

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Дисциплина «Вычислительная математика»

Отчет

По лабораторной работе №1

Выполнил:

Терновский И.Е

Преподаватель:

Перл О.В

Санкт-Петербург, 2023 г.

Цель работы

Разработать программу для подсчета корней СЛАУ.

Для прямых методов должно быть реализовано:

- Вычисление определителя
- Вывод треугольной матрицы (включая преобразованный столбец В)
- Вывод вектора неизвестных: x_1, x_2, \dots, x_n
- Вывод вектора невязок: r_1, r_2, \dots, r_n

Описание метода

Метод Основан на приведении матрицы системы к треугольному виду так, чтобы ниже ее главной диагонали находились только нулевые элементы

Прямым ходом метода Гаусса состоит в последовательном исключении неизвестных из уравнений системы. Сначала с помощью первого уравнения исключается x_1 из всех последующих уравнений системы. Затем с помощью второго уравнения исключается x_2 из третьего и всех последующих уравнений и т. д.

Обратный ход метода Гаусса состоит в последовательном вычислении искомых неизвестных: решая последнее уравнение, находим единственное в этом уравнении неизвестное x_n . Далее, используя это значение, из предыдущего уравнения вычисляем x_{n-1} и т. д. Последним найдем x_1 из первого уравнения.

Функция, реализовывающая сам метод На вход принимает матрицу (N×N+N)

```
def gauss_method(matrix):  
    ans_det = 1  
  
    def divide_array(array, number):  
        for _ in range(len(array)):  
            array[_] /= number  
  
    for nrow in range(len(matrix)):  
        divider = matrix[nrow][nrow]  
  
        if divider == 0:  
            return 0, []
```

```
ans_det *= divider
divide_array(matrix[nrow], divider)

for k in range(nrow + 1, len(matrix)):
    factor = matrix[k][nrow]
    for j in range(len(matrix[k])):
        matrix[k][j] -= factor * matrix[nrow][j]

for nrow in range(len(matrix) - 1, 0, -1):

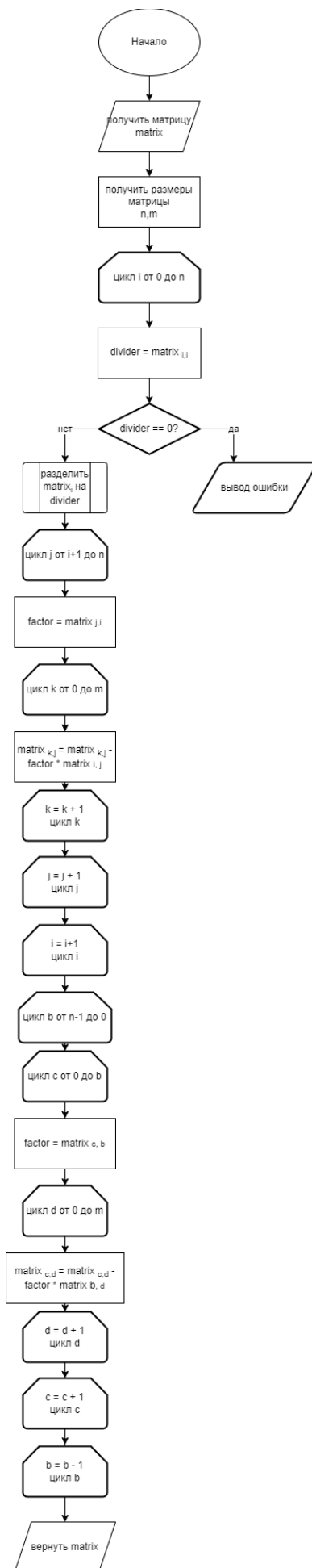
    row = matrix[nrow]

    for new_row in range(nrow):
        factor = matrix[new_row][nrow]
        for j in range(len(matrix[new_row])):
            matrix[new_row][j] -= factor * row[j]

return ans_det, matrix
```

Результатом возвращает детерминант и единичную матрицу.

Блок схема алгоритма:



Пример работы программы

Входной файл input.txt:

```
2 -6 4 -5 2 12
-1 2 0 1 -3 -29
-6 6 10 -1 5 -84
9 -9 -6 -3 -8 51
4 -8 2 10 3 -20
```

Результат работы:

Детерминант: -9580.0

Итоговая матрица:

```
1.0 0.0 0.0 0.0 0.0 -0.2225469728600551
0.0 1.0 0.0 0.0 0.0 -3.194154488517704
0.0 0.0 1.0 0.0 0.0 -10.136534446764074
0.0 0.0 0.0 1.0 0.0 -4.293110647181635
-0.0 -0.0 -0.0 -0.0 1.0 6.180375782881003
```

И ответ:

```
x1 = -0.2225469728600551, невязка: 0.0
x2 = -3.194154488517704, невязка: 0.0
x3 = -10.136534446764074, невязка: 0.0
x4 = -4.293110647181635, невязка: 0.0
x5 = 6.180375782881003, невязка: 0.0
```

входной файл input.txt

```
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

Результат работы:

Решения нет

Вывод

В ходе выполнения работы я научился работать с файлами в python, узнал что такое невязки и как их находить, а так же разобрался с алгоритмом решения СЛАУ методом гаусса при помощи компьютера