



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 № 3**  
**по курсу «Численные методы линейной алгебры»**  
**«Реализация метода Гаусса с перестановками»**

Студент группы ИУ9-71Б Яровикова А. С.

Преподаватель Посевин Д. П.

*Москва 2023*

# 1 Цель работы

Реализовать три варианта метода Гаусса с перестановками и научиться оценивать погрешность решения системы линейных уравнений для матриц произвольной размерности

## 2 Задание

1. Реализовать метод Гаусса с перестановками по столбцам, по строкам, по столбцам и строкам одновременно для действительных квадратных матриц произвольной размерности  $n$ .

2. Для проверки работоспособности алгоритмов необходимо использовать алгоритм тестирования задачи написанный в лабораторной работе №2 «Реализация метода Гаусса», который заключался в том, что мы заведомо определяем значения координат вектора  $x$ , данный вектор заведомо является решением уравнения  $A * x = b$ , вычисляем  $b$  путем прямого перемножения матрицы  $A$  на вектор  $x$  и далее производим поиск решения уравнения  $A * x = b$  тем или иным методом Гаусса, получая  $x_{chisl}$ , после чего производим сравнение полученного  $x_{chisl}$  с заданным  $x$ , а также решением  $x_{bibl}$ , полученным с использованием сторонней библиотеки выбранной студеном. При этом сравнение производится по Евклидовой норме разности вектора  $x - x_{chisl}$  и  $x - x_{bibl}$ .

3. На защите лабораторной работы студент должен показать умение оценивать погрешность вычислений в зависимости от выполнения условия диагонального преобладания матрицы, умение сравнивать погрешности вычислений полученных методом Гаусса с перестановками по столбцам, по строкам, по столбцам и строкам одновременно. Понимать связь теории с практикой.

4. Результат работы должен быть представлен в виде графиков зависимости абсолютной погрешности вычислений классическим методом Гаусса, методом Гаусса с перестановками по строкам, методом Гаусса с перестановками по столбцам, методом Гаусса с перестановками по столбцам и строкам, библиотечным методом от степени диагонального преобладания. Все графики должны быть построены на одной координатной плоскости. Напомним, что погрешность вычисления вектора  $x$  системы линейных алгебраических уравнений  $A \cdot x = b$  тем или

иным способом рассчитывается по Евклидовой норме разности точного решения и решения полученного соответствующим методом. Степень диагонального преобладания вычисляется, как максимальная разность по  $i$  между модулем диагонального элемента и суммы модулей вне диагональных элементов. Очевидно, что если значение степени диагонального преобладания положительна, то условие диагонального преобладания выполняется, в противном случае — не выполняется. Поэтому график должен быть построен как для отрицательных значений степени диагонального преобладания, так и для положительных.

### 3 Реализация

Исходный код программы представлен в листингах 1– 6.

## Листинг 1 — Вспомогательные функции

```
1 import numpy as np
2 from copy import deepcopy
3 import matplotlib.pyplot as plt
4
5 def euclidean_norm(vec):
6     res = 0
7     for el in vec:
8         res += el**2
9     return np.sqrt(res)
10
11 def mul_on_vector(matrix, vector):
12     res = []
13     for i in range(len(matrix)):
14         el = 0
15         for j in range(len(vector)):
16             el += matrix[i][j] * vector[j]
17         res.append(el)
18     return res
19
20 def generate_matrix(l, r, n):
21     a = np.random.uniform(l, r, (n, n))
22     return a
23
24 def increase_diag_elems(a, diag):
25     n = len(a)
26     for i in range(0, len(a)):
27         a[i][i] += diag * sum(abs(a[i][j]) if j != i else 0 for j in
range(n))
28     return a
29
30 def diag_dominance(a):
31     return max(abs(a[i][i]) - sum(abs(a[i][j]) if j != i else 0 for j in
range(len(a))) for i in range(len(a))
32
33 def test(method, A, x):
34     b = mul_on_vector(A, x)
35     x_calc = method(A, b)
36     return euclidean_norm(x - x_calc)
```

## Листинг 2 — Метод Гаусса

```

1
2 def gauss(matrix, vec):
3     n = len(matrix)
4     A = deepcopy(matrix)
5     b = deepcopy(vec)
6     x = np.zeros(shape=(n, ))
7     # towards
8     for i in range(n - 1):
9         if A[i][i] == 0:
10             for j in range(i + 1, n):
11                 if A[j][i] != 0:
12                     A[i], A[j] = A[j], A[i]
13                     break
14
15             for j in range(i + 1, n):
16                 c = - A[j][i] / A[i][i]
17                 A[j] += c * A[i]
18                 b[j] += c * b[i]
19
20     # backwards
21     for i in range(n - 1, -1, -1):
22         x[i] = b[i] / A[i][i]
23         for j in range(i - 1, -1, -1):
24             b[j] -= A[j][i] * x[i]
25
26     return np.array(x)

```

## Листинг 3 — Метод Гаусса с перестановкой по столбцам

```

1
2 def gauss_with_columns_permutation(matrix, vec):
3     n = len(matrix)
4     A = deepcopy(matrix)
5     b = deepcopy(vec)
6     x = np.zeros(shape=(n, ))
7     # towards
8     for i in range(n - 1):
9         # partial pivoting and swap
10         max_index = np.argmax(np.abs(A[i:, i])) + i
11         A[[i, max_index]] = A[[max_index, i]]
12         b[i], b[max_index] = b[max_index], b[i]
13         for j in range(i + 1, n):
14             f = A[j][i] / A[i][i]
15             A[j] -= f * A[i]
16             b[j] -= f * b[i]
17
18     # backwards
19     for i in range(n - 1, -1, -1):
20         x[i] = b[i] / A[i][i]
21         for j in range(i - 1, -1, -1):
22             b[j] -= A[j][i] * x[i]
23
24     return np.array(x)

```

#### Листинг 4 — Метод Гаусса с перестановкой по строкам

```
1
2 def gauss_with_rows_permutation(A, b):
3     n = len(A)
4     A = deepcopy(A)
5     b = deepcopy(b)
6     x_i = [i for i in range(n)]
7     x = np.zeros(shape=(n, ))
8
9     # towards
10    for i in range(n - 1):
11        # partial pivoting and swap
12        max_index = np.argmax(np.abs(A[i, i:])) + i
13        x_i[i], x_i[max_index] = x_i[max_index], x_i[i]
14        for j in range(n):
15            A[j][i], A[j][max_index] = A[j][max_index], A[j][i]
16        for j in range(i + 1, n):
17            f = A[j][i] / A[i][i]
18            A[j] -= f * A[i]
19            b[j] -= f * b[i]
20
21    # backwards
22    for i in range(n - 1, -1, -1):
23        x[i] = b[i] / A[i][i]
24        for j in range(i - 1, -1, -1):
25            b[j] -= A[j][i] * x[i]
26
27    x_copy = deepcopy(x)
28
29    for i, order in enumerate(x_i):
30        x[order] = x_copy[i]
31
32    return np.array(x)
```

## Листинг 5 — Метод Гаусса с перестановкой по строкам и столбцам

```

1
2 def gauss_with_rows_and_columns_permutation(A, b):
3     n = len(A)
4     A = deepcopy(A)
5     b = deepcopy(b)
6     x_i = [i for i in range(n)]
7     x = np.zeros(shape=(n, ))
8
9     # towards
10    for i in range(n - 1):
11        # partial pivoting and swap
12        max_index_col = np.argmax(np.abs(A[i:, i])) + i
13        max_index_row = np.argmax(np.abs(A[i, i:])) + i
14        if A[max_index_col][i] > A[i][max_index_row]:
15            A[[i, max_index_col]] = A[[max_index_col, i]]
16            b[i], b[max_index_col] = b[max_index_col], b[i]
17        else:
18            x_i[i], x_i[max_index_row] = x_i[max_index_row], x_i[i]
19            for j in range(n):
20                A[j][i], A[j][max_index_row] = A[j][max_index_row], A[j]
21                ][i]
22
23        for j in range(i + 1, n):
24            f = A[j][i] / A[i][i]
25            A[j] -= f * A[i]
26            b[j] -= f * b[i]
27
28    # backwards
29    for i in range(n - 1, -1, -1):
30        x[i] = b[i] / A[i][i]
31        for j in range(i - 1, -1, -1):
32            b[j] -= A[j][i] * x[i]
33
34    x_copy = deepcopy(x)
35
36    for i, order in enumerate(x_i):
37        x[order] = x_copy[i]
38
39    return np.array(x)

```

## Листинг 6 — Запуск программы и построение графиков зависимости

```
1 coefs = [i * 0.2 for i in range(1, 22, 3)]
2
3 for d in [10, 100, 300]:
4     diag = []
5
6     y_gauss = []
7     y_gauss_row = []
8     y_gauss_col = []
9     y_gauss_row_col = []
10    y_numpy = []
11    for c in coefs:
12        A = generate_matrix(-10, 10, d)
13        A = increase_diag_elems(A, c)
14        x = np.random.uniform(-10, 10, d)
15
16        diag.append(diag_dominance(A))
17        y_gauss.append(test(gauss, A, x))
18        y_gauss_row.append(test(gauss_with_rows_permutation, A, x))
19        y_gauss_col.append(test(gauss_with_columns_permutation, A, x))
20        y_gauss_row_col.append(test(
21            gauss_with_rows_and_columns_permutation, A, x))
22        y_numpy.append(test(np.linalg.solve, A, x))
23
24    plt.figure(figsize=(6, 5))
25    plt.title(f'matrix {d}x{d}')
26    plt.xlabel("Diagonal Dominance")
27    plt.ylabel("Absolute Error")
28    plt.plot(diag, y_gauss, label='Classic Gauss')
29    plt.plot(diag, y_gauss_row, label='Gauss Rows')
30    plt.plot(diag, y_gauss_col, label='Gauss Columns')
31    plt.plot(diag, y_gauss_row_col, label='Gauss Rows & Columns')
32    plt.plot(diag, y_numpy, label='Library')
33    plt.legend()
34    plt.grid(True)
35    plt.show()
```

## 4 Результаты

Результат запуска методов представлены на рисунках 1 - 3.



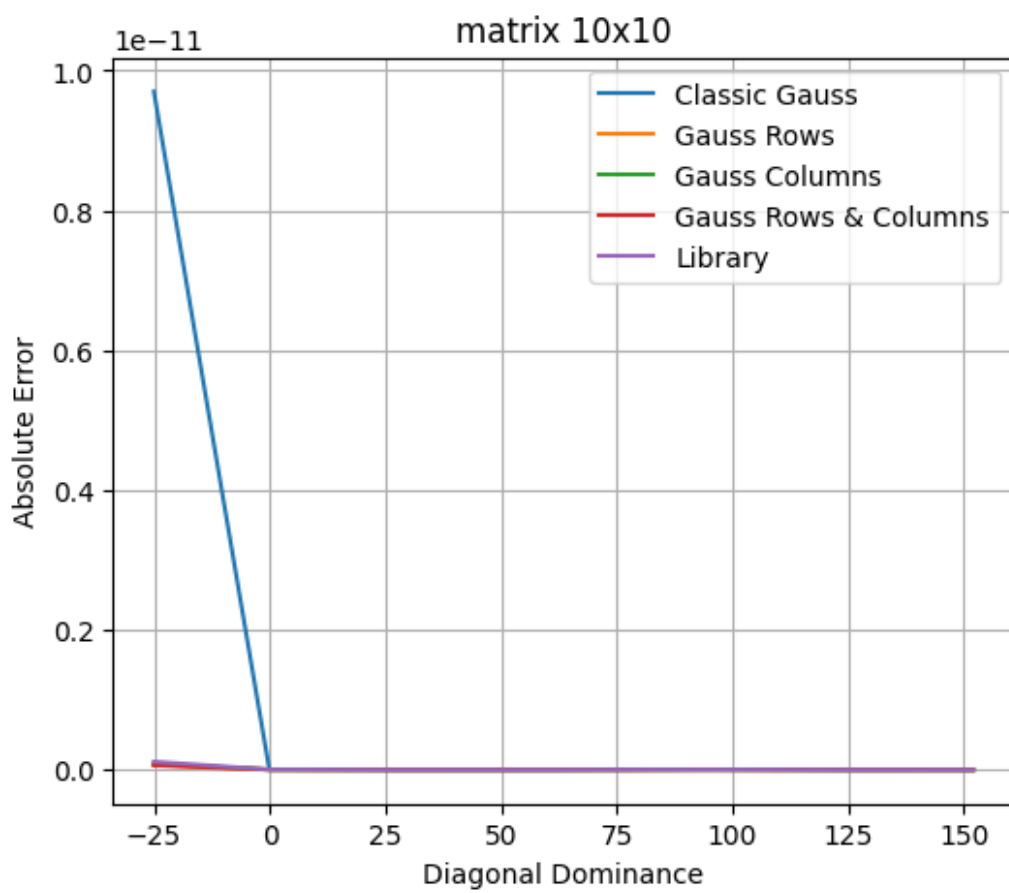


Рис. 1 — График зависимости для матрицы размерностью 10x10

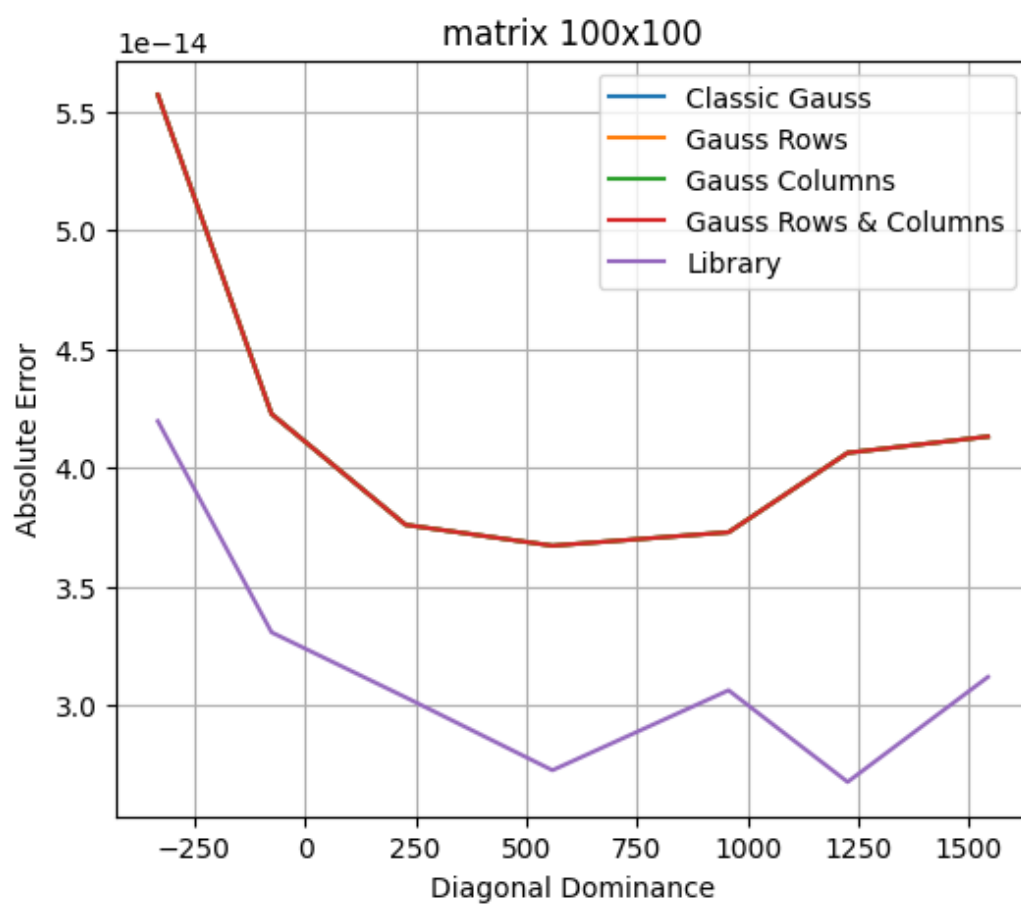


Рис. 2 — График зависимости для матрицы размерности  $100 \times 100$

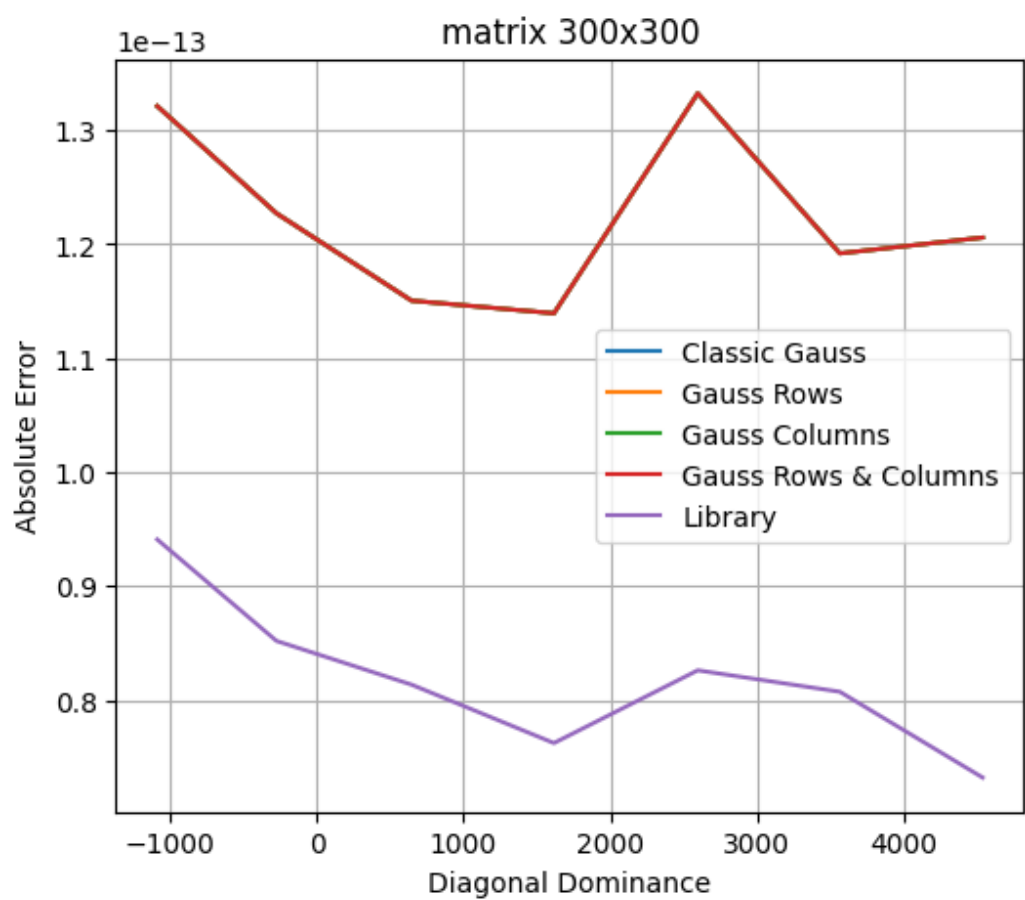


Рис. 3 — График зависимости для матрицы размерности 300x300

## 5 Выводы

В результате выполнения данной лабораторной работы были реализованы стандартный метод Гаусса, метод Гаусса с перестановками по столбцам, по строкам, по столбцам и строкам одновременно для действительных квадратных матриц произвольной размерности  $n$  на языке программирования Python.

Результат работы был представлен в виде графиков зависимости абсолютной погрешности вычислений классическим методом Гаусса, методом Гаусса с перестановками по строкам, методом Гаусса с перестановками по столбцам, методом Гаусса с перестановками по столбцам и строкам, библиотечным методом NumPy от степени диагонального преобладания.