# Лабораторная работа №6 по теме: «Метод сеток решения краевой задачи для уравнения эллиптического типа»

## Подготовил студент Михайлов Денис

## группы Б8117-02.03.01

**Постановка задачи:**

Получить решение краевой задачи для уравнения эллиптического типа.

**Дано:**

С начально-краевыми условиями:



**Ход решения:**

Решим краевую задачу с помощью схемы:



, где .

На границе области имеем:

Мы имеем:

Преобразуем схему и выразим :

Заметим, что формула существует, если . Решим полученную систему уравнений с помощью метода Зейделя. Для этого введем , параметр и возьмем начальное приближение . Для нахождения решения запустил цикл с следующим условием:

Далее находим нашу матрицу решений с помощью формулы , и формул (1), также нужно изменять параметр . Если после нахождения матрицы решений условие все еще выполняется, то в качестве начальных приближений берем уже найденную матрицу решений, после чего снова решаем вышеописанным способом. Если не выполняется, то в качестве ответа записываем текущую полученную матрицу решений.

Получим решение для наших условий:

[0.0 0.02 0.1 0.22 0.4 0.62 0.89 1.21 1.59 2.01 2.48]

[0.02 0.15 0.29 0.45 0.64 0.87 1.14 1.44 1.77 2.13 2.5]

[0.07 0.26 0.45 0.64 0.86 1.09 1.35 1.64 1.94 2.24 2.55]

[0.15 0.38 0.6 0.82 1.05 1.29 1.54 1.81 2.09 2.37 2.63]

[0.26 0.52 0.76 0.99 1.22 1.47 1.72 1.98 2.25 2.5 2.74]

[0.41 0.68 0.92 1.16 1.39 1.64 1.89 2.15 2.41 2.66 2.89]

[0.6 0.85 1.09 1.32 1.55 1.8 2.05 2.31 2.58 2.83 3.07]

[0.81 1.04 1.26 1.48 1.71 1.95 2.2 2.47 2.75 3.03 3.29]

[1.06 1.25 1.44 1.64 1.85 2.08 2.34 2.63 2.93 3.24 3.54]

[1.34 1.47 1.61 1.77 1.97 2.2 2.46 2.76 3.1 3.45 3.82]

[1.65 1.68 1.75 1.88 2.05 2.27 2.55 2.87 3.24 3.66 4.13]

**Код программы**

def Seidel(n):

    N = n - 1

    h = 1 / n

    x = [i \* h for i in range(n)]

    y = [i \* h for i in range(n)]

    u = [[0 for i in range(n)] for j in range(n)]

    M = 1

    eps = 0.0001

    for i in range(n):

        u[0][i] = func(x[0], y[i])

        u[N][i] = func(x[N], y[i])

        u[i][0] = func(x[i], y[0])

        u[i][N] = func(x[i], y[N])

    while eps < M:

        M = 0

        for i in range(1, N):

            for j in range(1, N):

                L = (u[i+1][j] + u[i-1][j] + u[i][j+1] + u[i][j-1]) / 4

                M = max(abs(u[i][j] - L), M)

                u[i][j] = L

    return u

def func(x, y):

    return 2 \* x \*\* 2 + 3 \* y \*\* 2