

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
Высшего образования
Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа 1 по Вычислительной Математике

Метод Гауса (без выбора главного элемента)

Вариант №11

Группа: Р3210

Выполнил:

Силаев З.А.

Проверил:

Наумова Н.А.

Г. Санкт-Петербург

2025

Оглавление

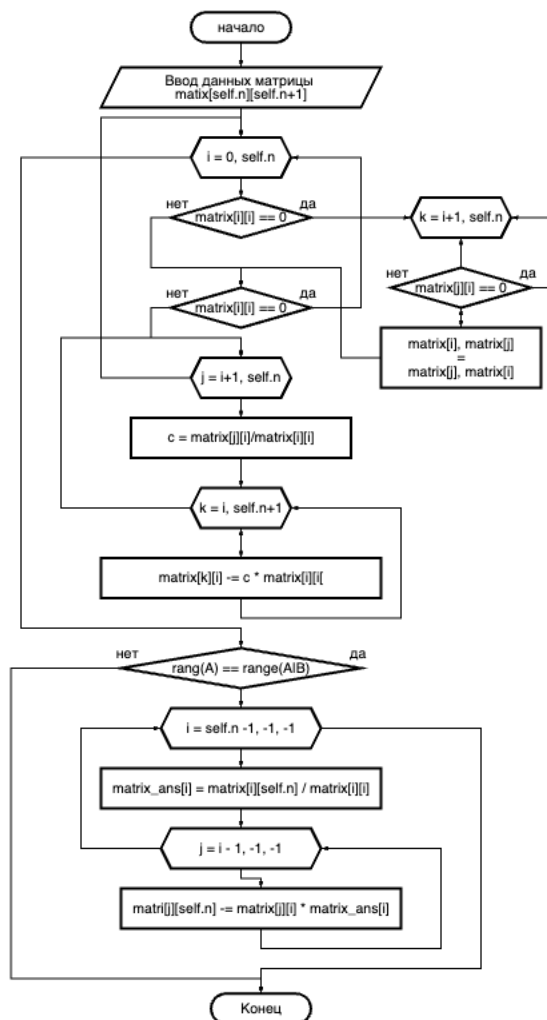
Цель работы	3
Описание метода, расчётные формулы	4
Листинг программы	5
Примеры и результаты работы программы	6
Работа из файла	Error! Bookmark not defined.
Работа из консоли	Error! Bookmark not defined.
Вывод	7

Цель работы

Написать программу, которая будет решать СЛАУ методом Гаусса. Проанализировать полученные результаты, оценить погрешность и на основании этого сделать вывод.

Описание метода, расчётные формулы

Подробнейшее описание работы метода Гаусса приведение на блок схеме ниже. В своем решении я разобью его на две функции, первая будет вычислять треугольную матрицу, вторая уже исходя из нее находить неизвестные.



Листинг программы

Код программы на гитхабе - https://github.com/Chousik/lab_CM

```
def get_triangle(self): 3 usages
    matrix_triangle = [[self.matrix[i][j] for j in range(self.size+1)] for i in range(self.size)]
    for i in range(self.size):
        if matrix_triangle[i][i] == 0:
            for j in range(i + 1, self.size):
                if matrix_triangle[j][i] != 0:
                    matrix_triangle[i], matrix_triangle[j] = matrix_triangle[j], matrix_triangle[i]
                    break
            if matrix_triangle[i][i] == 0:
                continue
            for j in range(i + 1, self.size):
                c = matrix_triangle[j][i] / matrix_triangle[i][i]
                for k in range(i, self.size + 1):
                    matrix_triangle[j][k] -= c * matrix_triangle[i][k]
    return matrix_triangle

def solve(self): 2 usages
    matrix = self.get_triangle()
    if any(all(matrix[i][j] == 0 for j in range(self.size)) for i in range(self.size)):
        print("Ошибка. Матрица не имеет решений или имеет больше одного решения.")
        return []
    matrix_ans = [0 for i in range(self.size)]
    for i in range(self.size - 1, -1, -1):
        matrix_ans[i] = matrix[i][self.size] / matrix[i][i]
        for j in range(i - 1, -1, -1):
            matrix[j][self.size] -= matrix[j][i] * matrix_ans[i]
    return matrix_ans
```

Примеры и результаты работы программы

Работа с файлом

```
1 - ввод с клавиатуры, 2 - из файла: 2
Введите имя файла: 1.csv
Определитель матрицы 22.0
Треугольный вид матрицы [[3, 2, 3, 4], [0.0, 1.3333333333333335, -1.0, -3.333333333333333], [0.0, -4.440892098500626e-16, 5.5, 9.999999999999998]]
Вектор неизвестных [0.2727272727272731, -1.1363636363636362, 1.818181818181818]
Вектор неувязок [0.0, 8.881784197001252e-16, 0.0]
Сравнение с решением с библиотеки numpy
Определитель матрицы с помощью np 22.000000000000004
Разница с нашим решением 3.552713678800501e-15
Вектор неизвестных с помощью np [ 0.27272727 -1.13636364  1.81818182]
Разница с нашим решением [np.float64(1.6653345369377348e-16), np.float64(4.440892098500626e-16), np.float64(4.440892098500626e-16)]
```

Работы с консолью

```
1 - ввод с клавиатуры, 2 - из файла: 1
Введите размер матрицы
3
Введите строки матрицы
3 2 3 4
4 4 3 2
1 4 4 3
[[3, 2, 3, 4], [4, 4, 3, 2], [1, 4, 4, 3]]
Определитель матрицы 22.0
Треугольный вид матрицы [[3, 2, 3, 4], [0.0, 1.3333333333333335, -1.0, -3.333333333333333], [0.0, -4.440892098500626e-16, 5.5, 9.999999999999998]]
Вектор неизвестных [0.2727272727272731, -1.1363636363636362, 1.818181818181818]
Вектор неувязок [0.0, 8.881784197001252e-16, 0.0]
Сравнение с решением с библиотеки numpy
Определитель матрицы с помощью np 22.000000000000004
Разница с нашим решением 3.552713678800501e-15
Вектор неизвестных с помощью np [ 0.27272727 -1.13636364  1.81818182]
Разница с нашим решением [np.float64(1.6653345369377348e-16), np.float64(4.440892098500626e-16), np.float64(4.440892098500626e-16)]
```

Случайные значения

```
/Users/chousik/PycharmProjects/bot/.venv/bin/python /Users/chousik/PycharmProjects/bot/lab1.py
1 - ввод с клавиатуры, 2 - из файла, 3 - случайные значения: 3
Введите размер матрицы
5
Определитель матрицы -2119803017.9999986
Треугольный вид матрицы [[27, 40, 100, 60, 86, 52], [0.0, 87.11111111111111, 75.77777777777777, -10.333333333333332, -7.111111111111111, 2.4444444444444446], [0.0, 0.0, -186.041
Вектор неизвестных [0.39996277616395015, -0.3633208852238742, 0.4069660198021287, -0.8373159293237692, 0.7590255143225767]
Вектор неувязок [0.0, 7.105427357601002e-15, 7.105427357601002e-15, -3.552713678800501e-15, 7.105427357601002e-15]
Сравнение с решением с библиотеки numpy
Определитель матрицы с помощью np -2119803018.000004
Разница с нашим решением 5.4836273193359375e-06
Вектор неизвестных с помощью np [ 0.39996278 -0.36332089  0.40696602 -0.83731593  0.75902551]
Разница с нашим решением [np.float64(5.551115123125783e-17), np.float64(5.551115123125783e-17), np.float64(1.1102230246251565e-16), np.float64(1.1102230246251565e-16), np.float6
Process finished with exit code 0
```

Вывод

В результате выполнения лабораторной работы была получена полностью рабочая программа для решения СЛАУ. При сравнении результатов работы библиотеки и моей программы мы получили почти идентичные результаты. Различия между ними появилось только из-за погрешностей округления, возникающих при работе с вещественными числами.