

Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Лабораторная работа №1
По вычислительной математике
Вариант 3

Выполнил:

Студент группы Р3206
Бижанов Расул Сунгатович

Преподаватель:

Рыбаков Степан Дмитриевич

ИТМО

г. Санкт-Петербург

2024 г.

Оглавление

Цель работы	2
Описание метода	2
Листинг программы	2
Примеры и результаты работы программы	3
Пример №1	3
Пример №2	4
Пример №3	4
Вывод	4

Цель работы

Реализовать один из методов вычисления СЛАУ, согласно варианту и предоставить программу, в которой:

1. Численный метод должен быть реализован в виде отдельной подпрограммы/метода/класса, в который исходные/выходные данные передаются в качестве параметров.
2. Размерность матрицы $n \leq 20$ (задается из файла или с клавиатуры - по выбору конечного пользователя).
3. Должна быть реализована возможность ввода коэффициентов матрицы, как с клавиатуры, так и из файла (по выбору конечного пользователя).

Описание метода

Прямой ход – приведение матрицы к ступенчатому виду.

Обратный ход – вычисление «снизу вверх» искомых неизвестных, подставляя каждый раз найденные решения в последующие, более верхние, строки.

Листинг программы

```
def gaussian_elimination(matrix, b):
    n = len(matrix)
    det = 1 # Определитель

    # Прямой ход
    for i in range(n):
        # Поиск максимального элемента в столбце
        max_index = i
        for j in range(i + 1, n):
            if abs(matrix[j][i]) > abs(matrix[max_index][i]):
                max_index = j

        # Обмен строк для улучшения точности
        matrix[i], matrix[max_index] = matrix[max_index], matrix[i]
        b[i], b[max_index] = b[max_index], b[i]

    if matrix[i][i] == 0:
```

```

        print("Был получен нулевой элемент на главной диагонали. Метод Гаусса не применим.")
        return None

# Приведение матрицы к треугольному виду
for j in range(i + 1, n):
    coef = matrix[j][i] / matrix[i][i]
    for k in range(i, n):
        matrix[j][k] -= coef * matrix[i][k]
    b[j] -= coef * b[i]

det *= matrix[i][i] # Обновление определителя

if det == 0:
    print("Определитель матрицы равен 0. Система имеет бесконечное количество решений или не имеет решений.")
    return None

# Обратный ход
x = [0] * n
for i in range(n - 1, -1, -1):
    x[i] = b[i]
    for j in range(i + 1, n):
        x[i] -= matrix[i][j] * x[j]
    x[i] /= matrix[i][i]

return x
https://github.com/BIZHAN28/CompMath1

```

Примеры и результаты работы программы

Пример №1

Выберите способ ввода данных:

1. С клавиатуры
2. Из файла

Введите номер: 1

Введите размерность матрицы ($n \leq 20$): 3

Введите коэффициенты матрицы:

5 1 -4

23 -3 5

5 -0.625 9

Введите столбец свободных членов:

2 2 2

Треугольная матрица:

[23.0, -3.0, 5.0]

[0.0, 1.6521739130434783, -5.086956521739131]

[0.0, 0.0, 7.996710526315789]

Решение:

x1 = 0.24598930481283424
x2 = 1.540106951871658
x3 = 0.1925133689839572

Вектор невязок:

[-4.440892098500626e-16, 2.220446049250313e-16, 0.0]

Пример №2

Выберите способ ввода данных:

1. С клавиатуры

2. Из файла

Введите номер: 2

Введите имя файла: Inputs

Был получен нулевой элемент на главной диагонали. Метод Гаусса не применим.

main.py	Inputs	:
1	4	
2	4 4 4 4	
3	4 4 4 4	
4	4 4 4 4	
5	4 4 4 4	
6	1 1 11 1	

Пример №3

Выберите способ ввода данных:

1. С клавиатуры

2. Из файла

Введите номер: 2

Введите имя файла: Inputs

Ошибка в файле: Количество элементов в строке должно быть равным размерности матрицы

main.py	Inputs	:
1	4	
2	4 4 4 4	
3	4 4	
4	4 4 4 4	
5	4 4 4 4	
6	1 1 11 1	

Вывод

Таким образом, в результате выполнения лабораторной работы я изучил принцип работы метода Гаусса на практике.