

**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 процессов»**

**Работа с матрицами. Метод Гаусса**

Выполнил: Днепров И. С.

Группа: 8О-407Б-17

Преподаватели: К.Г. Крашенинников,  
А.Ю. Морозов

Москва, 2021

## 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*.

В качестве вещественного типа данных необходимо использовать тип данных *double*. Библиотеку *Thrust* использовать только для поиска максимального элемента на каждой итерации алгоритма. В местах, где необходимо сравнение по модулю с нулем, в качестве нулевого значения использовать  $10^{-7}$ . Все результаты выводить с относительной точностью  $10^{-10}$ .

### Вариант 3. Решение квадратной СЛАУ

Необходимо решить систему уравнений  $Ax = b$ , где  $A$  – квадратная матрица  $n \times n$ ,  $b$  – вектор-столбец свободных коэффициентов длиной  $n$ ,  $x$  – вектор неизвестных.

### Входные данные:

На первой строке задано число  $n$  – размер матрицы. В следующих  $n$  строках, записано по  $n$  вещественных чисел – элементы матрицы. Далее, записываются  $n$  элементов вектора свободных коэффициентов.  $n \leq 10^4$ .

### Выходные данные:

Необходимо вывести  $n$  значений, являющиеся элементами вектора неизвестных  $x$ .

### Пример:

Входной файл	Выходной файл
2 1 2 3 4 5 6	-4.0000000000e+00 4.5000000000e+00

## **Программное и аппаратное обеспечение**

### **GPU**

Название: GeForce GT 545

Размер глобальной памяти: 3150381056

Размер константной памяти : 65536

Размер разделяемой памяти: 49152

Регистров на блок: 32768

Максимум потоков на блок: 1024

Размер варпа: 32

Максимальные размеры блока: 1024 x 1024 x 64

Максимальные размеры сетки: 65535 x 65535 x 65535

Количество мультипроцессоров : 3

### **CPU**

Название: Intel Core i7-3770

Частота: 3.40GHz

Размер кеша: 8192 KB

Количество ядер: 4

Количество потоков: 8

### **MEM**

Размер: 16 GB

Тип: ddr3

### **Прочее**

OS: Linux Ubuntu 16.04.6 Редактор: Atom

## Метод решения

Будем хранить матрицу по столбцам, как это принято в Фортране для объединения запросов к глобальной памяти, причём элементы матрицы  $b$  будут располагаться в памяти непосредственно следом за  $A$ , чтобы можно было без лишних затрат работать с расширенной матрицей системы  $(A | b)$ .

Организуем прямой проход метода Гаусса по такой расширенной матрице с выбором главного элемента с помощью метода *max\_element* библиотеки *thrust*.

На каждой итерации алгоритма после успешного выбора главного элемента будем вызывать ядро *direct\_move\_kernel* для обновления расширенной матрицы системы  $(A | b)$ . После прямого хода у нас получится ступенчатая матрица  $A$ .

Далее, организуем обратный ход метода Гаусса для главного минора и вектора  $b$ . В процессе обратного хода приведём матрицу системы сначала к простому виду  $(diag(a_1, a_2, \dots, a_k) | \tilde{b})$ , а затем путём деления строк  $\tilde{b}$  на соответствующие им диагональные элементы – к простейшему виду  $(E | \bar{b})$ .

$\bar{b}$  – и есть искомый вектор.

## Описание программы

Прототип	Тип	Описание
<code>struct abs_comparator</code>	Структура	Компаратор для поиска ведущего элемента, как максимального элемента в подстолбце по модулю
<code>__global__ void swap_rows(double *A, int m, int n, int i1, int i2)</code>	Ядро	Обмен местами строк в матрице $A$ размером $m \times n$ с индексами $i_1$ и $i_2$
<code>__global__ void direct_move_kernel(double *A, int m, int n, int lead_i, int lead_j)</code>	Ядро	Реализация одной итерации прямого хода метода Гаусса вычислением <i>значащих</i> значений в расширенной матрице системы
<code>__global__ void backward_move_kernel(const double *A, int n, double *b, int lead_i, int lead_j)</code>	Ядро	Реализация одной итерации обратного хода метода Гаусса вычислением <i>значащих</i> значений в расширенной матрице системы
<code>__global__ void division_kernel(const double *A, int n, double *b)</code>	Ядро	Деление всех элементов $\tilde{b}$ на ведущий
<code>int main()</code>	Функция	Главная точка входа в приложение

## Результаты

№ теста	Размер матрицы <i>n</i>	Размер сетки блоков		Размер сетки нитей		Время, микросекунды
		<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	
1	5	128	128	32	16	1.523e+03
2	100					2.610e+04
3	10000					1.013e+08
4	5	256	256	32	16	2.309e+03
5	100					6.320e+04
6	10000					9.13e+07
7	5	512	512	32	16	6.287e+03
8	100					1.914e+05
9	10000					8.13e+07
10	5	1024	1024	32	16	2.304e+04
11	100					5.732e+05
12	10000					7.56e+07

## **Выводы**

Получен навык распараллеливания вычислений в алгоритме Гаусса с помощью *CUDA*, изучены вопросы объединения запросов к глобальной памяти. Численно решена задача  $Ax = b$  с помощью этого алгоритма. Во время выполнения лабораторной работы сложностей не возникло.