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

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №1**

**по дисциплине «Вычислительная математика»**

Вариант: **17**

Преподаватель:   
Малышева Татьяна Алексеевна

Выполнил: Шмидт Александра

Группа: Р3215

Санкт-Петербург, 2025 г

# Цель работы

Изучить численные методы решения систем линейных алгебраических уравнений и реализовать один из них средствами программирования.

# Описание метода

Метод Гаусса является прямым методом, которые отличаются сравнительной простотой и универсальностью. Метод Гаусса основан на приведении матрицы к треугольному виду таким образом, чтобы ниже главной диагонали находились только нулевые элементы.

# Листинг программы

https://github.com/mrcdssclss/compmath\_lab1

import numpy as np  
  
def gauss(a):  
 n = len(a)  
 A = [row[:] for row in a]  
 x = n\*[0]  
 det = 1  
 swaps = 0  
  
 #приводим матрицу к треугольному виду (прямой ход)  
 for i in range(n):  
 if A[i][i] == 0:  
 f = False  
 for j in range(i + 1, n):  
 if A[j][i] != 0:  
 A[i], A[j] = A[j], A[i]  
 swaps += 1  
 f = True  
 break  
 if not f:  
 print("матрица вырождена (есть линейно зависимые строки)")  
 return None, None, None, None  
  
 det \*= A[i][i]  
  
 for j in range(i+1, n):  
 factor = A[j][i] / A[i][i]  
 for k in range(i, n+1):  
 A[j][k] -= A[i][k] \* factor  
  
 print("матрица в треугольном виде: ", A)  
  
 #обратный ход  
 for i in range(n-1, -1, -1):  
 x[i] = A[i][n]  
 for j in range(i+1, n):  
 x[i] -= A[i][j]\*x[j]  
 x[i] /= A[i][i]  
 round(x[i], 2)  
  
 #матрица коэффициентов  
 M = [row[:-1] for row in a]  
 #вектор правых частей  
 N = [row[-1] for row in a]  
 #вектор невязок (M-NX)  
 res = [sum(M[i][j] \* x[j] for j in range(n)) - N[i] for i in range(n)]  
 det \*= (-1)\*\*swaps  
  
 print("решение системы: ", x)  
 print("определитель матрицы: ", round(det,2))  
 print("вектор невязок: ", res)  
 print("количество перестановок: ", swaps)  
  
def numpy\_solution(a):  
 print("-"\*20)  
 A = np.array([row[:-1] for row in a])  
 B = np.array([row[-1] for row in a])  
  
 try:  
 det = round(np.linalg.det(A), 2) # определитель матрицы  
 X = np.round(np.linalg.solve(A, B), 2) # решение системы  
 except np.linalg.LinAlgError:  
 print("матрица вырождена")  
 return None, None  
 if det == 0:  
 print("матрица вырождена")  
 return None, None  
 print("решение numpy: ")  
 print("решение системы: ", X)  
 print("определитель матрицы: ", det)  
  
#главная функция,  
def main():  
 choice = input("Ввод матрицы с клавиатуры (1) или из файла (2): ")  
  
 if choice == "1":  
 a = keyboard\_read()  
 elif choice == "2":  
 a = file\_read()  
 else:  
 print("Неверный ввод")  
 return  
 if not a:  
 return  
  
 result = check(a)  
 print(result)  
 if "несовместна" in result:  
 return  
 elif "бесконечное число решений" in result:  
 return  
 else:  
 gauss(a)  
 numpy\_solution(a)  
  
  
#чтение матрицы из файла  
def file\_read():  
 filename = input("введите имя файла: ")  
 with open(filename) as f:  
 n = int(f.readline().strip())  
 a = [list(map(int, f.readline().split())) for \_ in range(n)]  
 return a  
  
def keyboard\_read():  
 n = int(input("введите размерность матрицы: "))  
 print("введите расширенную матрицу: ")  
 a = []  
 for i in range(n):  
 row = list(map(int, input().split()))  
 a.append(row)  
 return a  
  
def check(matrix):  
 n = len(matrix)  
 for row in matrix:  
 if all(x == 0 for x in row[:-1]) and row[-1] != 0:  
 return "Система несовместна"  
  
 for i in range(n):  
 col\_nonzero = any(matrix[i][j] != 0 for j in range(n))  
 if not col\_nonzero:  
 return "Система неопределена"  
  
 return ""  
  
main()

# Примеры работы программы

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# Вывод:

В результате выполнения данной лабораторной работой я познакомился с численными методами решения математических задач на примере систем алгебраических уравнений, реализовав на языке программирования Python метод простых итераций.