



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»

Студентка группы ИУ9-72Б Самохвалова П. С.

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 Практическая реализация

Исходный код программы представлен в листинге 1.

Листинг 1: Метод Штрассена

```
1 import time
2 from multiprocessing.pool import ThreadPool
3 import numpy as np
4 from num_methods import *
5
6
7 def strass(a, b, n_min):
8     n = len(a)
9     if n <= n_min:
10         return np.array(mult_matr_matr(a, b))
11     a11, a12, a21, a22 = split_into_parts(a)
12     b11, b12, b21, b22 = split_into_parts(b)
13     p1 = strass(a11 + a22, b11 + b22, n_min)
14     p2 = strass(a21 + a22, b11, n_min)
15     p3 = strass(a11, b12 - b22, n_min)
16     p4 = strass(a22, b21 - b11, n_min)
17     p5 = strass(a11 + a12, b22, n_min)
18     p6 = strass(a11 - a21, b11 + b12, n_min)
19     p7 = strass(a12 - a22, b21 + b22, n_min)
20     c11 = p1 + p4 - p5 + p7
21     c12 = p3 + p5
22     c21 = p2 + p4
23     c22 = p1 + p3 - p2 + p6
24     res = np.vstack((np.hstack((c11, c12)), np.hstack((c21, c22))))
25     return res
26
```

```

27
28 def strass_multiprocessing(a, b, n_min):
29     n = len(a)
30     if n <= n_min:
31         return np.array(mult_matr_matr(a, b))
32     a11, a12, a21, a22 = split_into_parts(a)
33     b11, b12, b21, b22 = split_into_parts(b)
34     pool = ThreadPool(processes=7)
35     p1 = pool.apply_async(strass, (a11 + a22, b11 + b22, n_min)).get()
36     p2 = pool.apply_async(strass, (a21 + a22, b11, n_min)).get()
37     p3 = pool.apply_async(strass, (a11, b12 - b22, n_min)).get()
38     p4 = pool.apply_async(strass, (a22, b21 - b11, n_min)).get()
39     p5 = pool.apply_async(strass, (a11 + a12, b22, n_min)).get()
40     p6 = pool.apply_async(strass, (a11 - a21, b11 + b12, n_min)).get()
41     p7 = pool.apply_async(strass, (a12 - a22, b21 + b22, n_min)).get()
42     c11 = p1 + p4 - p5 + p7
43     c12 = p3 + p5
44     c21 = p2 + p4
45     c22 = p1 + p3 - p2 + p6
46     res = np.vstack((np.hstack((c11, c12)), np.hstack((c21, c22))))
47     return res
48
49
50 def split_into_parts(matr):
51     n, m = matr.shape
52     n1, m1 = n // 2, m // 2
53     return matr[:n1, :m1], matr[:n1, m1:], matr[n1:, :m1], matr[n1:, m1:]
54
55
56 n = 128
57
58 a = np.array(generate_matrix(n, -10, 10))
59 b = np.array(generate_matrix(n, -10, 10))
60
61 n_min = 64
62
63 c1 = strass(a, b, n_min)
64 print("Strassen method")
65 print_matr(c1)
66 print()
67 c2 = strass_multiprocessing(a, b, n_min)
68 print("Strassen method with multiprocessing")
69 print_matr(c2)
70 print()
71 c3 = mult_matr_matr(a, b)

```

```

72 print("Standart algorithm")
73 print_matr(c3)
74 print()
75
76 x = [2 ** i for i in range(5, 10)]
77 y1 = []
78 y2 = []
79 y3 = []
80
81 for n in x:
82     a = np.array(generate_matrix(n, -10, 10))
83     b = np.array(generate_matrix(n, -10, 10))
84     t1 = time.time()
85     c1 = strass(a, b, n_min)
86     t2 = time.time()
87     y1.append(t2 - t1)
88     t3 = time.time()
89     c2 = strass_multiprocessing(a, b, n_min)
90     t4 = time.time()
91     y2.append(t4 - t3)
92     t5 = time.time()
93     c3 = mult_matr_matr(a, b)
94     t6 = time.time()
95     y3.append(t6 - t5)
96
97 plt.xlabel('n')
98 plt.ylabel('time')
99 plt.grid()
100 plt.plot(x, y1, color = "blue", label = "Strassen method")
101 plt.plot(x, y2, color = "red", label = "Strassen method multiprocessing"
102         )
102 plt.plot(x, y3, color = "green", label = "Standart algorithm")
103
104 plt.legend()
105 plt.show()

```

3 Результаты

Результаты работы программы представлены на рисунках 1 – 5.

```
Strassen method
-128.74400857560553 36.75353853707304 -167.32225658610395 196.84149706314656 -327.9876176642784 -333.78819694783083 -171.8239549958368 -36.99980582521562 -386.5
10.989431194213068 218.6342970073875 918.846534845321 399.6582881223338 19.081641006835355 349.79580214400835 -105.00833988958942 636.6696719027328 -314.80238080
236.01562135599863 217.56483120095555 -425.25904717973805 325.64401423067443 -609.6205953529569 157.04718968809982 -113.54324673559228 -675.0372047452688 -1033.3
-927.6022636125437 -184.69907509592997 202.48412582613105 211.6984188070728 -609.065324210861 -606.1645372653312 332.97197693253435 -471.95540032484405 -349.55495
134.73237109395052 -423.098463787800115 -487.0537188382706 -109.7970396138187 33.08287982513298 432.9661460009249 77.72410045060129 -310.6939296376972 435.80534898
-78.89870273171618 252.93667685070977 43.8761626699569 350.09728923748025 159.08203981856232 404.0800258082155 218.95432081834963 -95.40659516955674 149.09752389
454.24698082949124 -231.58165826819203 220.72448571360076 202.10986077416567 227.72541668474582 -171.57896572346175 -62.75776223412413 -64.74835033754528 -376.88
-211.22374908469478 415.606510599021 22.061049830639718 -717.282943018354 237.32285359867717 598.274968112735 402.77543409902546 960.7075216717863 443.86824400866
-895.786596611393 -115.35313006199215 -562.3201768596425 -626.11589338273 -566.2309195117961 801.8613513754603 -127.91144319204525 -49.8751751202343 -249.5283236
243.2616046961108 55.2258237913606 464.7149661340476 409.30704892412564 -639.379943885159 -293.85192341168397 -700.2198183608207 403.9960134060907 13.52478492404
70.79658795481765 -637.8970508925872 278.28300172178933 -35.52728559282153 -24.182257251611475 -155.5090952897193 392.22919438498195 -209.84412235311981 -342.896
364.5933225147832 509.3275020117831 608.4018719297655 453.74596172418666 29.521423821189558 -426.50247945529424 -134.31669286375734 -175.7888115104863 428.905199
-285.71963801485526 -103.44493876852005 280.58817088074835 -114.1255877811644 -335.1850908807065 184.78150581380712 144.14738523992878 355.85152460514576 178.038
69.47308701526526 241.67244706035956 -289.2634683508693 -266.80646328859757 -88.67852426023109 535.3088884971967 366.52692991657653 -195.07846447630993 -200.2070
-40.33801569958325 349.62470654087156 342.6337534363612 55.48873653178498 -19.802917683521912 -178.23206915446886 732.2475290202269 433.2566539834206 394.6083236
-424.9762619359917 356.02245664075053 75.71872157109298 -495.8797660667941 -371.07511548358355 -217.96645951854467 -224.5368803009502 168.17784657982338 -10.0675
30.2890444646265 539.1173190449498 -937.4634273459693 -43.113167323111384 -598.5311063222312 -272.95863262550677 -550.0936418275552 -634.4130963734299 -81.083434
```

Рис. 1 — Результат метода Штрассена

```
Strassen method with multiprocessing
-128.74400857560553 36.75353853707304 -167.32225658610395 196.84149706314656 -327.9876176642784 -333.78819694783083 -171.8239549958368 -36.99980582521562 -386.5
10.989431194213068 218.6342970073875 918.846534845321 399.6582881223338 19.081641006835355 349.79580214400835 -105.00833988958942 636.6696719027328 -314.80238080
236.01562135599863 217.56483120095555 -425.25904717973805 325.64401423067443 -609.6205953529569 157.04718968809982 -113.54324673559228 -675.0372047452688 -1033.3
-927.6022636125437 -184.69907509592997 202.48412582613105 211.6984188070728 -609.065324210861 -606.1645372653312 332.97197693253435 -471.95540032484405 -349.55495
134.73237109395052 -423.098463787800115 -487.0537188382706 -109.7970396138187 33.08287982513298 432.9661460009249 77.72410045060129 -310.6939296376972 435.80534898
-78.89870273171618 252.93667685070977 43.8761626699569 350.09728923748025 159.08203981856232 404.0800258082155 218.95432081834963 -95.40659516955674 149.09752389
454.24698082949124 -231.58165826819203 220.72448571360076 202.10986077416567 227.72541668474582 -171.57896572346175 -62.75776223412413 -64.74835033754528 -376.88
-211.22374908469478 415.606510599021 22.061049830639718 -717.282943018354 237.32285359867717 598.274968112735 402.77543409902546 960.7075216717863 443.86824400866
-895.786596611393 -115.35313006199215 -562.3201768596425 -626.11589338273 -566.2309195117961 801.8613513754603 -127.91144319204525 -49.8751751202343 -249.5283236
243.2616046961108 55.2258237913606 464.7149661340476 409.30704892412564 -639.379943885159 -293.85192341168397 -700.2198183608207 403.9960134060907 13.52478492404
70.79658795481765 -637.8970508925872 278.28300172178933 -35.52728559282153 -24.182257251611475 -155.5090952897193 392.22919438498195 -209.84412235311981 -342.896
364.5933225147832 509.3275020117831 608.4018719297655 453.74596172418666 29.521423821189558 -426.50247945529424 -134.31669286375734 -175.7888115104863 428.905199
-285.71963801485526 -103.44493876852005 280.58817088074835 -114.1255877811644 -335.1850908807065 184.78150581380712 144.14738523992878 355.85152460514576 178.038
69.47308701526526 241.67244706035956 -289.2634683508693 -266.80646328859757 -88.67852426023109 535.3088884971967 366.52692991657653 -195.07846447630993 -200.2070
-40.33801569958325 349.62470654087156 342.6337534363612 55.48873653178498 -19.802917683521912 -178.23206915446886 732.2475290202269 433.2566539834206 394.6083236
-424.9762619359917 356.02245664075053 75.71872157109298 -495.8797660667941 -371.07511548358355 -217.96645951854467 -224.5368803009502 168.17784657982338 -10.0675
30.2890444646265 539.1173190449498 -937.4634273459693 -43.113167323111384 -598.5311063222312 -272.95863262550677 -550.0936418275552 -634.4130963734299 -81.083434
```

Рис. 2 — Результат метода Штрассена с многопоточностью

```
Standard algorithm
-128.7440085756058 36.753538537073 -167.32225658610426 196.84149706314656 -327.9876176642784 -333.7881969470301 -171.82395499585408 -36.99980582521597 -386.54664
10.989431194212754 218.63429700738737 918.8465348453209 399.6582881223337 19.081641006834992 349.79580214400880 -105.00833988958935 636.6696719027328 -314.8023808
236.0156213559986 217.56483120095584 -425.2590471797381 325.6440142306744 -609.6205953529567 157.0471896881 -113.54324673559206 -675.0372047452684 -1033.34323019
-927.6022636125437 -184.6990750959296 202.4841258261309 211.6984188070727 -609.0653242108609 -606.1645372653121 332.97197693253435 -471.95540032484416 -349.55495
134.73237109395066 -423.09846378780011 -487.05371883827036 -109.797039613819 33.082879825133055 432.9661460009252 77.72410045060129 -310.6939296376972 435.8053489
-78.8987027317163 252.93667685070986 43.8761626699568 350.09728923748025 159.08203981856258 404.0800258082153 218.95432081834983 -95.406595169557 149.09752389165
454.2469808294911 -231.58165826819234 220.72448571360087 202.10986077416499 227.72541668474574 -171.57896572346186 -62.75776223412427 -64.7483503375453 -376.8857
-211.2237490849478 415.6065105990207 22.06104983063986 -717.2829430183542 237.32285359867757 598.274968112735 402.775434099026 960.7075216717861 443.868244008663
-895.7865966113931 -115.35313006199195 -562.3201768596426 -626.1158933827303 -566.230919511796 801.8613513754602 -127.91144319204506 -49.87517512023396 -249.5283
243.26160469611074 55.22582379136035 464.7149661340472 409.3070489241256 -639.379943885159 -293.8519234116833 -700.2198183608206 403.9960134060911 13.52478492404
70.79658795481754 -637.8970508925873 278.28300172178945 -35.52728559282154 -24.18225725161102 -155.5090952897196 392.22919438498207 -209.8441223531196 -342.89631
364.5933225147829 509.3275020117833 608.4018719297659 453.7459617241866 29.521423821189714 -426.502479455295 -134.3166928637575 -175.78881151048603 428.905199949
-285.7196380148552 -103.44493876852013 280.5881708807481 -114.12558778116474 -335.18509088070643 184.781505813807 144.14738523992892 355.8515246051454 178.03867
69.47308701526514 241.67244706035981 -289.2634683508693 -266.80646328859785 -88.67852426023113 535.308888497197 366.5269299165763 -195.07846447631 -200.207080580
-40.33801569958364 349.6247065408708 342.63375343636125 55.48873653178489 -19.802917683521862 -178.23206915446866 732.2475290202266 433.2566539834202 394.6083236
-424.9762619359917 356.02245664075014 75.71872157109259 -495.87976606679376 -371.0751154835834 -217.96645951854458 -224.5368803009088 168.17784657982315 -10.067
30.28904446462593 539.1173190449499 -937.4634273459691 -43.11316732311142 -598.5311063222308 -272.9586326255065 -550.0936418275554 -634.4130963734303 -81.0834345
```

Рис. 3 — Результат стандартного алгоритма

4 Выводы

В результате выполнения лабораторной работы были реализованы метод Штрассена, метод Штрассена с многопоточностью, проведены сравнения времени работы со стандартным алгоритмом.

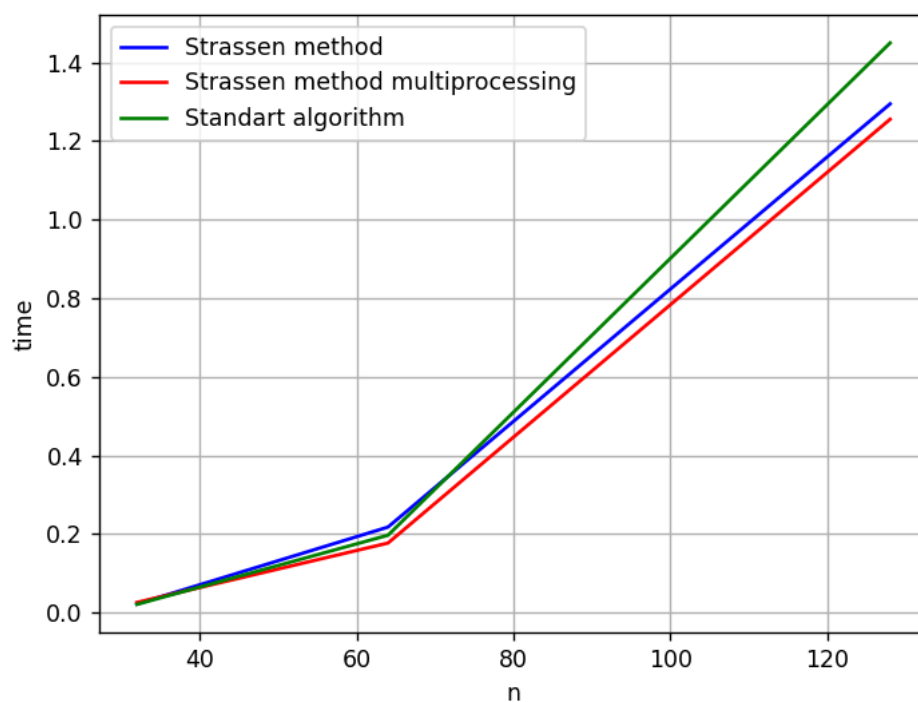


Рис. 4 — График зависимости времени работы от размера матрицы

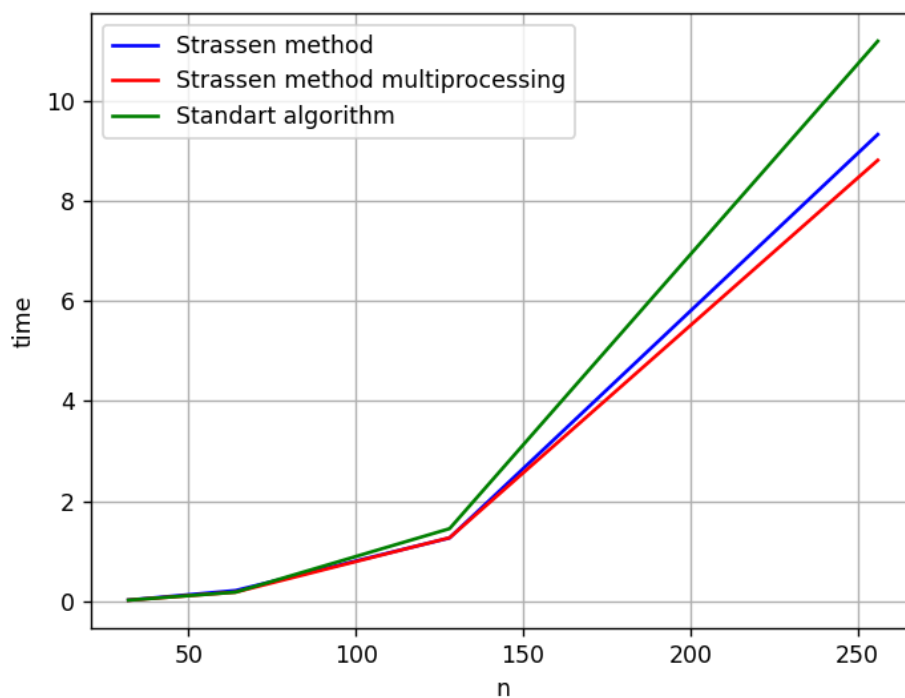


Рис. 5 — График зависимости времени работы от размера матрицы