Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

‘ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА’

Вариант №7

*Выполнил:*

Студент группы P32091

Кравец Роман Денисович

*Преподаватель:*

Рыбаков Степан Дмитриевич



Санкт-Петербург, 2023

**Цель работы**

Изучить численные методы решения систем линейных алгебраических уравнений и реализовать один из них средствами программирования.

**Описание метода**

Метод простых итераций

Суть метода заключается в нахождении по приближённому значению величины следующего приближения, являющегося более точным.

**Листинг программы**

def diagonal(matrix):

    # Диагональное преобладание

    sorted = dict()

    column = 0

    for i in range (len(matrix)):

        maxValue = sys.float\_info.min

        sum = 0

        for j in range (len(matrix)):

            previous = maxValue

            maxValue = max(maxValue, matrix[i][j])

            if (previous != maxValue):

                column = j

            sum += abs(matrix[i][j])

        sum -= maxValue

        if (maxValue>=sum):

            if (column in sorted):

                print("Приведение к диагональному представлению невозможно")

                return matrix

            else:

                sorted[column] = matrix[i]

        else:

            print("Приведение к диагональному представлению невозможно")

            return matrix

    result = []

    #Переставим уравнения местами так, чтобы выполнялось условие преобладания диагональных элементов

    for i in range(len(matrix)):

        if (i in sorted):

            result.append(sorted.get(i))

        else:

            print("Приведение к диагональному преобладанию невозможно")

            return matrix

    print("Переставили строки так, чтобы выполнялось условие преобладания диагональных элементов:")

    for row in result:

        for col in row:

            print('{:10}'.format(round(col, 3)), end='')

        print()

    return result

def transform(matrix):

    #Выражаем из первого уравнения x\_1, из второго x\_2, из третьего x\_3

    result = [[0] \* len(matrix[0]) for i in range(len(matrix))]

    for i in range(len(matrix)):

        element = matrix[i][i]

        for j in range(len(matrix[0])):

            if (i != j):

                result[i][j] = matrix[i][j] / element

                if (j != (len(matrix[0])-1)):

                    result[i][j] \*= -1

    print("Матрица C:")

    for i in range(len(result)):

        for j in range(len(result[0])-1):

            # print(result[i][j])

            print('{:10}'.format(round(result[i][j], 3)), end='')

        print()

    print("Матрица d:")

    for i in range(len(result)):

        print('{:10}'.format(round(result[i][len(result[0])-1], 3)))

    return result

def check(matrix):

    #Проверяем условие сходимости

    maxValue = sys.float\_info.min

    for i in range(len(matrix)):

        sum = 0

        for j in range(len(matrix[0])-1):

            sum += matrix[i][j]

        maxValue = max(sum, maxValue)

    return maxValue < 1

def iterate(matrix, precision, limit):

    # матрица d

    current = []

    for i in range(len(matrix)):

        current.append(matrix[i][len(matrix[0])-1])

    maxValue = sys.float\_info.min

    criteria = sys.float\_info.min

    counter = 0

    tmp = 0

    while True:

        previous = current.copy()

        for i in range(len(matrix)):

            for j in range(len(matrix[0])-1):

                    tmp += matrix[i][j] \* previous[j]

            tmp += matrix[i][len(matrix[0])-1]

            current[i] = tmp

            maxValue = max(maxValue, abs(current[i] - previous[i]))

            tmp = 0

        criteria = maxValue

        maxValue = sys.float\_info.min

        counter+=1

        if (counter == limit or criteria < precision):

            break

    if (counter == limit):

        print("Результат не получен за " + str(limit) + " итераций")

        return None

    #Вектор погрешностей

    for i in range(len(current)):

        previous[i] = abs(current[i]-previous[i])

    print("Вектор погрешностей: ")

    for col in previous:

        print(col)

    print("Результат получен за " + str(counter) + " итераций")

    return current

def solve(matrix, precision):

    matrix = diagonal(matrix)

    matrix = transform(matrix)

    if (check(matrix)):

        print("Достаточное условие сходимости выполняется")

    else:

        print("Достаточное условие сходимости не выполняется")

    return iterate(matrix, precision, 100000)

**Примеры работы программы**

Лабораторная работа #1 (7 вариант)

Метод простых итераций

Взять коэффициенты из файла (file) или ввести с клавиатуры (console) ?

console

Введите порядок матрицы: 3

Вводите коэффициенты матрицы через пробел строка за строкой:

2 2 10 14

10 1 1 12

2 10 1 13

Введите желаемую точность: 0.01

Переставили строки так, чтобы выполнялось условие преобладания диагональных элементов:

10.0 1.0 1.0 12.0

2.0 10.0 1.0 13.0

2.0 2.0 10.0 14.0

Матрица C:

0 -0.1 -0.1

-0.2 0 -0.1

-0.2 -0.2 0

Матрица d:

1.2

1.3

1.4

Достаточное условие сходимости выполняется

Вектор погрешностей:

0.0019320000000000448

0.0024600000000001288

0.0030840000000000867

Результат получен за 5 итераций

Вектор неизвестных:

1.0

0.999

0.999

**Вывод**

В результате выполнения данной лабораторной работой я познакомился с численными методами решения математических задач на примере систем алгебраических уравнений, реализовав на языке программирования Python метод простой итерации.