



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

## **ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине: «Архитектура параллельных вычислительных систем»

Студент Никифорова Ирина Андреевна

Группа РК6-12М

Тип задания лабораторная работа

Тема лабораторной работы №1. Библиотека OpenMP

Студент Никифорова И.А.  
*подпись, дата* *фамилия, и.о.*

Преподаватель Жук Д.М.  
*подпись, дата* *фамилия, и.о.*

Оценка \_\_\_\_\_

*Москва, 2020 г.*

## Оглавление

Задание на лабораторную работу	3
Цель выполнения лабораторной работы	3
Описание выполненных изменений	4
Замеры производительности	5
Выводы	7
Использованная литература	8

## **Задание на лабораторную работу**

С помощью неявной разностной схемы решить нестационарное уравнение теплопроводности для прямоугольной пластины размером 5\*5см. Начальное значение температуры пластины - 20 градусов.

Граничные условия следующие: левая граница теплоизолирована, на нижней границе поддерживается 20, на остальной части границы температура 800 градусов.

Полученную программу сделать параллельной с помощью библиотеки OpenMP в местах использования метода Гаусса. Изучить влияние количества потоков на скорость выполнения программы.

## **Цель выполнения лабораторной работы**

**Цель выполнения лабораторной работы** – познакомиться с библиотекой OpenMP и попробовать ее использование для прикладной программы.

## Описание выполненных изменений

Для выполнения задания была использована уже написанная ранее (в 6 семестре) программа.

Чтобы использовать библиотеку OpenMP, в программу был добавлен заголовочный файл *omp.h*. Для установки количества используемых потоков использовалась функция *omp\_set\_num\_threads(num\_of\_threads)*.

Параллельно были выполнены два блока программы: прямой и обратный ход метода Гаусса (внутренние циклы). Для этого использовалась директива *#pragma omp parallel for*. Листинги 1 и 2 представляю параллельные участки программы.

### Листинг 1. Прямой ход метода Гаусса

```
template <typename T>
void Matrix<T>::forward_Gauss(Matrix<double>* fr) {
    for (int i = 0; i < _rows - 1; i++) {
        make_diag_one_Gauss(i, fr);
        #pragma omp parallel for
        for (int k = i + 1; k < _rows; k++) {
            T coeff = _arr[k][i];
            subtract_rows_Gauss(i, k, coeff, fr);
        }
    }
    make_diag_one_Gauss(_rows - 1, fr);
}
```

### Листинг 2. Обратный ход метода Гаусса

```
template <typename T>
void Matrix<T>::backward_Gauss(Matrix<double>* fr) {
    for (int i = _rows - 1; i > 0; i--) {
        #pragma omp parallel for
        for (int k = i - 1; k >= 0; k--) {
            T coeff = _arr[k][i];
            subtract_rows_Gauss(i, k, coeff, fr);
        }
    }
}
```

## Замеры производительности

Были проведены измерения скорости работы программы в зависимости от размера задачи и количества потоков. Замеры производились на компьютере с процессором Intel® Core™ i3 CPU M 370 @ 2.40GHz × 4, который имеет 2 ядра и 4 треда (рис. 1).

CPU Specifications	
# of Cores ?	2
# of Threads ?	4
Processor Base Frequency ?	2.40 GHz
Cache ?	3 MB Intel® Smart Cache
Bus Speed ?	2.5 GT/s
TDP ?	35 W

Рис. 1. Характеристики использованного процессора.

Размер задачи менялся следующим образом:

1. сетка 10×10, время 10с;
2. сетка 20×20, время 10с;
3. сетка 20×20, время 20с.

Количество потоков - 1, 2, 4 и 8.

По результатам сделанных замеров была составлена таблица 1.

Таблица 1. Результаты тестов на производительность.

	10×10×10	20×20×10	20×20×20
1	0.212301	10.4079	20.1092
2	0.116146	6.4839	10.8288
4	0.125866	9.11086	8.62454
8	0.211664	8.1251	10.6931

Как видно из таблицы 1, при маленькой размерности задачи существенного увеличения скорости выполнения не наблюдается. При увеличении размерности же, очевидное улучшение появляется при разделении работы на два потока. Далее улучшение либо отсутствует совсем (8 потоков), либо нестабильно (4 потока). Это объясняется архитектурой использованного процессора, в котором 2 физических ядра и 4 виртуальных ядра.

## **Выводы**

В работы была изучена библиотека OpenMP. Она была применена для выполнения теплового расчета. Использование параллельных вычислений для метода Гаусса позволило сократить время работы приблизительно в 2 раза, что при увеличении размерности является существенным преимуществом.

## **Использованная литература**

1. Лекционные материалы по дисциплине «Архитектура параллельных вычислительных систем».
2. <https://ark.intel.com/content/www/us/en/ark/products/49020/intel-core-i3-370m-processor-3m-cache-2-40-ghz.html>
3. [https://pro-prof.com/forums/topic/matrix-triangulation\\_cplusplus](https://pro-prof.com/forums/topic/matrix-triangulation_cplusplus)
4. <https://software.intel.com/content/www/us/en/develop/articles/getting-started-with-openmp.html>