**Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**

**Лабораторная работа №1**

**Вариант 5**

Выполнил:

Воронов Григорий Алексеевич

Проверил:

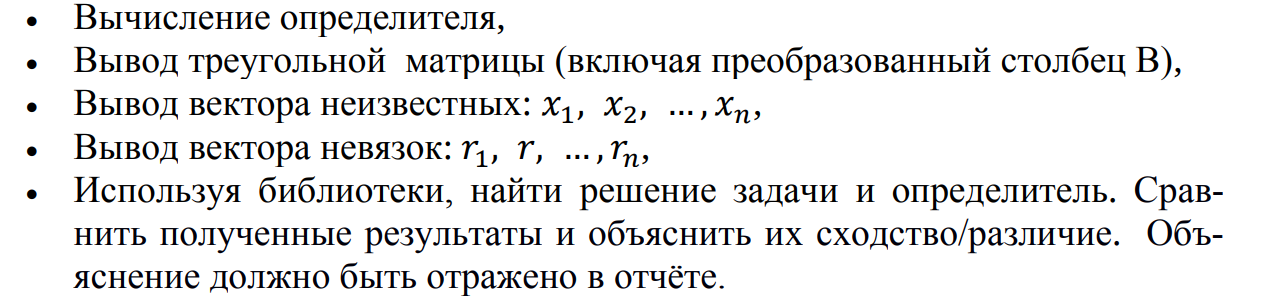
Рыбаков Степан Дмитриевич

г. Санкт-Петербург

2025

**Цель работы:**

Реализовать метод Гаусса с выбором главного элемента по столбцам для решения СЛАУ, позволяя вводить данные с клавиатуры или из файла. Программа должна вычислять определитель, приводить матрицу к треугольному виду, находить вектор неизвестных и вектор невязок, а также сравнивать результат с библиотечными методами.



**Описание метода:**

Чтобы избежать вычислительной погрешности при значениях ведущих элементов, близких к нулю по абсолютной величине, применяется метод Гаусса с выбором главного элемента.

Эта схема является одной из модификаций метода Гаусса. Идеей метода Гаусса с выбором главного элемента является такая перестановка уравнений, чтобы на k-ом шаге исключения ведущим элементом 𝑎ii оказывался наибольший по модулю элемент k-го столбца. Т.е. на очередном шаге k в уравнениях, начиная от k до последнего (i=k, k+1, …, n) в столбце k выбирают максимальный по модулю элемент и строки i и k меняются местами. Это выбор главного элемента «по столбцу».

Выбор главного элемента «по строке» - на очередном шаге k в строке k, начиная со столбца k (j = k, k+1, …, n) справа выбирается максимальный по модулю элемент. Столбцы j и k меняются местами.

**Листинг программы:**

**def gauss\_pivot(A, b):**

**n = len(A)**

**det = 1**

**for i in range(n):**

**max\_row = i**

**for k in range(i + 1, n):**

**if abs(A[k][i]) > abs(A[max\_row][i]):**

**max\_row = k**

**if A[max\_row][i] == 0:**

**raise ValueError("Матрица вырожденная")**

**A[i], A[max\_row] = A[max\_row], A[i]**

**b[i], b[max\_row] = b[max\_row], b[i]**

**det \*= A[i][i] \* (-1 if i != max\_row else 1)**

**for j in range(i + 1, n):**

**factor = A[j][i] / A[i][i]**

**for k in range(i, n):**

**A[j][k] -= factor \* A[i][k]**

**b[j] -= factor \* b[i]**

**x = [0] \* n**

**for i in range(n - 1, -1, -1):**

**sum\_ax = sum(A[i][j] \* x[j] for j in range(i + 1, n))**

**x[i] = (b[i] - sum\_ax) / A[i][i]**

**return x, det, A, b**

**def residual\_vector(A, b, x):**

**n = len(A)**

**residuals = [sum(A[i][j] \* x[j] for j in range(n)) - b[i] for i in range(n)]**

**return residuals**

**def read\_matrix\_from\_file(filename):**

**with open(filename, 'r') as f:**

**n = int(f.readline().strip())**

**A = [list(map(float, f.readline().split())) for \_ in range(n)]**

**b = list(map(float, f.readline().split()))**

**return A, b**

**def get\_matrix\_input():**

**n = int(input("Введите размерность матрицы (n <= 20): "))**

**A = []**

**print("Введите коэффициенты матрицы A построчно:")**

**for \_ in range(n):**

**A.append(list(map(float, input().split())))**

**print("Введите столбец свободных членов b:")**

**b = list(map(float, input().split()))**

**return A, b**

**choice = input("Выберите способ ввода данных (1 - файл, 2 - клавиатура): ")**

**if choice == '1':**

**filename = input("Введите имя файла: ")**

**A, b = read\_matrix\_from\_file(filename)**

**else:**

**A, b = get\_matrix\_input()**

**x, det, A\_triangular, b\_new = gauss\_pivot([row[:] for row in A], b[:])**

**residuals = residual\_vector(A, b, x)**

**print("=====================================")**

**print("Треугольная матрица:")**

**for row, bv in zip(A\_triangular, b\_new):**

**print(row + [bv])**

**print("\nВектор неизвестных:")**

**print(x)**

**print("\nОпределитель матрицы:")**

**print(det)**

**print("\nВектор невязок:")**

**print(residuals)**

**print("=====================================")**

**# Решение через библиотеки**

**import numpy as np**

**x\_np = np.linalg.solve(np.array(A, dtype=float), np.array(b, dtype=float))**

**det\_np = np.linalg.det(np.array(A, dtype=float))**

**print("\nРешение через numpy:")**

**print(x\_np)**

**print("\nОпределитель через numpy:")**

**print(det\_np)**

**Пример работы:**

Выберите способ ввода данных (1 - файл, 2 - клавиатура): 2

Введите размерность матрицы (n <= 20): 3

Введите коэффициенты матрицы A построчно:

2 2 4

6 9 3

3 8 2

Введите столбец свободных членов b:

7 5 1

=====================================

Треугольная матрица:

[6.0, 9.0, 3.0, 5.0]

[0.0, 3.5, 0.5, -1.5]

[0.0, 0.0, 3.142857142857143, 4.904761904761905]

Вектор неизвестных:

[1.03030303030303, -0.6515151515151515, 1.5606060606060608]

Определитель матрицы:

66.0

Вектор невязок:

[0.0, -8.881784197001252e-16, -4.440892098500626e-16]

=====================================

Решение через numpy:

[ 1.03030303 -0.65151515 1.56060606]

Определитель через numpy:

65.99999999999997

**Вывод:**

В ходе работы я реализовал метод Гаусса с выбором главного элемента по столбцу для решения СЛАУ, что позволило мне глубже понять его принципы и применение на практике.

Мною было реализовано два решения – при помощи библиотеки numpy и без. Результаты практически идентичны, что подтверждает правильность метода Гаусса с выбором главного элемента. А незначительная разница в числах обуславливается спецификой внутренней реализации вычислений в numpy и способом хранения чисел с плавающей точкой в ЭВМ.