Министерство образования Республики Беларусь

Учреждение Образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Архитектура вычислительных систем»

«К защите допустить»

Руководитель курсовой работы

***должность***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ***Фамилия И.И***

\_\_.\_\_.2019

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему:

***«РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЕ С ПОМОЩЬЮ ТЕХНОЛИГИИ ‘CUDA’»***

Выполнил студент группы 753505

Таланец Артём Витальевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

Курсовая работа представлен на

Проверку \_\_.\_\_.2019

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

Минск 2019

Содержание

Предисловие3

1. **Постановка задачи**4
2. **Исторические сведения**5
3. **Теоретическая часть**6
   1. СЛАУ6
   2. CUDA-1
4. **Основная часть**-1
   1. dfs-1
   2. dfs-1

Вывод-1

Литература -1

## Предисловие

Решение систем линейных алгебраических уравнений - одна из основных задач вычислительной линейной алгебры. Хотя задача решения именно системы линейных уравнений сравнительно редко представляет самостоятельный интерес для прикладных задач, но от умения эффективно решать данные системы часто зависит сама возможность математического моделирования самых разнообразных процессов с применением ЭВМ. Значительная часть численных методов решения различных (в особенности - нелинейных) задач включает в себя решение систем линейных уравнений как элементарный шаг соответствующего алгоритма.

Данный курсовой проект описывает особенности решения СЛАУ с помощью технологии CUDA, реализацию алгоритмов решения и их тестирование.

## Постановка задачи

**Цель** **работы:** реализация и тестирование алгоритма решения СЛАУ на аппаратно-программной платформе CUDA.

**Объект исследования:** СЛАУ.

**Предмет исследования:** решения СЛАУ с помощью технологии CUDA.

**Задачи:**

Рассмотреть понятие СЛАУ и способы их решения. Проанализировать особенности решения систем линейных алгебраических уравнений на CUDA. Реализовать и протестировать алгоритмы решения СЛАУ на CUDA.

## Исторические сведения

Задачи, соответствующие современным задачам на составление и решение систем уравнений с несколькими неизвестными, встречаются еще в вавилонских и египетских рукописях II века до н.э., а также в трудах древнегреческих, индийских и китайских мудрецов. В китайском трактате "Математика в девяти книгах" словесно изложены правила решения систем уравнений, были замечены некоторые закономерности при решении.

Идею общего метода решения систем линейных уравнений высказал Лейбниц в 1693 году. Она была реализована швейцарским математиком Крамером в 1752 году. Он сформулировал и обосновал правило, носящее теперь его имя, которое позволяет решать системы n линейных уравнений с n неизвестными и буквенными коэффициентами. По правилу Крамера каждая неизвестная равна отношению двух определителей. Крамер, фактически, заложил основы теории определителей, хотя и не предложил для них удобного обозначения (это сделал в 1841 году А. Кэли). В 1772 году Вандермонд опубликовал обширное исследование определителей, один из которых носит теперь его имя. Систематическое изложение этой теории принадлежит Бине и Коши. Их труды по теории определителей относятся к периоду 1812-1815 гг.

## Теоретическая часть

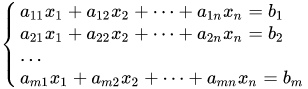
## СЛАУ

Система линейных алгебраических уравнений (линейная система, также употребляются аббревиатуры СЛАУ, СЛУ) — система уравнений, каждое уравнение в которой является линейным — алгебраическим уравнением первой степени.

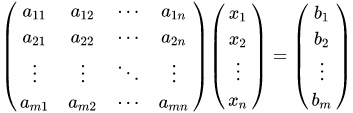
В классическом варианте коэффициенты при переменных, свободные члены и неизвестные считаются вещественными числами, но все методы и результаты сохраняются (либо естественным образом обобщаются) на случай любых полей, например, комплексных чисел.

Решение систем линейных алгебраических уравнений — одна из классических задач линейной алгебры, во многом определившая её объекты и методы. Кроме того, линейные алгебраические уравнения и методы их решения играют важную роль во многих прикладных направлениях, в том числе в линейном программировании, эконометрике.

Общий вид системы линейных алгебраических уравнений:

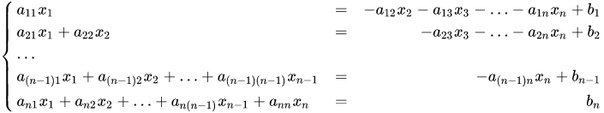


Система линейных алгебраических уравнений может быть представлена в матричной форме как:



или:

https://sun9-51.userapi.com/c854528/v854528135/19a878/kY3w6Wcyp3Q.jpg

Одним из классических итерационных методов решения СЛАУ является метод Гаусса-Зейделя. Чтобы пояснить суть метода перепишем задачу в виде 

Эта запись может быть представлена как

https://sun9-18.userapi.com/c854528/v854528665/19d47e/Uqm2PhCZOSE.jpg

Итерационный процесс в методе Гаусса — Зейделя строится по формуле

https://sun9-51.userapi.com/c854528/v854528665/19d4a1/5Cq05Rmr68M.jpg

после выбора соответствующего начального приближения https://sun9-44.userapi.com/c854528/v854528665/19d4a6/N3PfhsIkqBw.jpg.

Метод Гаусса — Зейделя можно рассматривать как модификацию метода Якоби. Основная идея модификации состоит в том, что новые значения https://sun9-62.userapi.com/c854528/v854528665/19d4ad/FVgmPSyBlHY.jpgиспользуются здесь сразу же по мере получения, в то время как в методе Якоби они не используются до следующей итерации.

## CUDA

CUDA (Compute Unified Device Architecture) — программно-аппаратная архитектура параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность благодаря использованию графических процессоров фирмы Nvidia.

CUDA SDK позволяет программистам реализовывать на специальных упрощённых диалектах языков программирования Си, C++ и Фортран алгоритмы, выполнимые на графических и тензорных процессорах Nvidia. Архитектура CUDA даёт разработчику возможность по своему усмотрению организовывать доступ к набору инструкций графического или тензорного ускорителя и управлять его памятью. Функции, ускоренные при помощи CUDA, можно вызывать из различных языков, в т.ч. Python, MATLAB и т.п.

Первоначальная версия CUDA SDK была представлена 15 февраля 2007 года. В основе интерфейса программирования приложений CUDA лежит язык Си с некоторыми расширениями. Для успешной трансляции кода на этом языке в состав CUDA SDK входит собственный Си-компилятор командной строки nvcc компании Nvidia. Компилятор nvcc создан на основе открытого компилятора Open64 и предназначен для трансляции host-кода (главного, управляющего кода) и device-кода (аппаратного кода) (файлов с расширением .cu) в объектные файлы, пригодные в процессе сборки конечной программы или библиотеки в любой среде программирования, например, в NetBeans.

В архитектуре CUDA используется модель памяти грид, кластерное моделирование потоков и SIMD-инструкции. Применима не только для высокопроизводительных графических вычислений, но и для различных научных вычислений с использованием видеокарт nVidia. Учёные и исследователи широко используют CUDA в различных областях, включая астрофизику, вычислительную биологию и химию, моделирование динамики жидкостей, электромагнитных взаимодействий, компьютерную томографию, сейсмический анализ и многое другое. В CUDA имеется возможность подключения к приложениям, использующим OpenGL и Direct3D. CUDA — кроссплатформенное программное обеспечение для таких операционных систем, как Linux, Mac OS X и Windows.

22 марта 2010 года nVidia выпустила CUDA Toolkit 3.0, который содержал поддержку OpenCL.

Платформа CUDA впервые появились на рынке с выходом чипа NVIDIA восьмого поколения G80 и стала присутствовать во всех последующих сериях графических чипов, которые используются в семействах ускорителей GeForce, Quadro и NVidia Tesla.

Первая серия оборудования, поддерживающая CUDA SDK, G8x, имела 32-битный векторный процессор одинарной точности, использующий CUDA SDK как API (CUDA поддерживает тип double языка Си, однако сейчас его точность понижена до 32-битного с плавающей запятой). Более поздние процессоры GT200 имеют поддержку 64-битной точности (только для SFU), но производительность значительно хуже, чем для 32-битной точности (из-за того, что SFU всего два на каждый потоковый мультипроцессор, а скалярных процессоров — восемь). Графический процессор организует аппаратную многопоточность, что позволяет задействовать все ресурсы графического процессора. Таким образом, открывается перспектива переложить функции физического ускорителя на графический ускоритель (пример реализации — PhysX). Также открываются широкие возможности использования графического оборудования компьютера для выполнения сложных неграфических вычислений: например, в вычислительной биологии и в иных отраслях науки.

По сравнению с традиционным подходом к организации вычислений общего назначения посредством возможностей графических API, у архитектуры CUDA отмечают следующие преимущества в этой области:

* Интерфейс программирования приложений CUDA (CUDA API) основан на стандартном языке программирования Си с некоторыми ограничениями. По мнению разработчиков, это должно упростить и сгладить процесс изучения архитектуры CUDA
* Разделяемая между потоками память (shared memory) размером в 16 Кб может быть использована под организованный пользователем кэш с более широкой полосой пропускания, чем при выборке из обычных текстур
* Более эффективные транзакции между памятью центрального процессора и видеопамятью
* Полная аппаратная поддержка целочисленных и побитовых операций
* Поддержка компиляции кода GPU средствами открытого проекта LLVM

## Основная часть

## Графические представления поточечной и равномерной сходимости функциональных рядов

## Вывод

## Список использованных источников

1. Числовые ряды [Электронный ресурс]: Студенческая библиотека онлайн – режим доступа к библиотеке: <https://studbooks.net/2258064/matematika_himiya_fizika/chislovye_ryady>
2. Свободная энциклопедия Википедия, статья "Дирихле, Петер Густав Лежён" [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%80%D0%B8%D1%85%D0%BB%D0%B5,\_%D0%9F%D0%B5%D1%82%D0%B5%D1%80\_%D0%93%D1%83%D1%81%D1%82%D0%B0%D0%B2\_%D0%9B%D0%B5%D0%B6%D1%91%D0%BD](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%80%D0%B8%D1%85%D0%BB%D0%B5,_%D0%9F%D0%B5%D1%82%D0%B5%D1%80_%D0%93%D1%83%D1%81%D1%82%D0%B0%D0%B2_%D0%9B%D0%B5%D0%B6%D1%91%D0%BD%20)
3. Свободная энциклопедия Википедия, статья "Функция Дирихле" [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_%D0%94%D0%B8%D1%80%D0%B8%D1%85%D0%BB%D0%B5>
4. Жевняк Р.М., Карпук А.А., Высшая математика: Дифференцильные уравнения. Ряды. Уравнения математической физики. Теория функций комплексной переменной/Р.М. Жевняк, А.А. Карпук: Учебное пособие — Мн.: ИРФ “Обозрение”, 1997. — 570 с.:ил.
5. А.В. Игнатьева, Т.И. Краснощекова, В.Ф. Сминов, Курс высшей математики: Учебное пособие —Москва, И-51, Издательство «Высшая школа», 1966. — 692 с.:ил.