



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 _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 8
по курсу «Численные методы линейной алгебры»
«Метод Штрассена»

Студент группы ИУ9-71Б Баев Д.А

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

1. Реализовать метод Штрассена.
2. Реализовать рекурсию через многопоточность.
3. Сравнить точность результата со стандартным алгоритмом умножения.
4. Построить на одном графике зависимость времени t (сек) умножения двух матриц размера $N \times N$ стандартным алгоритмом, методом Штрассена и методом Штрассена с многопоточностью от размера матрицы N .

2 Исходный код

Исходный код программы представлен в листингах 1– ??.

Листинг 1 — Реализация метода Штрассена

```
1 def strassen(A, B, border=32):
2     if len(A) <= border:
3         return mul_matrix(A, B)
4     A11, A12, A21, A22 = cut_matrix(A)
5     B11, B12, B21, B22 = cut_matrix(B)
6
7     P1 = strassen(A11 + A22, B11 + B22, border)
8     P2 = strassen(A21 + A22, B11, border)
9     P3 = strassen(A11, B12-B22, border)
10    P4 = strassen(A22, B21 - B11, border)
11    P5 = strassen(A11 + A12, B22, border)
12    P6 = strassen(A21 - A11, B11 + B12, border)
13    P7 = strassen(A12 - A22, B21 + B22, border)
14
15
16    C11 = P1 + P4 - P5 + P7
17    C12 = P3 + P5
18    C21 = P2 + P4
19    C22 = P1 + P3 - P2 + P6
20
21    return np.vstack((np.hstack((C11, C12)), np.hstack((C21, C22))))
```

Листинг 2 — Реализация метода Штрассена с многопоточностью

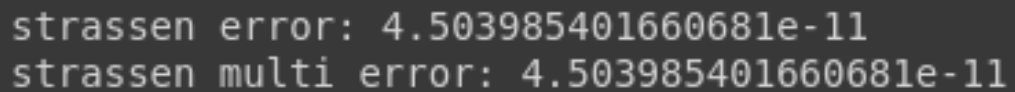
```
1 def strassen_multiprocessing(A, B, border=32):
2     if len(A) <= border:
3         return mul_matrix(A, B)
4     A11, A12, A21, A22 = cut_matrix(A)
5     B11, B12, B21, B22 = cut_matrix(B)
6
7     pool = multiprocessing.pool.ThreadPool(processes=7)
8
9     P1 = pool.apply_async(strassen_multiprocessing, (A11 + A22, B11 +
10     B22, border)).get()
11     P2 = pool.apply_async(strassen_multiprocessing, (A21 + A22, B11,
12     border)).get()
13     P3 = pool.apply_async(strassen_multiprocessing, (A11, B12-B22,
14     border)).get()
15     P4 = pool.apply_async(strassen_multiprocessing, (A22, B21 - B11,
16     border)).get()
17     P5 = pool.apply_async(strassen_multiprocessing, (A11 + A12, B22,
18     border)).get()
19     P6 = pool.apply_async(strassen_multiprocessing, (A21 - A11, B11 +
20     B12, border)).get()
21     P7 = pool.apply_async(strassen_multiprocessing, (A12 - A22, B21 +
22     B22, border)).get()
23
24     C11 = P1 + P4 - P5 + P7
25     C12 = P3 + P5
26     C21 = P2 + P4
27     C22 = P1 + P3 - P2 + P6
28
29     return np.vstack((np.hstack((C11, C12)), np.hstack((C21, C22))))
```

Листинг 3 — Построение графика

```
1 dims = [2 ** i for i in range(2, 9)]
2 times_common = []
3 times_st = []
4 times_st_multi = []
5
6 for dim in tqdm(dims):
7     A = generate_matrix(n=dim)
8     B = generate_matrix(n=dim)
9
10    start = time()
11    _ = strassen(A, B)
12    times_st.append(time() - start)
13
14    start = time()
15    _ = strassen_multiprocessing(A, B)
16    times_st_multi.append(time() - start)
17
18    start = time()
19    _ = mul_matrix(A, B)
20    times_common.append(time() - start)
21
22
23 plt.figure(figsize=(8, 4))
24 plt.xlabel("n")
25 plt.ylabel("t")
26 plt.plot(dims, times_common, label="Common", color="blue")
27 plt.plot(dims, times_st, label="Strassen", color="green")
28 plt.plot(dims, times_st_multi, label="Strassen_multiprocessing", color="red")
29 plt.legend()
30 plt.show()
```

3 Результаты

На рисунке 1 приведено сравнение погрешностей обычного метода Штрассена и метода Штрассена с многопоточностью.



```
strassen error: 4.503985401660681e-11
strassen_multi_error: 4.503985401660681e-11
```

Рис. 1 — Сравнение погрешностей

Графики зависимости времени выполнения от размерности матриц приведены на рисунке 2.

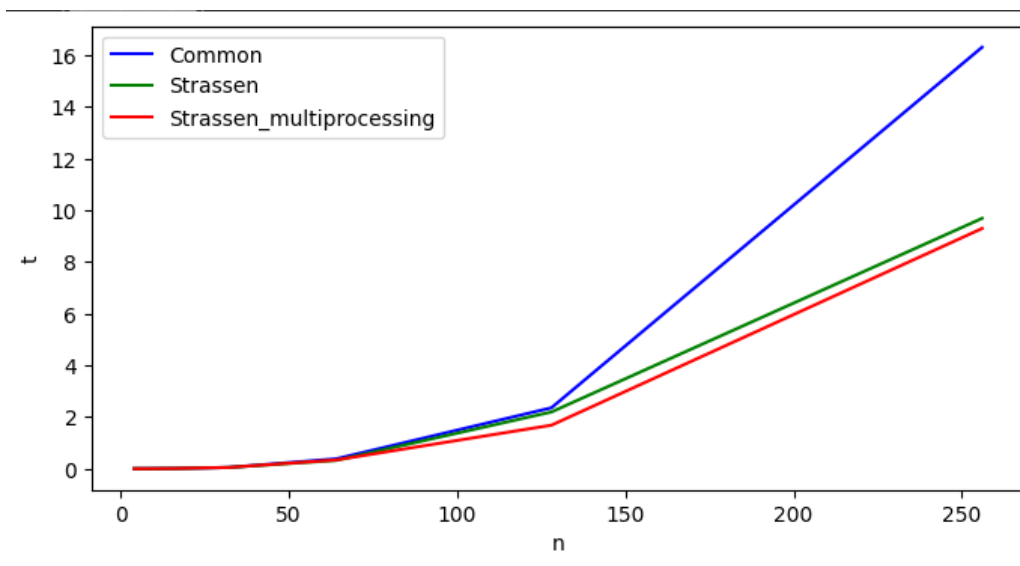


Рис. 2 — Графики зависимости t от N

4 Выводы

В рамках данной лабораторной работы был реализован метод Штрассена, а также его многопоточная модификация. Также было проведено сравнение времени выполнения этих методов и обычного метода перемножения матриц в зависимости от размерности матрицы. Было установлено, что с ростом размерности метод Штрассена и его модификация демонстрируют куда лучший результат, нежели стандартный метод.