



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 № 2
по курсу «Численные методы линейной алгебры»
«Реализация метода Гаусса и оценка погрешностей вычислений»

Студент группы ИУ9-71Б Баев Д.А

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

1. Реализовать метод Гаусса для действительных квадратных матриц произвольной размерности n . Возможность быстро менять размерность матрицы n в дальнейшем потребуется для проведения численных экспериментов по оценке скорости выполнения алгоритма и его точности.

2. Реализовать возможность ручного ввода элементов матрицы произвольной размерности.

3. Реализовать возможность генерации матриц со случайными элементами в заданном диапазоне $[-a, b]$, где a и b принадлежат \mathbb{R} . При этом необходимо уметь регулировать условие диагонального преобладания, другими словами реализовать возможность принудительного увеличения на заданный порядок среднее значение генерируемых диагональных элементов a_{ii} матрицы A системы уравнений $A \cdot x = b$.

4. Реализовать алгоритм тестирования задачи, который заключается в том, что мы заведомо определяем значения координат вектора x , данный вектор заведомо является решением уравнения $A \cdot x = b$, вычисляем b путем прямого перемножения матрицы A на вектор x и далее производим поиск решения уравнения $A \cdot x = b$ методом Гаусса, получая $x_{\text{числ}}$. После этого производим сравнение полученного $x_{\text{числ}}$ с заданным x , а также решением $x_{\text{библ}}$, полученным с использованием сторонней библиотеки выбранной студеном. При этом сравнение производится по евклидовой норме разности вектора $(x - x_{\text{числ}})$ и $(x - x_{\text{библ}})$.

2 Исходный код

Исходный код программы представлен в листингах 1–4.

Листинг 1 — Метод Гаусса

```
1 def gauss(A, b):
2     n = len(A)
3     A = deepcopy(A)
4     b = deepcopy(b)
5
6     for i in range(n - 1):
7         if A[i][i] == 0:
8             for j in range(i + 1, n):
9                 if A[j][i] != 0:
10                    A[i], A[j] = A[j], A[i]
11                    break
12
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     x = np.zeros(shape=(n, ))
19
20     for i in range(n - 1, -1, -1):
21         x[i] = b[i] / A[i][i]
22         for j in range(i - 1, -1, -1):
23             b[j] -= A[j][i] * x[i]
24
25     return np.array(x)
```

Листинг 2 — Чтение матрицы из файла

```
1 def read_matrix_from_file(filename):
2     matrix = []
3     with open(filename, 'r') as file:
4         lines = file.readlines()
5         for line in lines:
6             row = [int(x) for x in line.split()]
7             matrix.append(row)
8     return np.array(matrix)
```

Листинг 3 — Генерация матрицы

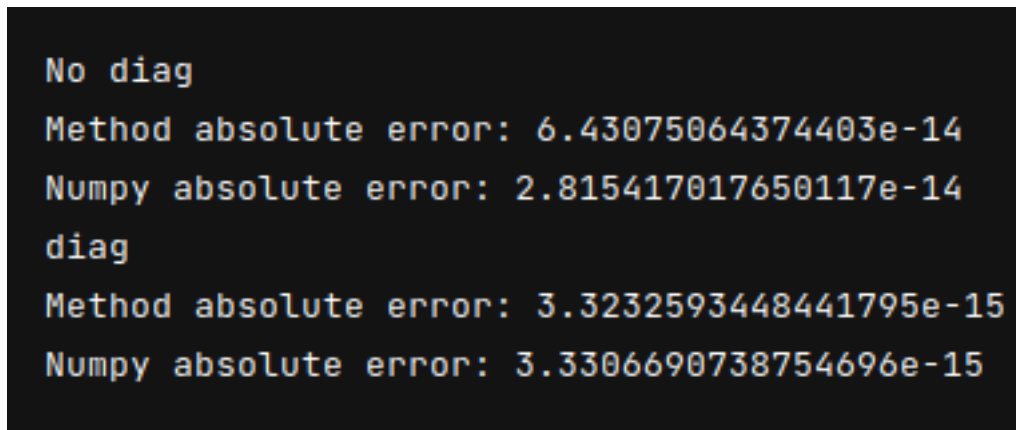
```
1 def generate_matrix(a, b, n, diag=None):
2     A = np.random.uniform(a, b, (n, n))
3     if diag is not None:
4         for i in range(len(A)):
5             A[i][i] += diag * sum(abs(A[i][j]) if j != i else 0 for j in
6                                     range(n))
6     return A
```

Листинг 4 — Тестирование

```
1 def test_method(method, a = -10, b = 10, n = 100, A=None, x=None):  
2     if A is None:  
3         A = generate_matrix(a, b, n)  
4     if x is None:  
5         x = np.random.uniform(a, b, size=n)  
6     b = mul_matrix_by_vector(A, x)  
7  
8     x_method = method(A, b)  
9     x_np = np.linalg.solve(A, b)
```

3 Результаты

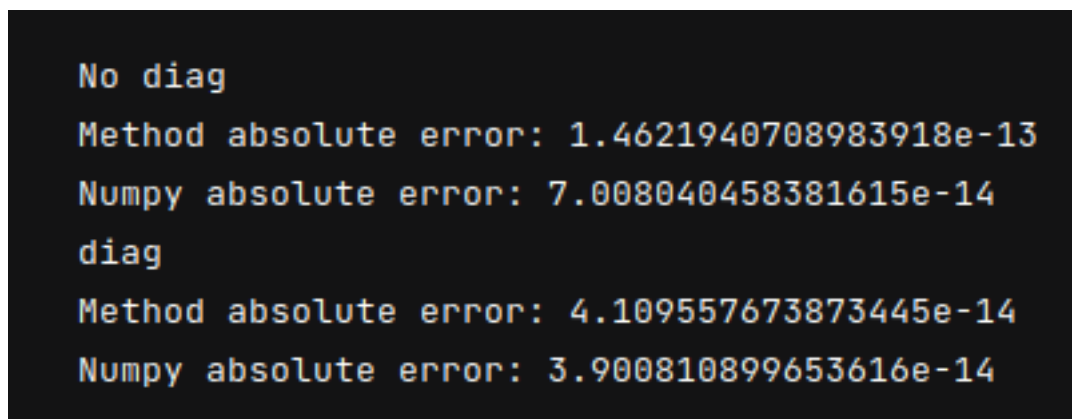
Результат проверки работы метода на матрице размерности 10x10 представлен на рисунке 1.



```
No diag  
Method absolute error: 6.43075064374403e-14  
Numpy absolute error: 2.815417017650117e-14  
diag  
Method absolute error: 3.3232593448441795e-15  
Numpy absolute error: 3.3306690738754696e-15
```

Рис. 1 — Результат проверки работы метода на матрице размерности 10x10

Результат проверки работы метода на матрице размерности 100x100 представлен на рисунке 2.



```
No diag  
Method absolute error: 1.4621940708983918e-13  
Numpy absolute error: 7.008040458381615e-14  
diag  
Method absolute error: 4.109557673873445e-14  
Numpy absolute error: 3.900810899653616e-14
```

Рис. 2 — Результат проверки работы метода на матрице размерности 100x100

Результат проверки работы метода на матрице размерности 300x300 представлен на рисунке 3.

```
No diag
Method absolute error: 1.2010458156486547e-13
Numpy absolute error: 1.1886258420223608e-14
diag
Method absolute error: 1.1065239156100681e-13
Numpy absolute error: 1.1047084252635322e-13
```

Рис. 3 — Результат проверки работы метода на матрице размерности 300x300

4 Выводы

В результате выполнения лабораторной работы был реализован классический метод Гаусса, а также вспомогательные методы с генерацией матрицы с возможным диагональным преобладанием и тестированием работы метода.

Была проведена оценка абсолютной погрешности получившегося значения и было проведено сравнение с результатом работы библиотечного метода из библиотеки Numpy. Классический метод Гаусса показал менее точный результат на матрице без диагонального преобладания и примерно одинаковый результат на матрице с диагональным преобладанием.