Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Отчет по лабораторной работе №1**

**по дисциплине «Вычислительная математика»**

**Вариант 11**

**Выполнили**:

Студенты группы P3265

Кручинина Дарья Сергеевна

Москвитина Полина

**Преподаватель:**

Машина Екатерина Алексеевна

**Цели работы**

Изучить численные методы решения систем линейных алгебраических уравнений и реализовать один из них, написав программу, которая бы решала систему алгебраических уравнений указанным математическим методом (методом Гаусса с выбором главного элемента по столбцам)

**Описание метода, расчётные формулы**

Метод Гаусса с выбором главного элемента по столбцам основан на приведении матрицы системы к треугольному виду так, чтобы ниже ее главной диагонали находились только нулевые элементы, путем последовательного исключения переменных.

Основные шаги метода Гаусса:

1. Прямой ход (Приведение к треугольному виду):

Проходим по каждой строке системы уравнений.

Для каждой строки выбираем ведущий элемент (наибольший по модулю) и используем его для обнуления всех элементов под ним в столбце.

После этого матрица принимает треугольный вид.

1. Обратный ход:

Начиная с последнего уравнения, находим значение последней неизвестной.

Подставляем это значение в предыдущие уравнения и находим значения предыдущих неизвестных.

Продолжаем этот процесс до тех пор, пока не найдем значения всех неизвестных.

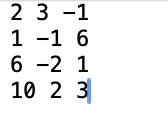
**Листинг программы**

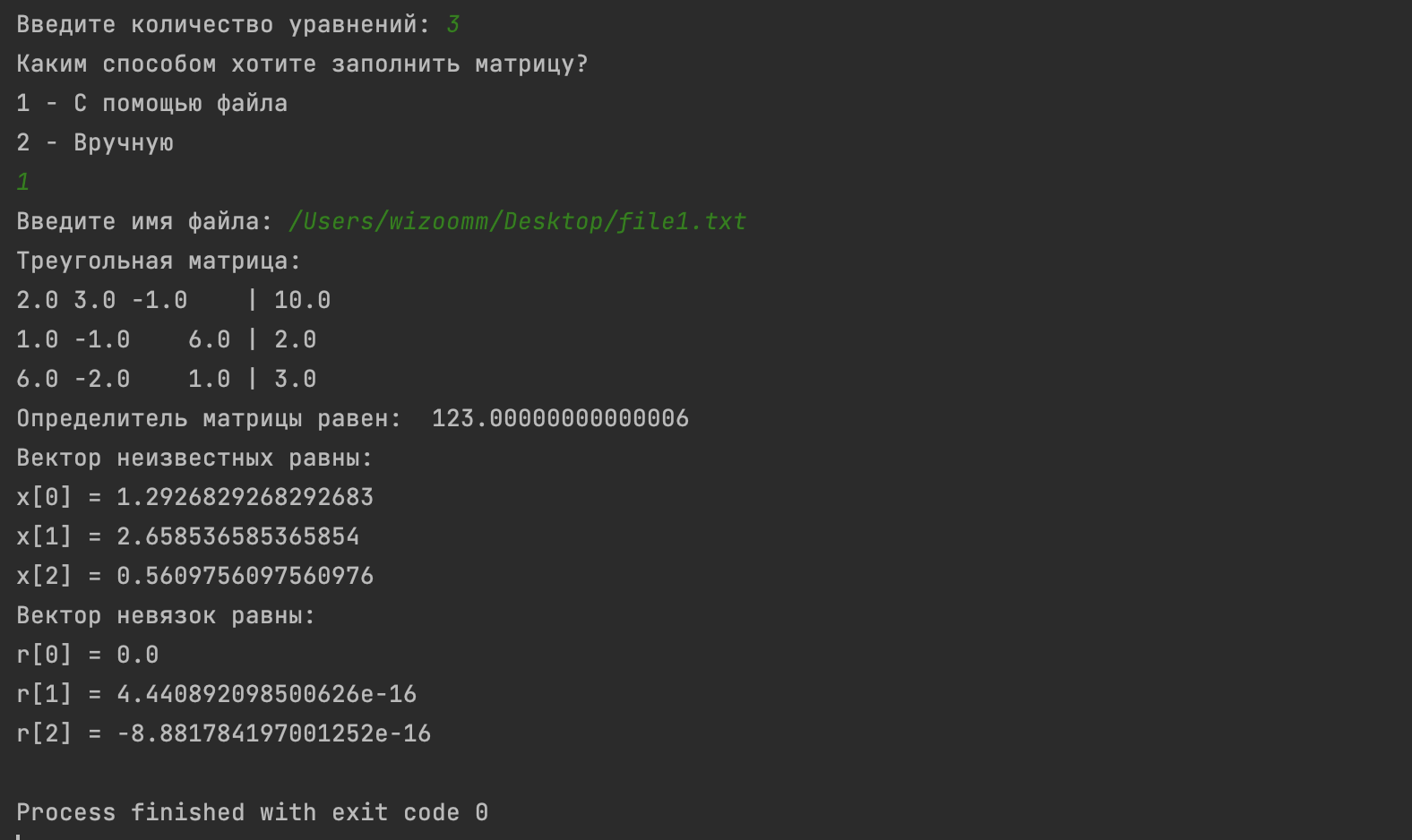
Код написан на языке Python

import numpy as np  
E = 0.0001  
def calculate\_determinant(matrix, n):  
 det = 1.0  
  
 if n == 2:  
 det = matrix[0][0] \* matrix[1][1] - matrix[0][1] \* matrix[1][0]  
 return det  
 if n == 3:  
 det = matrix[0][0] \* matrix[1][1] \* matrix[2][2] + matrix[0][1] \* matrix[1][2] \* matrix[2][0] + matrix[0][2] \* matrix[1][0] \* matrix[2][1] - matrix[0][2] \* matrix[1][1] \* matrix[2][0] - matrix[0][0] \* matrix[1][2] \* matrix[2][1] - matrix[0][1] \* matrix[1][0] \* matrix[2][2]  
 return det  
  
 for i in range(n):  
 pivot = i  
 for j in range(i + 1, n):  
 if abs(matrix[j][i]) > abs(matrix[pivot][i]):  
 pivot = j  
 if pivot != i:  
 det \*= -1.0  
 matrix[i], matrix[pivot] = matrix[pivot], matrix[i]  
 det \*= matrix[i][i]  
 if abs(det) < E:  
 return 0.0  
 for j in range(i + 1, n):  
 coefficient = matrix[j][i] / matrix[i][i]  
 for k in range(i, n):  
 matrix[j][k] -= coefficient \* matrix[i][k]  
 return det  
  
def gauss\_solve(matrix, y\_values, n):  
 answers = [0] \* n  
 k, index = 0, 0  
 for k in range(n):  
 max\_val = abs(matrix[k][k])  
 index = k  
 for i in range(k + 1, n):  
 if abs(matrix[i][k]) > max\_val:  
 max\_val = abs(matrix[i][k])  
 index = i  
  
 matrix[k], matrix[index] = matrix[index], matrix[k]  
 y\_values[k], y\_values[index] = y\_values[index], y\_values[k]  
  
 for i in range(k, n):  
 temp = matrix[i][k]  
 if abs(temp) < E:  
 continue  
 for j in range(k, n):  
 matrix[i][j] = matrix[i][j] / temp  
 y\_values[i] = y\_values[i] / temp  
 if i == k:  
 continue  
 for j in range(n):  
 matrix[i][j] = matrix[i][j] - matrix[k][j]  
 y\_values[i] = y\_values[i] - y\_values[k]  
  
 for k in range(n - 1, -1, -1):  
 answers[k] = y\_values[k]  
 for i in range(k):  
 y\_values[i] = y\_values[i] - matrix[i][k] \* answers[k]  
 return answers  
  
  
  
  
  
  
def print\_matrix(matrix\_a, vector\_b, n):  
 print("Треугольная матрица: ")  
 for i in range(n):  
 for j in range(n):  
 print(matrix\_a[i][j], end="\t")  
 print("|", vector\_b[i])  
  
def print\_results(vector\_x, residuals):  
 print("Вектор неизвестных равны: ")  
 for i in range(len(vector\_x)):  
 print(f"x[{i}] = {vector\_x[i]}")  
 print("Вектор невязок равны: ")  
 for i in range(len(residuals)):  
 print(f"r[{i}] = {residuals[i]}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 n = int(input("Введите количество уравнений: "))  
 if n > 20:  
 print("Количество уравнений превышает допустмиое :(")  
 exit()  
  
 matrix\_a = np.zeros((n, n))  
 vector\_y = np.zeros(n)  
  
 print("Каким способом хотите заполнить матрицу?")  
 print("1 - С помощью файла")  
 print("2 - Вручную")  
 choice = int(input())  
 if choice == 1:  
 filename = input("Введите имя файла: ")  
 try:  
 with open(filename, 'r') as file:  
 for i in range(n):  
 matrix\_a[i] = list(map(float, file.readline().split()))  
 vector\_y = list(map(float, file.readline().split()))  
 except IOError:  
 print("Ошибка чтения файла")  
 exit()  
 else:  
 for i in range(n):  
 print(f"Введите коэффициенты {i + 1}-го уравнения через пробел: ")  
 matrix\_a[i] = list(map(float, input().split()))  
 print("Введите правые части уравнений через пробел ")  
 vector\_y = list(map(float, input().split()))  
  
 print\_matrix(matrix\_a, vector\_y, n)  
 det = np.linalg.det(matrix\_a)  
 print("Определитель матрицы равен: ", det)  
 if det != 0.0:  
 vector\_x = np.linalg.solve(matrix\_a, vector\_y)  
 residuals = np.dot(matrix\_a, vector\_x) - vector\_y  
 print\_results(vector\_x, residuals)  
 else:  
 print("Определитель матрицы равен 0 => система либо имеет бесконечное множество решений, либо не имеет решений, т. е. несовместна")

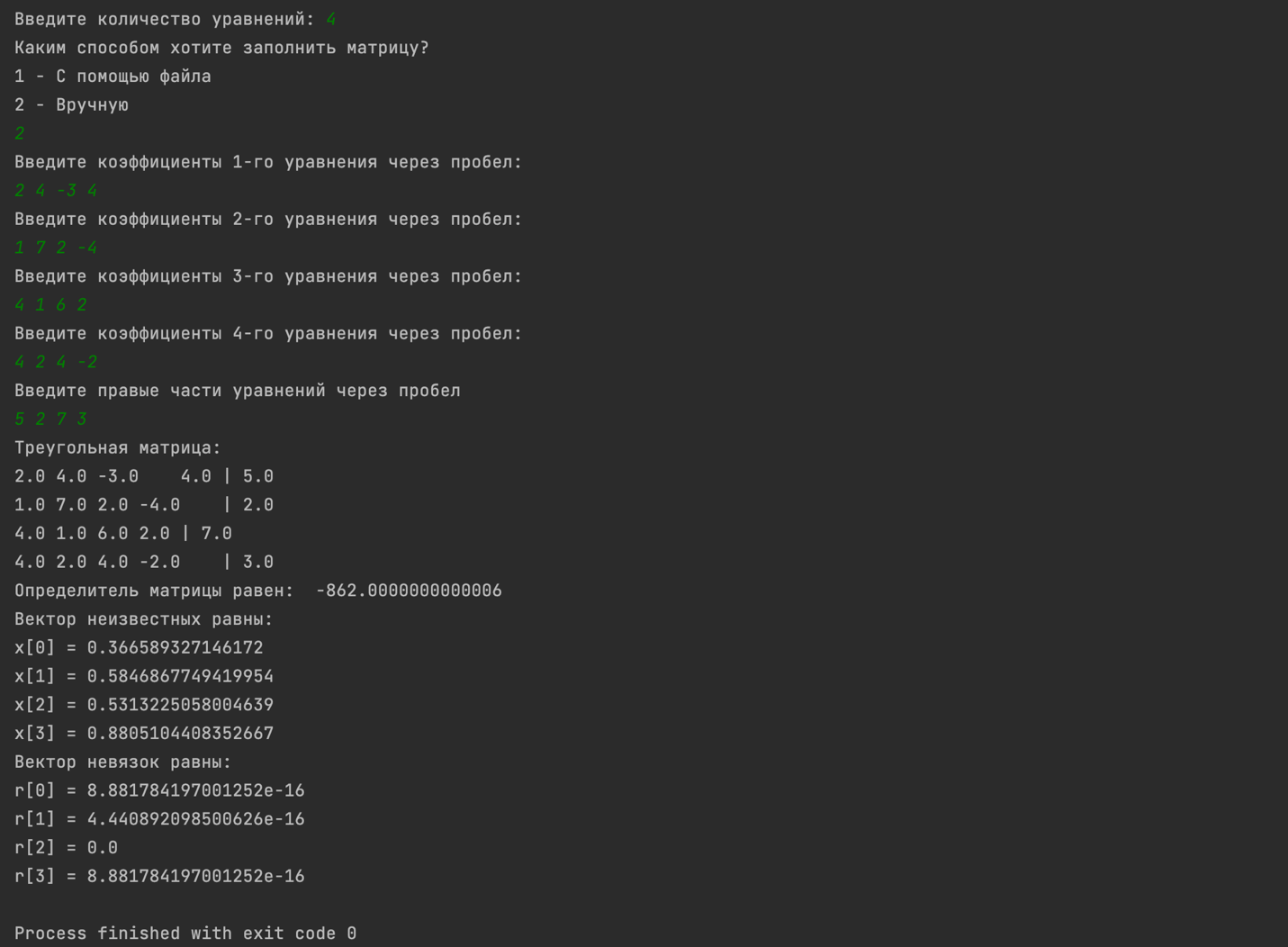
**Примеры и результаты работы программы**

Ввод с файла





Ввод с клавиатуры



**Вывод**

В ходе работы мы изучили метод Гауссас выбором главного элемента по столбцам, а также познакомились с различными способами решения СЛАУ. А также написали программу, в которой данные могут подаваться с клавиатуры или файла.