Міністерство освіти і науки України

Київський національний університет імені Тараса Шевченка

Кафедра обчислювальної математики факультету кібернетики

**Паралельні обчислення у методі ‘сіток’ розв’язання еліптичних рівнянь на комп’ютері гібридної архітектури**

**Текстова частина до курсової роботи**

**за спеціальністю „Прикладна математика” 6.040301**

#### Керівник курсової роботи

доктор фіз.-мат. наук

Хіміч Олександр Михайлович

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2015 р.

Виконав студент

Оленченко Ілля Андрійович

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2015 р.

Київ 2015

**TODO в текстову частину**

I) Текстова частина (ТЧ):

0) общее оформление http://web.znu.edu.ua/lab/fordep/oformlenie/diplom\_specialista.htm

1) прочитать как должно быть оформлена работа

2) індивідуальне завдання на КР(МР);

3) календарний план;

4) зміст;

5) анотація 1 ст. ;

6) вступ 1-3 ст.;

7) основна частина (її розділи): 30-50 ст.;

– аналіз існуючих методів (алгоритмів) вирішення поставленої задачі;

– обґрунтування вибору рішення;

– вибір принципу дії системи чи обґрунтування методик;

– розробка структурної і (або) функціональної схеми;

– розробка принципової схеми;

– експериментальні дослідження;

– метрологічні характеристики;

– алгоритмічне та програмне забезпечення;

8) висновки;

9) література;

10) глосарій;

11) додатки;

II) носій інформації на якому розміщені текстова частина роботи, програми, матеріали та презентація доповіді.

Наведені заголовки основної частини є рекомендованими для КР(МР). Основна частина КР(МР) повинна мати аналітично-розрахунковий характер.

Дозволяється вводити нові частини за вказівками керівника.

Кожний розділ ТЧ може складатися з підрозділів, пунктів, підпунктів, параграфів.

Обсяг ТЧ повинен складати 30-50 сторінок машинописного тексту на аркушах фор­мату А4, причому об’єм основної частини повинен складати не мен­ше 70 % всієї роботи.

Вступ

В сучасний період розвитку обчислювальної техніки актуальність числових методів, що дозволяють розв’язувати широкий клас задач за допомогою ЕОМ, продовжує зростати. Особливо гостро постає питання оптимального використання усіх запропонованих компонентів з архітектури комп’ютера. За останні 10 років розвиток графічних процесорів (GPU) при використання у сукупності із центральним процесором (CPU) нестримними темпами відкриває все нові межі використання ЕОМ. Актуальність роботи зумовлена нестримним рухом з плином часу. На вже розв’язані задачі можна подивитися під іншим кутом, а саме використання графічних процесорів разом з центральним для розв’язання диференціальних рівнянь.

Архітектура комп’ютера від лише послідовної роботи вже давно відійшла, бо очікування монопольного процесу, який зайняв весь можливий простір і не використовує його за назначенням є не раціональним. Зараз ми маємо багатопроцесорні системи, які можуть виконувати декілька процесів одночасно. Проте кількість процесорів у ЦПУ достатньо обмежена, тоді як ГПУ дає можливості дуже великої кількості процесорів, а саме виконання деяких дій одночасно. Звичайно використання лише однієї частини комп’ютера гібридної архітектури MIMD не може дати оптимального часу, тож є сенс для розв’язання такої задачі використовувати обидва процесора.

//тут еще описать СИМД СИСД МИСД и МИМД их разницу.

Розвиває та покращує ринок ГПУ на сьогодні компанія NVidia. Її розробка CUDA дає можливості використовувати ГПУ разом з ЦПУ. Різниця між двома підходами полягає у тому, що маючи вектор чи матрицю значень, програма не буде послідовно обчислювати потрібні значення (як ЦПУ) а візьме цілий вектор (матрицю) і розрахує наступне наближення одразу для всіх елементів, чим і буде прискорений час обчислення.

Розв’язання диференціальних рівнянь завжди було суттєвою проблемою багатьох задач з моменту існування таких задач. Розв’язуючи ту чи іншу реальну проблему за допомогою математики дослідники будують математичні моделі, які в свою чергу у багатьох випадках зводяться до розв’язання диференціального рівняння бо саме диференційні рівняння краще за будь які інші окреслюють суть процесу.

Було б чудово, якщо ЕОМ мали засоби вирішувати такий великий клас задач аналітично, проте саме лише використання машини для пошуку розв’язка змушує нас відмовитися від точних розв’язків на користь наближених в дякому наборі точок, через те що ЕОМ не вміє вирішувати такі рівняння аналітично. Замість точної задачі можна використати наближення диференційного оператора у вигляді різницевої схеми, поставити йому у відповідність граничні початкові умови і знайти наближений розв’зок розв’язавши систему лінійних алгебраїчних рівнянь.

Такі алгоритми передбачають розв’язання задач із потрібною точністю та будь якою кількістю потрібних точок, де ми можемо знайти наближений розв’язок. Список методів для розв’язання яких можемо перераховувати достатньо довго (метод Гауса, метод Гауса-Жордана, метод Гауса-Зейделя, матричний метод розв’язання систем лінійних алгебраїчних рівнянь, метод квадратного кореня, метод Крамера, метод прогонки, метод Якобі, метод релаксації, розгалуження Холецького, проекційні методи, метод регуляризації Тихонова, ітераційні методи), обирання того чи іншого метода залежить від вигляду отриманої системи її властивостей, бажаної швидкості знаходження розв’язку. Для використання методу на ЕОМ, маючи різницеву залежність для знаходження наближеного розв’язку, ітеративні матимуть перевагу при розпаралелюванні. Кожний наступна ітерація знаходиться з використанням попередніх обчислень, доки процес не зійдется в деякій точці. Гарантія збіжності забезпечується кроками методу та початковим наближенням. Такі методи, маючи коректну постановку збігаються при достатній кількості операцій. Тож головне питання для ітеративного процесу складається у питанні часу, бо в деяких умовах використання для швидкого обчислення у реальному часі, при надточних обчисленнях від яких можуть залежати не тільки отриманий розв’язок статки, а навіть життя людини.

Для кожного методу можна написати оптимальну паралельну оптимізацію, яка має можливості покращити час виконання операцій.

Суб’єктом роботи є сіткові методи накладені на еліптичні рівняння поставлені як задача Діріхлє.

Метою роботи є отримання результату оптимізації за допомогою КУДА та надання методів паралельного обчислювання таких задач.

Аналіз існуючих методів (алгоритмів) вирішення поставленої задачі