|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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)** |

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ТЕОРЕТИЧЕСКАЯ ИНФОРМАТИКА И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛАБОРАТОРНАЯ РАБОТА №6

по дисциплине: «Численные методы линейной алгебры»

ИУ9-72б

Ф. Р. Базартинова

Выполнил студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Д.П. Посевин

Проверил **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

1 Постановка задачи

Необходимо реализовать умножение двух квадратных матриц размера, являющегося степенью двойки классическим методом, методом Штрассена и многопоточным методом Штрассена.

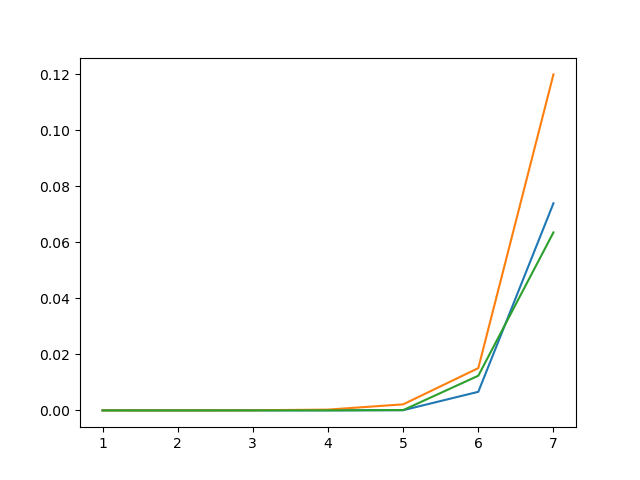
2 Программная реализация

Лабораторная работа была выполнена на языке Python. Исходный код основной программы приведён в листинге 1.

|  |
| --- |
| Листинг 1 - Реализация |
| def mulStrassen(a, b, nMin):  n = len(a)  if n <= nMin:  return np.matmul(a, b)  else:  a11, a21, a12, a22, b11, b12, b21, b22 = splitted(a, b, n)  with fur.ProcessPoolExecutor(max\_workers=2) as executor:  futP1 = executor.submit(mulStrassen, (a11 + a22), (b11 + b22), nMin)  futP2 = executor.submit(mulStrassen, (a21 + a22), b11, nMin)  futP3 = executor.submit(mulStrassen, a11, (b12 - b22), nMin)  futP4 = executor.submit(mulStrassen, a22, (b21 - b11), nMin)  futP5 = executor.submit(mulStrassen, (a11 + a12), b22, nMin)  futP6 = executor.submit(mulStrassen, (a21 - a11), (b11 + b12), nMin)  futP7 = executor.submit(mulStrassen, (a12 - a22), (b21 + b22), nMin)  P1 = futP1.result()  P2 = futP2.result()  P3 = futP3.result()  P4 = futP4.result()  P5 = futP5.result()  P6 = futP6.result()  P7 = futP7.result()  AB11 = P1 + P4 - P5 + P7  AB12 = P3 + P5  AB21 = P2 + P4  AB22 = P1 - P2 + P3 + P6  AB = np.zeros((n, n))  for i in range(n):  for j in range(n):  if i < n // 2 and j < n // 2:  AB[i, j] = AB11[i, j]  elif i < n // 2 and j >= n // 2:  AB[i, j] = AB12[i, j - n // 2]  elif i >= n // 2 and j < n // 2:  AB[i, j] = AB21[i - n // 2, j]  else:  AB[i, j] = AB22[i - n // 2, j - n // 2]  return AB  def strassen(a, b, nMin):  n = len(a)  if n <= nMin:  return np.matmul(a, b)  else:  a11, a21, a12, a22, b11, b12, b21, b22 = splitted(a, b, n)  P1 = strassen((a11 + a22), (b11 + b22), nMin)  P2 = strassen((a21 + a22), b11, nMin)  P3 = strassen(a11, (b12 - b22), nMin)  P4 = strassen(a22, (b21 - b11), nMin)  P5 = strassen((a11 + a12), b22, nMin)  P6 = strassen((a21 - a11), (b11 + b12), nMin)  P7 = strassen((a12 - a22), (b21 + b22), nMin)  res = np.zeros((n, n))  for i in range(n //2):  for j in range(n // 2):  res[i][j] = P1[i][j] + P4[i][j] - P5[i][j] + P7[i][j]  for i in range(n // 2):  for j in range(n // 2):  res[i + n // 2][j + n // 2] = P1[i][j] + P3[i][j] - P2[i][j] + P6[i][j]  for i in range(n // 2):  for j in range(n // 2):  res[i + n // 2][j] = P2[i][j] + P4[i][j]  for i in range(n // 2):  for j in range(n // 2):  res[i][j + n // 2] = P3[i][j] + P5[i][j]  return res |

3 Тестирование

Для тестирования необходимо сгенерировать две рандомные матрицы. Были выбраны матрицы размером 2^(i), i = 1,...,7 элементов. На рис. 1 представлен график зависимости скорости работы алгоритмов от размера матрицы.

Рис.1. График зависимостей

4 Вывод

В результате работы была создана и протестирована программа, которая выполняет умножение двух матриц классическим методом и методом Штрассена. Увеличение скорости работы метода Штрассена можно добиться с помощью использования нескольких потоков.