



УНИВЕРСИТЕТ ИТМО

**Системное и прикладное программное обеспечение.
Программная инженерия.**

Лабораторная работа №1.

Дисциплина: Вычислительная математика.

Преподаватель: Малышева Татьяна Алексеевна.

Выполнил: Бусыгин Иван.

Группа: Р3212.

Вариант: 6.

Санкт-Петербург
2022 год

Цель работы.

Реализовать алгоритм простых итераций для решения системы линейных алгебраических уравнений.

Задание.

В программе численный метод должен быть реализован в виде отдельной подпрограммы или класса, в который входные/выходные данные передаются в качестве параметров.

Размерность матрицы $n \leq 20$.

Должна быть реализована возможность ввода коэффициентов матрицы, как с клавиатуры, так и из файла (по выбору конечного пользователя).

Точность задается с клавиатуры/файла.

Проверка диагонального преобладания (в случае, если диагональное преобладание в исходной матрице отсутствует, сделать перестановку строк/столбцов до тех пор, пока преобладание не будет достигнуто). В случае невозможности достижения диагонального преобладания выводить соответствующее сообщение.

Вывод вектора неизвестных: x_1, x_2, \dots, x_n

Вывод количества итераций, за которое было найдено решение.

Вывод вектора погрешностей: $|x_i^{(k)} - x_i^{(k-1)}|$

Описание метода.

В начале мы стараемся расставить строки матрицы так, чтобы достигалось диагональное преобладание. Для этого поочередно в каждом столбце ищется элемент, который больше суммы всех остальных в строке (в левой части системы), и соответствующее уравнение переставляется на первую позицию, следующее на вторую и т.д.. Если в какой-то момент преобладающего элемента в столбце не оказывается, выводим соответствующее сообщение.

Далее выражаем в уравнениях диагональные неизвестные через остальные. В качестве первого приближения для неизвестных берём значения получившихся свободных членов. А далее на каждой итерации вычисляем новые значения неизвестных через полученные выражения, подставляя значения, полученные на предыдущей итерации.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \quad x_i^{(k+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j^k, \quad i = 1, 2, \dots, n$$

Класс на Python, реализующий данный метод.

```
class EquationsSystem:
    def __init__(self, A, b):
        self._A = A.astype('float64')
        self._b = b.astype('float64')

    def moveRow(self, array, indexFrom, indexTo):
        row = array[indexFrom]
        array = np.delete(array, indexFrom, axis = 0)
        return np.insert(array, indexTo, row, axis = 0)

    def orderRows(self):
        absA = np.absolute(self._A)
        for j in range(equationsAmount):
            for i in range(equationsAmount - j):
                if absA[j + i][j] >= absA[j + i][j].sum() + absA[j + i][j + 1:].sum():
                    absA = self.moveRow(absA, j + i, j)
                    self._A = self.moveRow(self._A, j + i, j)
                    self._b = self.moveRow(self._b, j + i, j)
                    break
            else:
                return False
        return True

    def solveSystem(self, accuracy):
        C = self._A.copy()
        d = self._b.copy()

        for i in range(equationsAmount):
            d[i] /= C[i][i]
            C[i] = C[i] / -C[i][i]
            C[i][i] = 0

        oldX = d.copy()
        maxChange = np.inf
        iteration = 1

        while maxChange > accuracy:
            newX = C @ oldX + d
            changes = np.absolute(newX - oldX)
            maxChange = changes.max()
            oldX = newX
            iteration += 1

        if accuracy < 1:
            newX = np.around(newX, decimals = math.ceil(math.log10(1 / accuracy)))

        return (newX, changes, iteration)
```

Примеры работы программы.

Данные из файла:

Желаете ли вы использовать файл с входными данными? Если да, введите 'y': y

Введите путь к файлу: Данные.txt

Вектор неизвестных: [1.2619 -0.83334 0.88095]

Вектор погрешностей: [9.98069191e-06 9.41346338e-06 6.74590767e-06]

Число итераций: 72

Программа завершила работу.

Система, приводимая к виду с диагональным преобладанием:

Желаете ли вы использовать файл с входными данными? Если да, введите 'y':

Введите точность (погрешность) вычислений: 0.00001

Введите количество неизвестных: 2

Введите через пробел на отдельных строках коэффициенты системы 2 уравнений:

3.3 2 3

-1 2.12 5

Вектор неизвестных: [-0.40462 2.16763]

Вектор погрешностей: [3.31443315e-06 8.59876524e-06]

Число итераций: 21

Программа завершила работу.

Система, не приводимая к виду с диагональным преобладанием:

Желаете ли вы использовать файл с входными данными? Если да, введите 'y':

Введите точность (погрешность) вычислений: 0.1

Введите количество неизвестных: 3

Введите через пробел на отдельных строках коэффициенты системы 3 уравнений:

1 2 5 2

3 3 2 1

4 1 2 4

Данную систему невозможно привести к виду, удовлетворяющему условию диагонального преобладания.

Программа завершила работу.

Выводы.

Алгоритм реализован, и теперь я могу использовать его вместо онлайн сервисов для решения систем уравнений.