

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №4
по курсу «Программирование графических процессоров»**

Работа с матрицам. Метод Гаусса.

Выполнил: А.В. Куликов
Группа: 8О-408Б
Преподаватели: К.Г. Крашенинников,
А.Ю. Морозов

Москва, 2020

Условие

Цель работы: Использование объединения запросов к глобальной памяти.

Реализация метода Гаусса с выбором главного элемента по столбцу. Ознакомление с библиотекой алгоритмов для параллельных расчетов Thrust.

Вариант 4. Необходимо вычислить LU-разложение квадратной матрицы: $A = LU$, где A – матрица $n \times n$, L – нижняя треугольная матрица, с единичными элементами на диагонали, U – верхняя треугольная матрица. Дополнительно нужно получить вектор перестановок строк p , где $p[i]$ содержит номер строки с которой произошла перестановка на i -ой итерации.

Входные данные. На первой строке задано число n – размер матрицы. В следующих n строках, записано по n вещественных чисел – элементы матрицы. $n \leq 8 * 10^3$.

Выходные данные. Необходимо вывести на n строках, по n чисел – элементы матриц L и U объединенные в одну матрицу. Далее записываются n элементов вектора перестановок p .

Программное и аппаратное обеспечение

Видеокарта	GeForce GTX 1650
Compute capability	7.5
Графическая память	3911 Мб
Разделяемая память	48 Кб
Константная память	64 Кб
Количество регистров на блок	65536
Максимальное кол-во блоков	$2147483647 * 65535 * 65535$
Максимальное кол-во нитей в блоке	1024
Кол-во мультипроцессоров	16
Ядер CUDA	896

Процессор	AMD Ryzen 5 3550H
ОЗУ	8 Гб
ЖД	

Операционная система	Ubuntu 20.04.6 LTS
IDE	VS Code
Компилятор	nvcc V10.1, mpi V3.3.2

Метод решения

Задача решается все тем же способом, что и обычное LUP-разложение на CPU за исключением того, что все возможные операции (поиск максимума, перестановка строк, пересчет подматрицы), составляющие каждую итерацию, проводятся на GPU, остальные действия происходят на CPU.

Описание программы

В файле main.cu находится основная логика программы, включая ввод/вывод, работа с памятью, а также сама функция, реализующая алгоритм LUP-разложения.

В файле error.h находится функционал для обработки ошибок CUDA API и хендлеры для стандартных сигналов SIGSEGV, SIGABRT.

Результаты

Тестирование ядер с различными конфигурациями

Маленькая матрица 10x10

Параметры сетки	Время исполнения
16 блоков 16 потоков	0m 0,257s
64 блоков 64 потоков	0m 0,261s
256 блоков 256 потоков	0m 0,269s
1024 блоков 1024 потоков	0m 0,255s

Лучший результат: 0m 0,255s при 1024 блоках, 1024 потоках.

Средняя матрица 1000x1000

Параметры сетки	Время исполнения
16 блоков 16 потоков	0m 1,473s
64 блоков 64 потоков	0m 1,103s
256 блоков 256 потоков	0m 1,089s
1024 блоков 1024 потоков	0m 1,065s

Лучший результат: 0m 1,065s при 1024 блоках, 1024 потоках.

Большая матрица 10000x10000

Параметры сетки	Время исполнения
16 блоков 16 потоков	9m 51,792s
64 блоков 64 потоков	3m 28,835s
256 блоков 256 потоков	2m 56,425s
1024 блоков 1024 потоков	2m 29,722s

Лучший результат: 2m 29,722s при 1024 блоках, 1024 потоках.

Сравнение с CPU

Маленькая матрица 10x10

Результат: 0m 0,002s

Средняя матрица 1000x1000

Результат: 0m 1,367s

Большая матрица 10000x10000

Результат: 10m 4,027s

Выводы

Данный алгоритм используется для в решении СЛАУ, которые, в свою очередь, возникают во многих задачах численных методов.

Особых проблем, кроме глупых ошибок, в решении задачи не возникало. В решении данной задачи программа с использованием GPU значительно превзошла реализацию, работающую только на CPU.