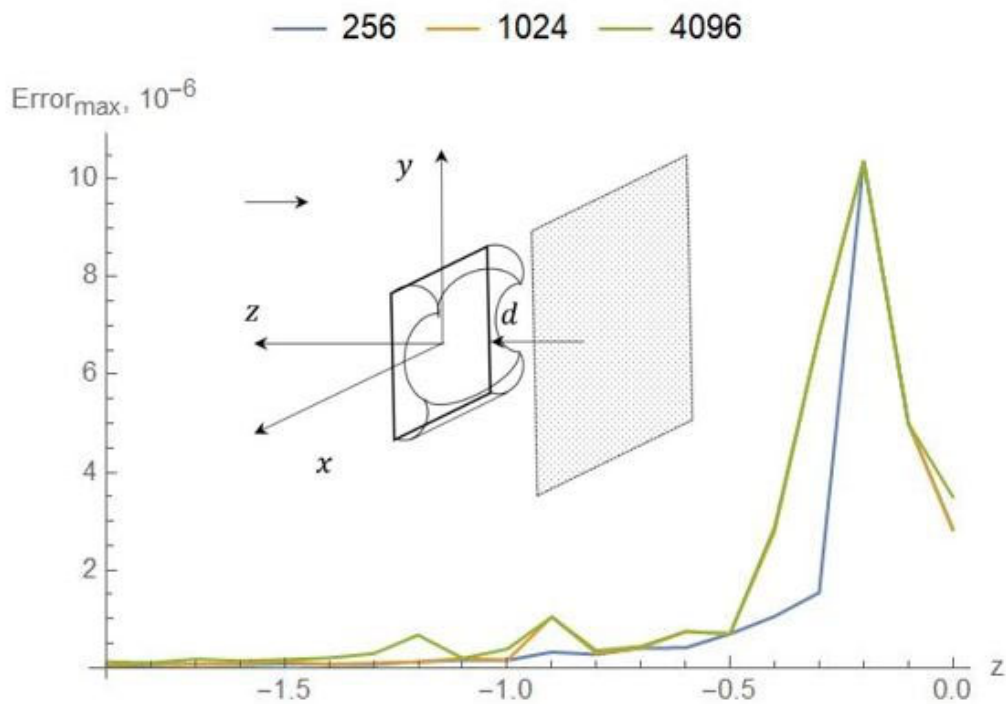


Рис. 3 – Залежність максимальної похибки в тестових точках від відстані $d = z$ до пластинки при різній кількості точок.

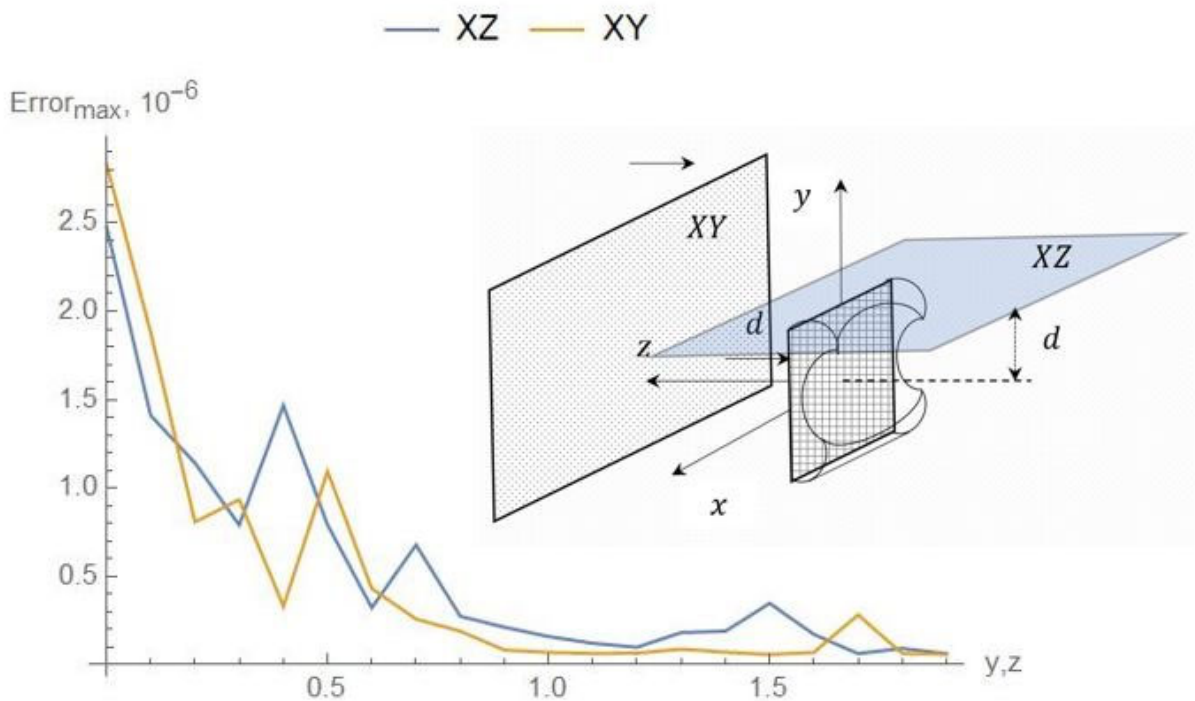


Рис. 4 – Залежність максимальної похибки в тестових точках від відстані d до пластинки або до площини XZ .

На рис. 4 зображена залежність максимальної відносної похибки результатів обчислення на GPU швидкості в тестових точках, що розміщувались у двох площинах біля пластини, перпендикулярної потоку. Тестові точки розміщувались двома способами:

- на вертикальній площині XY , яка паралельна до пластинки і знаходиться від неї на відстані $d = z$ по осі

z (перед пластинкою, з боку набігаючого потоку). Розмір області в площині XY , де розміщувались точки, брався як і для попереднього випадку – $x \in [-2, 2]$, $y \in [-2, 2]$.

– на горизонтальній площині XZ , що перпендикулярна до пластинки і знаходилась на відстані $d = y$ по осі y . Кількість тестових точок становила $265 * 16$. Із рис. 4 видно, що максимальна відносна похибка не перевищує 10^{-5} .

Перспективи подальших досліджень. Автори вважають, що дані дослідження можуть бути використані для розробки ефективної методики моделювання на GPU обтікання будинків в житлових масивах. Можливість перенесення розрахунків на графічний процесор і розпаралелювання дасть змогу обраховувати великі житлові квартали і проектувати оптимальне розміщення будинків з точки зору комфортного проживання населення.

Висновки. У роботі представлені результати дослідження щодо точності розрахунків на GPU з використанням бібліотеки OpenTK на мові C# при розрахунках поля швидкостей за формулою Біо – Савара. Було розглянуто дві системи вихрових відрізків: відрізки, що лежать на симетричному відносно початку координат відрізьку та відрізки, які виникають при моделюванні обтікання пластинки, що встановлена перпендикулярно до набігаючого потоку при моделюванні методом дискретних вихрових рамок. У випадку вихрових відрізків на відрізьку система вихрових точок задавалась таким чином, щоб відстань від тестових точок до відрізьку з вихровими відрізьками змінювалась. У випадку пластинки, точки розміщувались на площині, а відстань від площини до пластинки варіювалась. Обмеження на швидкість (радіус Ренкіна) не вводилось. Перевірка точності здійснювалась шляхом визначення максимальної відносної похибки в усіх тестових точках при обчисленнях з подвійною точністю на центральному процесорі та звичайною точністю на графічному процесорі. Аналізувалась залежність максимальної відносної похибки від відстані до системи вихрових відрізків.

Із результатів дослідження можна зробити наступні висновки:

1. Розрахунки на графічних процесорах швидкості від системи вихрових відрізків вносять відносну похибку, розмір якої для різних вихрових систем, що складаються із вихрових відрізків, знаходиться в діапазоні $10^{-3} \div 10^{-6}$.
2. При наближенні до системи вихрових відрізків відносна похибка істотно збільшується.
3. При віддаленні від системи вихрових відрізків відносна похибка зменшується до певної величини і далі зменшується дуже повільно.
4. Найбільша похибка спостерігається в точках, які знаходяться близько до кінців вихрових відрізків і лежать з ними майже на одній осі.
5. При використанні радіуса Ренкіна можна припустити, що похибка в ближній зоні не буде перевищувати 10^{-6} .

Дана методика є ефективною при моделюванні методом дискретних вихрових рамок і забезпечує достатню точність при знаходженні швидкості. Може бути використана для пришвидшення розрахунків приблизно у 20 – 30 разів.

Список літератури

1. Довгий С. О., Буланчук Г. Г., Буланчук О. М. Моделювання динаміки вихрових структур у середовищі Unity з використанням комп'ютерних шейдерів // Сьома міжнародна науково-практична конференція “Комп'ютерна гідромеханіка”, 29 – 30 вересня 2020 р. – Київ, 2020. – С. 29 – 30.
2. Ling Fang Cao. Development of a GPU-accelerated flow simulation method for wind turbine applications // PhD Thesis, University of Sheffield. – 2022. – 237 p.
3. Andrzej Kosior, Henryk Kudela. Parallel computations on GPU in 3D using the vortex particle method // Computers & Fluids. – 2013. – № 80 (2013). – P. 423 – 428. <https://doi.org/10.1016/j.compfluid.2012.01.014>.
4. Chloé Mimeau, Iraj Mortazavi. A Review of Vortex Methods and Their Applications: From Creation to Recent Advances // Fluids. – 2021. Vol. 6(2), 68. <https://doi.org/10.3390/fluids6020068>.
5. Mauricio Alfredo Vines Neuwirth. Vortex Methods for Fluid Simulation in Computer Graphics // Ph.D. Thesis, University of Ottawa. – Ottawa, 2013. – 183 p.
6. Shiyong Xiong, Rui Tao, Yaorui Zhang, Fan Feng, Bo Zhu. Incompressible Flow Simulation on Vortex Segment Clouds // ACM Trans. Graph. – 2021. – Vol. 40. – issue 4. – Article 98. – P. 1–12. <https://doi.org/10.1145/3450626.3459865>.
7. Вольф Д. Язык шейдеров. Книга рецептов / Пер. с англ. А. Н. Киселева. – М. : ДМК Пресс, 2015. – 368 с.
8. Довгий С. А., Лифанов И. К. Методы решения интегральных уравнений. – К. – Наукова думка, 2002. – 343 с.

References (transliterated)

1. Dovgiiy S. O., Bulanchuk G. G., Bulanchuk O. M. Modelyuvannya dynamiky vykhrovykh struktur u seredovyshhi Unity z vykorystannyyam komp'yuternykh sheyderiv [Modeling the dynamics of vortex structures in the Unity environment using computer shaders]. *S'oma mizhnarodna naukovo-praktychna konferentsiya "Komp'yuterna gidromekhanika"*, 29 – 30 veresnya 2020 r. [Seventh International Scientific and Practical Conference "Computational Hydromechanics", September 29-30, 2020]. Kyiv, 2020. pp. 29–30.
2. Ling Fang Cao. *Development of a GPU-accelerated flow simulation method for wind turbine applications*. PhD Thesis, University of Sheffield. 2022. 237 p.
3. Andrzej Kosior, Henryk Kudela. Parallel computations on GPU in 3D using the vortex particle method. *Computers & Fluids*. 2013, vol. 80 (2013), pp. 423–428. <https://doi.org/10.1016/j.compfluid.2012.01.014>.

4. Chloé Mimeau, Iraj Mortazavi. A Review of Vortex Methods and Their Applications: From Creation to Recent Advances. *Fluids*. 2021, vol. 6(2), 68. <https://doi.org/10.3390/fluids6020068>.
5. Mauricio Alfredo Vines Neuwirth. *Vortex Methods for Fluid Simulation in Computer Graphics*. Ph.D. Thesis, University of Ottawa. Ottawa, 2013. 183 p.
6. Shiyong Xiong, Rui Tao, Yaorui Zhang, Fan Feng, Bo Zhu. Incompressible Flow Simulation on Vortex Segment Clouds. *ACM Trans. Graph.* 2021, vol. 40, issue 4, Article 98, pp. 1–12. <https://doi.org/10.1145/3450626.3459865>.
7. Wolf D. *Yazyk sheyderov. Kniga retseptov* [The language of shaders. Book of recipes]. (Rus. ed. Kiselev A. N.) Moscow, DMK Press Publ., 2015. 368 p.
8. Dovgiy S. A., Lyfanov I. K. *Metody resheniya integral'nykh uravneniy* [Methods of solving integral equations]. Kyiv, Naukova dumka Publ., 2002. 343 p.

Надійшла (received) 10.04.2023

Відомості про авторів / Сведения об авторах / Information about authors

Довгий Станіслав Олексійович – доктор фізико-математичних наук, професор, академік НАН України, президент Малої академії наук України, м. Київ; тел.: (044) 489-55-99; e-mail: s.dovgii@gmail.com.

Довгий Станіслав Алексеевич – доктор физико-математических наук, профессор, академик НАН Украины, президент Малой Академии Наук Украины, г. Киев; тел.: (044) 489-55-99; e-mail: s.dovgii@gmail.com.

Dovgiy Stanislav Oleksiiovych – Doctor of Physical and Mathematical Sciences, Professor, Academician of the National Academy of Sciences of Ukraine, President of the Junior Academy of Sciences of Ukraine, Kyiv; tel.: (044) 489-55-99; e-mail: s.dovgii@gmail.com.

Буланчук Галина Григорівна – кандидат фізико-математичних наук, доцент кафедри вищої та прикладної математики, ДВНЗ «Приазовський державний технічний університет», м. Дніпро; тел.: (098) 201-83-08; e-mail: ggbulan7@gmail.com.

Буланчук Галина Григорьевна – кандидат физико-математических наук, доцент кафедры высшей и прикладной математики, ГВНЗ «Приазовский государственный технический университет», г. Днепр; тел.: (098) 201-83-08; e-mail: ggbulan7@gmail.com.

Bulanchuk Galyna Hryhorivna – Candidate of Physical and Mathematical Sciences, Associate Professor, Pryazovskiy State Technical University, Dnipro; tel.: (098) 201-83-08; e-mail: ggbulan7@gmail.com.

Буланчук Олег Миколайович – кандидат фізико-математичних наук, доцент, методист лабораторії математичних наук, Науково-методичний центр НЦ «Мала академія наук України», м. Київ; тел.: (096) 347-56-60; e-mail: obulan65@gmail.com.

Буланчук Олег Николаевич – кандидат физико-математических наук, доцент, методист лаборатории математических наук, Научно-методический центр НЦ «Малая академия наук Украины», г. Киев; тел.: (096) 347-56-60; e-mail: obulan65@gmail.com.

Bulanchuk Oleh Mykolayovych – Candidate of Physical and Mathematical Sciences, Associate Professor, Methodologist at the Laboratory of Mathematical Sciences, Scientific and Methodological Center of the National Center of the Small Academy of Sciences of Ukraine, Kyiv; tel.: (096) 347-56-60; e-mail: obulan65@gmail.com.

Листопадава Валентина Вікторівна – кандидат фізико-математичних наук, доцент, доцент кафедри математичної фізики та диференціальних рівнянь, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ; тел.: (045) 204-82-46; e-mail: listopadovavv@gmail.com.

Листопадава Валентина Викторовна – кандидат физико-математических наук, доцент, доцент кафедры математической физики и дифференциальных уравнений, Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», г. Киев; тел.: (045) 204-82-46; e-mail: listopadovavv@gmail.com.

Lystopadova Valentyna Viktorivna – Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor at the department of Mathematical Physics and Differential Equations, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv; tel.: (045) 204-82-46; e-mail: listopadovavv@gmail.com.