Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

КТиУ, кафедра Информатики и Прикладной Математики

Лабораторная работа №1

по дисциплине

«Вычислительная математика»

«Решение системы линейных алгебраических уравнений методом Гаусса»

Выполнил:

Студент группы P3217

Григорьев Георгий

Санкт-Петербург

2018 г.

1. Описание метода

Метод Гаусса — классический метод решения системы линейных алгебраических уравнений, суть которого заключается в том, что посредством последовательных исключений неизвестных данная система превращается в ступенчатую, в частности в верхнетреугольную. Далее последовательно находятся все неизвестные, начиная снизу вверх.

Пусть задана система:

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n & = & b_1 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n & = & b_m \end{cases}$$

Тогда с помощью элементарных преобразований над строками систему можно привести к виду

$$\begin{cases} \alpha_{1j_1}x_{j_1} + \alpha_{1j_2}x_{j_2} + \ldots + \alpha_{1j_r}x_{j_r} + \ldots + \alpha_{1j_n}x_{j_n} & = & \beta_1 \\ \qquad \alpha_{2j_2}x_{j_2} + \ldots + \alpha_{2j_r}x_{j_r} + \ldots + \alpha_{2j_n}x_{j_n} & = & \beta_2 \\ & \ldots & \\ \qquad\qquad \alpha_{rj_r}x_{j_r} + \ldots + \alpha_{rj_n}x_{j_n} & = & \beta_r \\ 0 & = & \beta_{r+1} \\ & \ldots & \\ 0 & = & \beta_m \end{cases}$$

где коэффициенты под главной диагональю равны нулю.

Получив треугольную систему, сделаем обратный подъем, находя неизвестные и подставляя их с следующие уравнения для нахождения оставшихся неизвестных.

2. Листинг программы

```
import numpy as np

def gaussSeidel(A, b, x, N, tol):
      """
      :param: A – матрица коэффициентов
      :param: b – столбец свободных членов
      :param: x – начальное приближение
      :param: N – итоговая размерность
      :param: tol – точность
      """

      # максимальное кол-во допустимых итераций
      MAX_ITERATIONS = 1000000
      EPSILON = 1e-13

    # инициализируем список ответов нулями
      xprev = [0.0 for i in range(N)]
      for i in range(MAX_ITERATIONS):
      # заполняем значениями из x
            for j in range(N):
                  xprev[j] = x[j]
        # накапливаем сумму
            for j in range(N):
```

```
            summm = 0.0
            for k in range(N):
                if (k != j):
                    summm += A[j][k] * x[k]
            x[j] = (b[j] - summm) / (A[j][j] + EPSILON)

        diff1norm = []
        oldnorm = []
        for j in range(N):
            diff1norm.append(abs(x[j] - xprev[j]))
            oldnorm.append(abs(xprev[j]))
        if sum(oldnorm) == 0.0:
            norm = diff1norm
        else:
            norm = [a / (b + EPSILON) for a, b in zip(diff1norm, oldnorm)]

        # выход из цикла — проверка условия на погрешность
        if (sum(norm) < tol) and i != 0:
            x_format = ', '.join([f"{a:.4E}" for a in x])
            print(f"Итеративный процесс сходится к [{x_format}].")
            print(f"Заняло {i + 1} итераций.")
            norm_format = ', '.join([f"{a:.4E}" for a in norm])
            print(f"Столбец погрешностей: [{norm_format}].")
            return
    print("Итеративный процесс не сошелся.")

if __name__ == '__main__':
    n = input("Введите размерность: (Enter для 2): ")
    if not n:
        n = 2
    n = int(n)

    tol = input("Введите точность (Enter для 1e-13): ")
    if not tol:
        tol = 1e-13
    tol = float(tol)

    if input("Хотите ввести свои значения коэффициентов? д/н ")[0].lower() == "д":
        matrix = []
        for _ in range(n):
            matrix.append(list(map(float, input().split())))
        vector = list(map(float, input().split()))
    else:
        if input("Хотите использовать случайные значения коэффициентов? д/н ")
[0].lower() == "д":
            matrix = np.random.random((n, n)).tolist()
            vector = np.random.random(n).tolist()
            print("\n".join([" ".join(list(map(str, a))) for a in matrix]))
            print(vector)
            gaussSeidel(matrix, vector, np.zeros(n).tolist(), n, tol)
        else:
            matrix2 = [
            [1, 2],
            [0, 0]]
            vector2 = [3, 0]

            matrix6 = [
            [15, 1, -6, 2, 1, 4],
            [1, 40, 3, -8, 4, 3],
            [2, 7, 62, 4, -7, 3],
            [1, 3, 1, 48, 3, -5],
            [4, 7, 5, 5, 99, 6],
            [1, 1, 1, 1, 1, -7]]
            vector6 = [0, 2, 1, 1, 4, -8]
```

```
n, tol = 6, 1e-13
gaussSeidel(matrix6, vector6, np.zeros(n).tolist(), n, tol)
print()
n, tol = 2, 1e-13
gaussSeidel(matrix2, vector2, np.zeros(n).tolist(), n, tol)
print()
```

3. Примеры

6
15 1 -6 2 1 4
1 40 3 -8 4 3
2 7 62 4 -7 3
1 3 1 48 3 -5
4 7 5 5 99 6
1 1 1 1 1 -7

0 2 1 1 4 -8

-3.3016E-01, 8.9092E-03, -3.9430E-02, 1.4474E-01, -1.9431E-02, 1.1092E+00

2
1 2
0 0

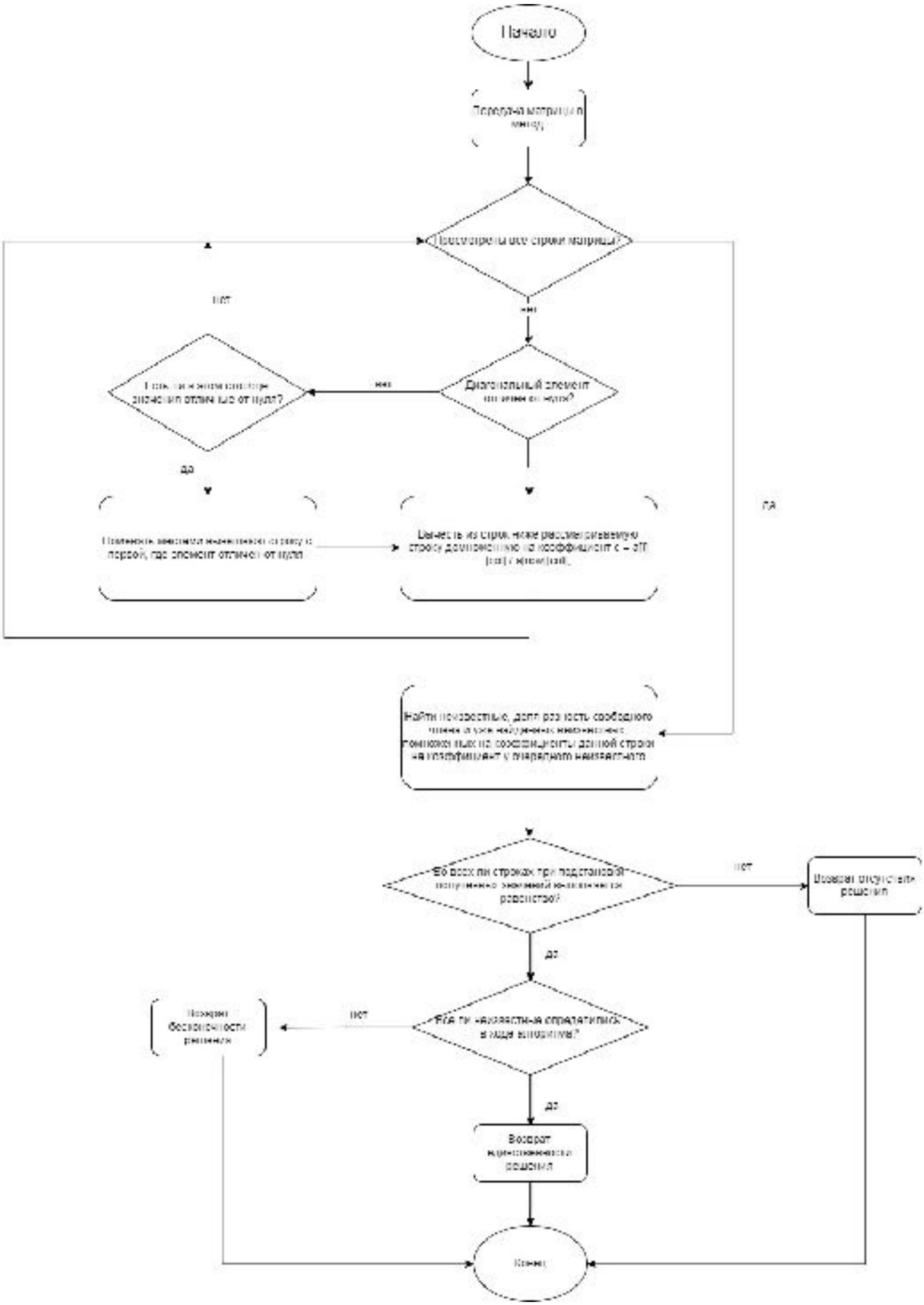
3 0

3.0000E+00, 0.0000E+00

3
0.7286592874963329 0.6842175247955062 0.8634558785876988
0.9218036475487008 0.706502619273745 0.28558474223028596
0.30899201324622816 0.3649644690806928 0.3991518877087774
0.39043399353734254 0.7773724551689634 0.7791683007655045

-5.5031E+00, 9.1518E+00, -2.1559E+00

Вывод: метод Гаусса-Зейделя показал хорошие результаты для выбранных вручную значений коэффициентов, но часто не сходится для случайных значений при высокой точности. Также при бесконечном решении он все равно находит одно из них.



Начало
Рассмотрены все строки матрицы?
нет
да
нет
да
да
нет
да
нет
да