

## Лабораторная работа №4

### Работа с матрицам. Метод Гаусса.

**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. Использование *двухмерной сетки потоков*. Исследование производительности программы с помощью утилиты nvprof (*обязательно отразить в отчете*).

В качестве вещественного типа данных необходимо использовать тип данных double. Библиотеку Thrust использовать только для поиска максимального элемента на каждой итерации алгоритма. В вариантах(1,5,6,7), где необходимо сравнение по модулю с нулем, в качестве нулевого значения использовать  $10^{-7}$ . Все результаты выводить с относительной точностью  $10^{-10}$ .

#### Вариант 1. Вычисление детерминанта матрицы.

**Входные данные.** На первой строке задано число  $n$  -- размер матрицы. В следующих  $n$  строках, записано по  $n$  вещественных чисел -- элементы матрицы.  $n \leq 10^4$ .

**Выходные данные.** Необходимо вывести одно число -- детерминант матрицы.

**Пример:**

Входной файл	Выходной файл
3 1 2 3 4 5 6 7 8 7	6.0000000000e+00

#### Вариант 2. Вычисление обратной матрицы.

**Входные данные.** На первой строке задано число  $n$  -- размер матрицы. В следующих  $n$  строках, записано по  $n$  вещественных чисел -- элементы матрицы.  $n \leq 10^4$ .

**Выходные данные.** Необходимо вывести на  $n$  строках, по  $n$  чисел -- элементы обратной матрицы.

**Пример:**

Входной файл	Выходной файл
2 1 2 3 4	-2.0000000000e+00 1.0000000000e+00 1.5000000000e+00 -5.0000000000e-01

### 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

### Вариант 4. LU-разложение матрицы.

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  $p$ .

#### Пример:

Входной файл	Выходной файл
2 1 2 3 4	3.0000000000e+00 4.0000000000e+00 3.3333333333e-01 6.6666666667e-01 1 1
3 1 2 3 4 5 6 7 8 7	7.0000000000e+00 8.0000000000e+00 7.0000000000e+00 1.4285714286e-01 8.5714285714e-01 2.0000000000e+00 5.7142857143e-01 5.0000000000e-01 1.0000000000e+00 2 2 2

### Вариант 5. Решение произвольной СЛАУ.

Необходимо найти *любое* решение системы уравнений  $Ax = b$ , где  $A$  -- матрица  $n \times m$ ,  $b$  -- вектор-столбец свободных коэффициентов длиной  $n$ ,  $x$  -- вектор неизвестных длиной  $m$ .

**Входные данные.** На первой строке заданы числа  $n$  и  $m$  -- размеры матрицы. В следующих  $n$  строках, записано по  $m$  вещественных чисел -- элементы матрицы.

Далее записываются  $n$  элементов вектора свободных коэффициентов.  $n * m \leq 10^8$ .

**Выходные данные.** Необходимо вывести  $m$  значений, являющиеся элементами вектора неизвестных  $x$ .

**Пример:**

Входной файл	Выходной файл
2 3 1 2 3 4 5 6 5 14	1.0000000000e+00 2.0000000000e+00 0.0000000000e+00
3 2 1 2 3 4 5 6 7 8 9	-6.0000000000e+00 6.5000000000e+00

### Вариант 6. Нахождение ранга матрицы.

**Входные данные.** На первой строке заданы числа  $n$  и  $m$  -- размеры матрицы. В следующих  $n$  строках, записано по  $m$  вещественных чисел -- элементы матрицы.

$n * m \leq 10^8$ .

**Выходные данные.** Необходимо вывести одно число -- ранг матрицы.

**Пример:**

Входной файл	Выходной файл
2 3 1 2 3 2 4 6	1
3 3 1 2 3 4 5 6 7 8 9	2

### Вариант 7. Решение матричного уравнения.

Необходимо найти *любое* решение матричного уравнения  $AX = B$ , где  $A$  -- матрица  $n \times m$ ,  $X$  -- неизвестная матрица  $m \times k$ ,  $B$  -- матрица  $n \times k$ .

**Входные данные.** На первой строке заданы числа  $n$ ,  $m$  и  $k$  -- размеры матриц. В следующих  $n$  строках, записано по  $m$  вещественных чисел -- элементы матрицы  $A$ . Далее записываются  $n$  строк, по  $k$  чисел -- элементы матрицы  $B$ .

$$n * m + m * k + n * k \leq 1.2 * 10^8.$$

**Выходные данные.** Необходимо вывести на  $m$  строках, по  $k$  чисел -- элементы неизвестной матрицы  $X$ .

**Пример:**

Входной файл	Выходной файл
1 2 1 1 2 5	5.0000000000e+00 0.0000000000e+00
2 2 2 1 2 3 4 1 0 0 1	-2.0000000000e+00 1.0000000000e+00 1.5000000000e+00 -5.0000000000e-01
2 3 2 1 2 3 4 8 6 3 4 7 1	5.0000000000e-01 -3.5000000000e+00 0.0000000000e+00 0.0000000000e+00 8.3333333333e-01 2.5000000000e+00

### Вариант на “два”. Быстрое транспонирование матрицы

Необходимо реализовать транспонирование матрицы с бесконфликтом использованием разделяемой памяти и объединением запросов к глобальной памяти.

**Входные данные.** На первой строке заданы числа  $n$  и  $m$  -- размеры матрицы. В следующих  $n$  строках, записано по  $m$  вещественных чисел -- элементы матрицы.

$$n * m \leq 10^8.$$

**Выходные данные.** Необходимо вывести на  $m$  строках, по  $n$  чисел -- элементы транспонированной матрицы.

**Пример:**

Входной файл	Выходной файл
2 3 1 2 3 4 5 6	1.0000000000e+00 4.0000000000e+00 2.0000000000e+00 5.0000000000e+00 3.0000000000e+00 6.0000000000e+00