

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ
к лабораторной работе
на тему

Метод сеток решения задачи Дирихле для уравнения Пуассона

Выполнил: студент группы 153501
Тимофеев Кирилл Андреевич

Проверил: Анисимов Владимир Яковлевич

Минск 2023

Цели выполнения задания

- изучить метод разностных аппроксимаций для уравнения Пуассона;
- составить алгоритмы решения задачи Дирихле для уравнения Пуассона методом сеток применимыми для организации вычислений на ПЭВМ;
- составить программы решения задачи Дирихле для уравнения Пуассона по разработанным алгоритмам;
- выполнить тестовые примеры и проверить правильность работы программ;
- получить численное решение заданной задачи Дирихле для уравнения Пуассона.

Краткие теоретические сведения

Пусть $D = \{(x, y): 0 < x < 1, 0 < y < 1\}$ – открытый квадрат, Γ – его граница, $\bar{D} = D \cup \Gamma$ – замкнутый квадрат, $f(x, y)$ – заданная на \bar{D} достаточно гладкая функция. Задача Дирихле состоит в следующем. Требуется найти непрерывную на \bar{D} функцию $u(x, y)$, удовлетворяющую на открытом квадрате D уравнению Пуассона

$$-\Delta u \equiv -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x, y) \quad (2.48)$$

и принимающую на границе квадрата значение равное нулю, т. е.

$$u(x, y) = 0 \quad \text{на } \Gamma. \quad (2.49)$$

Задача Дирихле (2.48), (2.49) имеет единственное решение $u(x, y)$.

Положим $h = l/N$, $x_k = kh$, $y_m = mh$, $f_{km} = f(x_k, y_m)$. Построим сетки

$$\omega_h = \{(x_k, y_m): k, m = 0, 1, \dots, N\},$$

$$\omega'_h = \{(x_k, y_m): k, m = 1, 2, \dots, N-1\},$$

таблица 2.1.

$\omega_h^* = \omega_h \setminus \omega'_h$ (ω_h^* — множество узлов, лежащих на Γ). Зададим нормы

$$\|v\|_h = \max_{\omega_h} |v_{km}|, \quad \|v\|'_h = \max_{\omega'_h} |v_{km}|.$$

Разностная схема:

$$\Delta v_{km} = f_{km}, \quad k, m = 1, 2, \dots, N-1, \quad (2.50)$$

$$v_{km} = 0 \quad \text{на} \quad \omega_h^*, \quad (2.51)$$

Разностное уравнение (2.50) в более подробной записи имеет вид

$$-\frac{v_{k-1,m} - 2v_{km} + v_{k+1,m}}{h^2} - \frac{v_{k,m-1} - 2v_{km} + v_{k,m+1}}{h^2} = f_{km}. \quad (2.52)$$

Его шаблон изображен на рис. 2.6

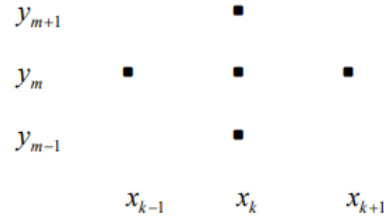


Рис. 2.6

Решение v разностной задачи Дирихле находится методом последовательных приближений по схеме переменных направлений, где $f_{km}^{v-1/2} = f(x_k, y_m)$, v_{km}^0 — произвольные. Можно доказать, что $\lim_{V \rightarrow \infty} v_{km}^V = v_{km}$, $k, m = 1, 2, \dots, N-1$, при любых начальных приближениях v_{km}^0 , причем наибольшая скорость сходимости достигается при $\tau \approx h/\pi$. Здесь положена в основу идея о стабилизации при $t \rightarrow +\infty$ решения уравнения теплопроводности к решению уравнения Пуассона, если f не зависит от t .

Разностная схема (2.51), (2.52) устойчива по правой части, т. е. разностная задача

$$\begin{aligned} \Delta z_{km} &= \xi_{km}, \quad k, m = 1, 2, \dots, N-1, \\ z_{km} &= 0 \quad \text{на} \quad \omega^* \end{aligned}$$

при любом $h = 1/N$, $N \geq 2$, имеет единственное решение z и это решение удовлетворяет неравенству

$$\|z\|_h \leq c \|\xi'\|_h, \quad (2.53)$$

где c — некоторая постоянная, не зависящая от h и сеточной функции ξ .

Предположим, что решение задачи Дирихле (2.48), (2.49) достаточно гладкое на замкнутом квадрате \bar{D} , а именно, $u(x, y) \in C_4(\bar{D})$. Тогда разностное уравнение (2.51) аппроксимирует дифференциальное уравнение (2.48) на решение задачи (2.48), (2.49) со вторым порядком относительно h , т. е.

$$\|\psi\|'_h = O(h^2), \quad (2.54)$$

где

$$\psi_{km} = \Lambda u_{km} - f_{km}, \quad (2.55)$$

есть невязка для разностного уравнения. При получении оценки (2.54) используется тот факт, что частным производным u''_{xx} и u''_{yy} , входящим в уравнение (2.48), в разностном уравнении (2.52) отвечают вторые разностные производные, аппроксимирующие указанные частные производные с точностью второго порядка по h . Поскольку краевое условие (2.49) аппроксимируется на сетке ω_h^* согласно (2.53) точно, то из (2.54) и устойчивости разностной схемы (2.50), (2.52) по правой части вытекает сходимость ее решения v к решению $u \in C_4(\bar{D})$ задачи (2.48), (2.49) со вторым порядком относительно h , т. е.

$$\|u - v\|_h = O(h^2). \quad (2.56)$$

Действительно, из уравнения (2.52), равенства (2.56) и условий (2.49), (2.53) вытекает, что погрешность $r = u - v$ на сетке ω_h является решением разностной задачи

$$\begin{aligned} \Lambda r_{km} &= \psi_{km}, & k, m = 1, 2, \dots, N-1, \\ r_{km} &= 0 \text{ на } \omega_h^*. \end{aligned}$$

таблица 2.1.

Отсюда и из (2.54), (2.55) следует (2.56). Разностная схема (2.50), (2.52) обладает вторым порядком точности.

Случай произвольной области.

Рассмотрим задачу Дирихле

$$-\Delta u(x, y) = f(x, y) \quad \text{на } G, \quad (2.57)$$

где

$$u(x, y) = \varphi(x, y) \quad \text{на } \Gamma, \quad (2.58)$$

где G – некоторая конечная область (рис.2.7), Γ – граница области G ;

$f(x, y)$ – заданная на области G функция; $\varphi(x, y)$ – заданная на границе Γ функция.

Строится, как и ранее, квадратная сетка с шагом h . Во всех расположенных в области G узлах сетки, которые можно соединить с четырьмя ближайшими узлами отрезками прямых, не пересекая границу Γ , разностное уравнение задается в следующем виде:

$$\Lambda v_{km} = f_{km}, \quad (2.59)$$

где Λ – оператор (2.52). Указанные узлы обозначены на рис. 2.7 точками. Шаблон разностного уравнения (2.57) показан на рис. 2.6. В узлах, находящихся в области G вблизи ее границы Γ (отмеченных на рис. 2.7 треугольниками), для задания разностных уравнений применяется линейная интерполяция в направлении оси x или оси y . Например, в точке с номером 0 уравнение имеет вид

$$v_0 = \frac{\rho_1}{\rho_1 + \rho_2} v_2 + \frac{\rho_2}{\rho_1 + \rho_2} \varphi_1, \quad (2.60)$$

где ρ_1 – расстояние от точки 0 до точки 1 на границе Γ , в которой берется заданное значение функции φ , обозначенное через φ_1 ; v_0, v_2 – неизвестные в точках 0, 2; $\rho_2 = h$ – расстояние между этими точками.

Здесь для простоты используется один индекс. Формула (2.60) означает линейную интерполяцию между точками 1, 2 в точку 0.

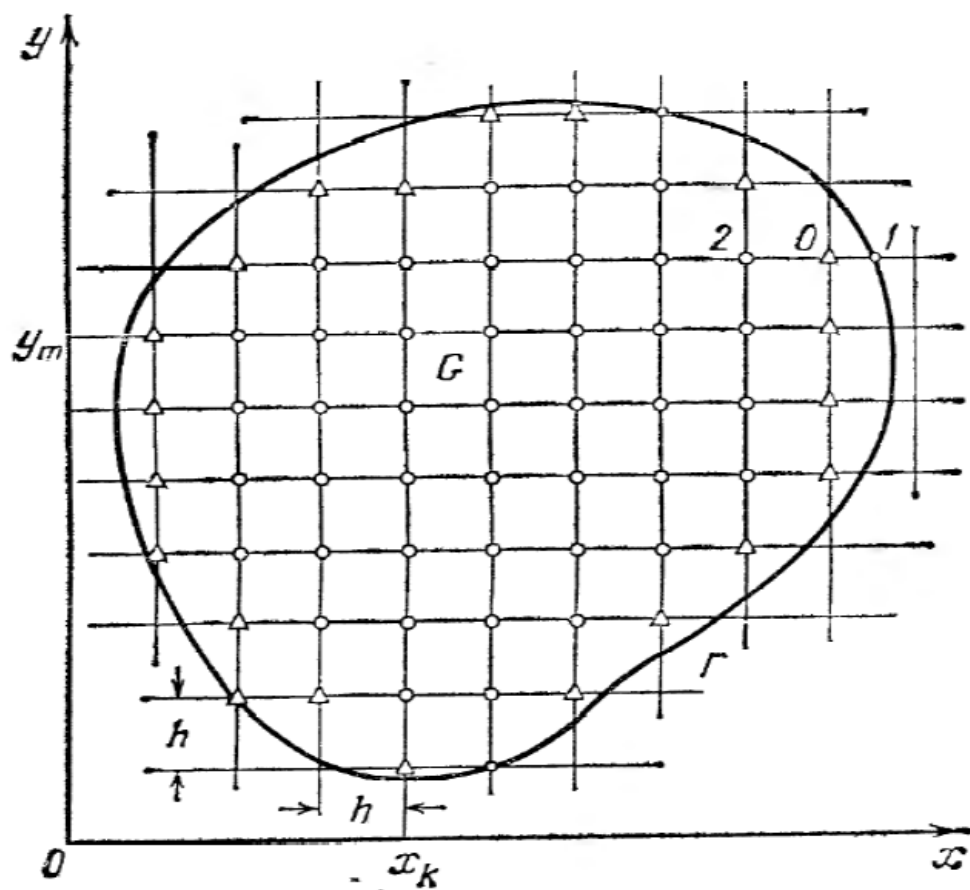


Рис. 2.7.

Аналогично разностные уравнения задаются в остальных узлах, обозначенных треугольниками. При этом расстояния от точки, в которую производится интерполяция, до обеих крайних точек не должны превышать h и одна или обе крайние точки должны лежать на границе Γ . Уравнения (2.59), имеющие более подробную запись (2.52), разрешим относительно v_{km} :

$$v_{km} = \frac{v_{k-1,m} + v_{k+1,m} + v_{k,m-1} + v_{k,m+1}}{4} + \frac{h^2}{4} f_{km}. \quad (2.61)$$

Итак, в каждом узле, обозначенном кружком, задано уравнение (2.61), а в каждом узле, отмеченном треугольником, уравнение имеет вид (2.60). Общее число уравнений совпадает с числом неизвестных. Полученная **таблица 2.1.**

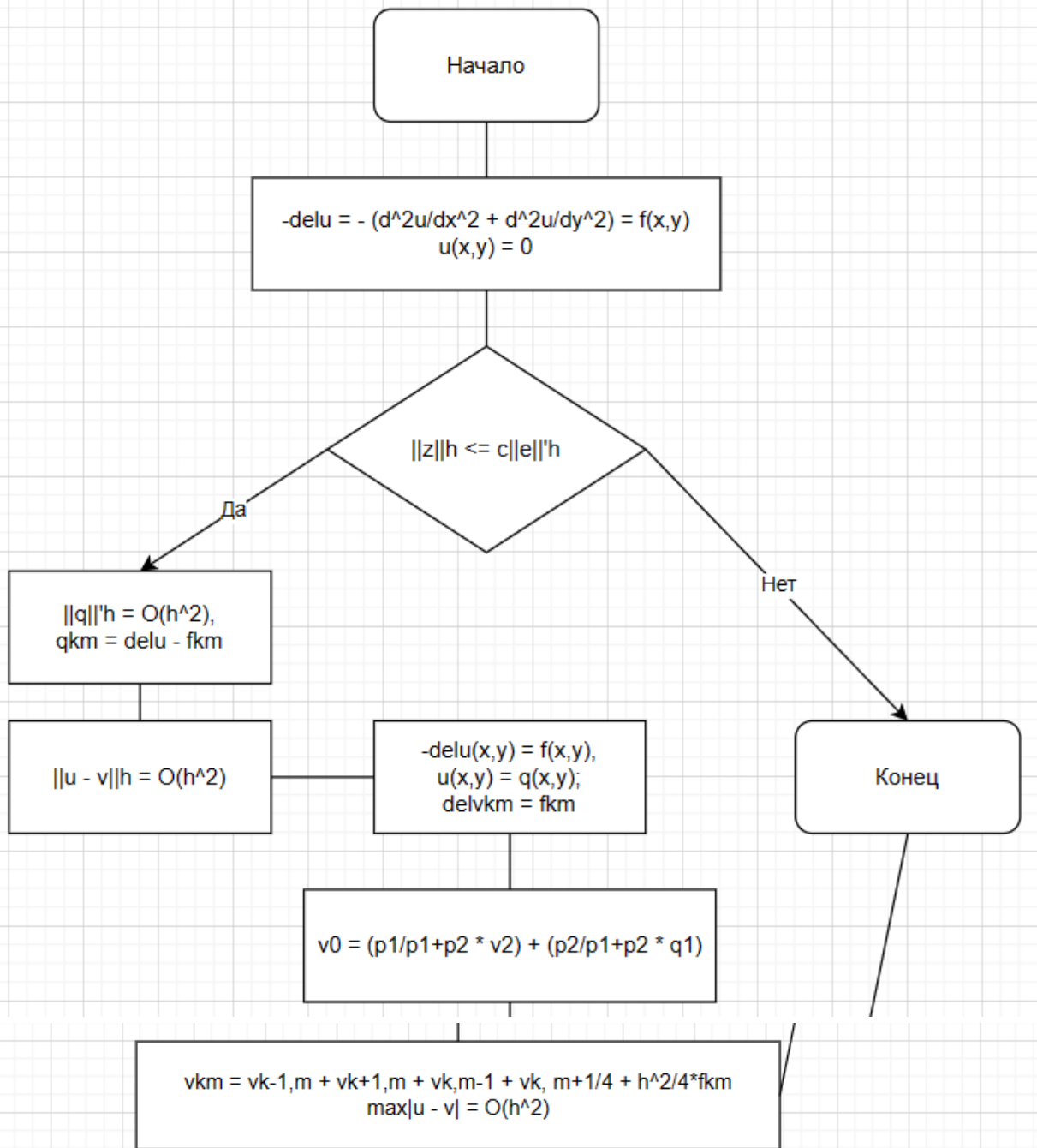
система линейных алгебраических уравнений имеет единственное решение v , для нахождения которого могут быть применены методы простых итераций и Зейделя.

Если $u(x, y) \in C_4(\bar{G})$, решение задачи Дирихле то справедлива оценка

$$\max_{G_h} |u - v| = O(h^2), \quad (2.62)$$

где G_h – множество всех узлов, обозначенных кружками и треугольниками.

Решение $u(x, y)$ принадлежит классу $C_4(\bar{G})$, например, если граница Γ обладает трижды непрерывно дифференцируемой производной, функция φ длины s дуги границы Γ имеет ограниченную пятую производную, а $f(x, y) \in C_3(\bar{G})$.



Задание

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ №15

Пластина прямоугольной формы с вырезом на одной из сторон жестко закреплена по краям и равномерно нагружена по площади рис 2.8. Прогиб пластины определяется из уравнения Пуассона. Рассчитать прогиб $W(x, y)$ по данным, приведенным в табл. 2.11 A, B – размеры пластины; h – ее толщина; R – радиус выреза; P – нагрузка; E – модуль упругости; ν – коэффициент Пуассона. Граничное условие $W=0$.

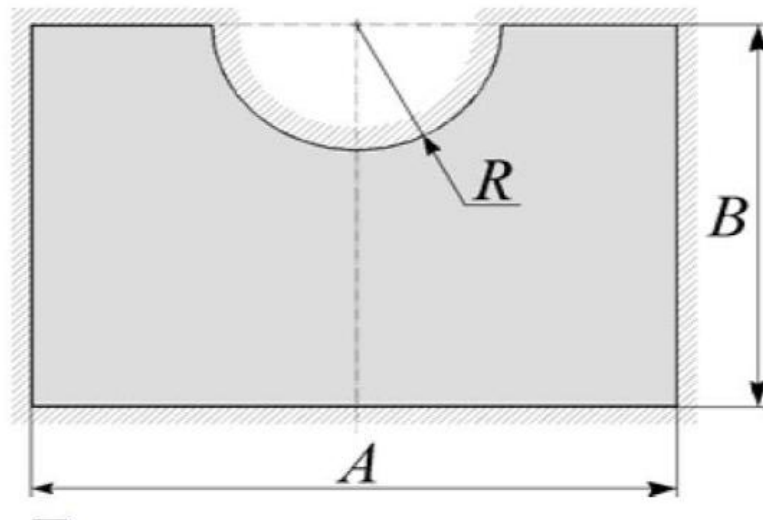


Рис 2.8

$$\left(\frac{\partial^2 w(x, y)}{\partial x^2} + \frac{\partial^2 w(x, y)}{\partial y^2} \right) = P/D,$$

где $D = Eh^3 / [12(1-\nu^2)]$ – изгибная жесткость, E – модуль упругости, h – толщина пластины, ν – коэффициент Пуассона.

Таблица исходных данных к заданию

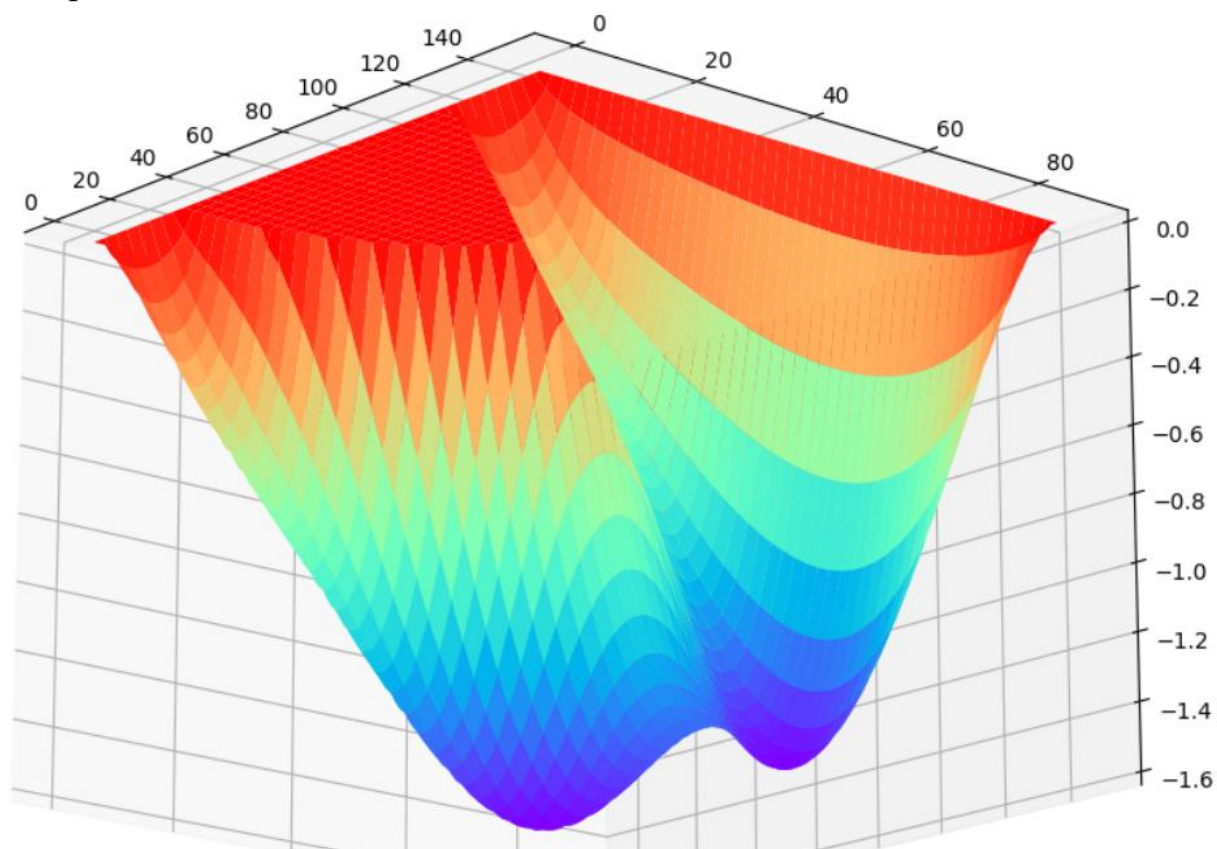
Таблица 2.11

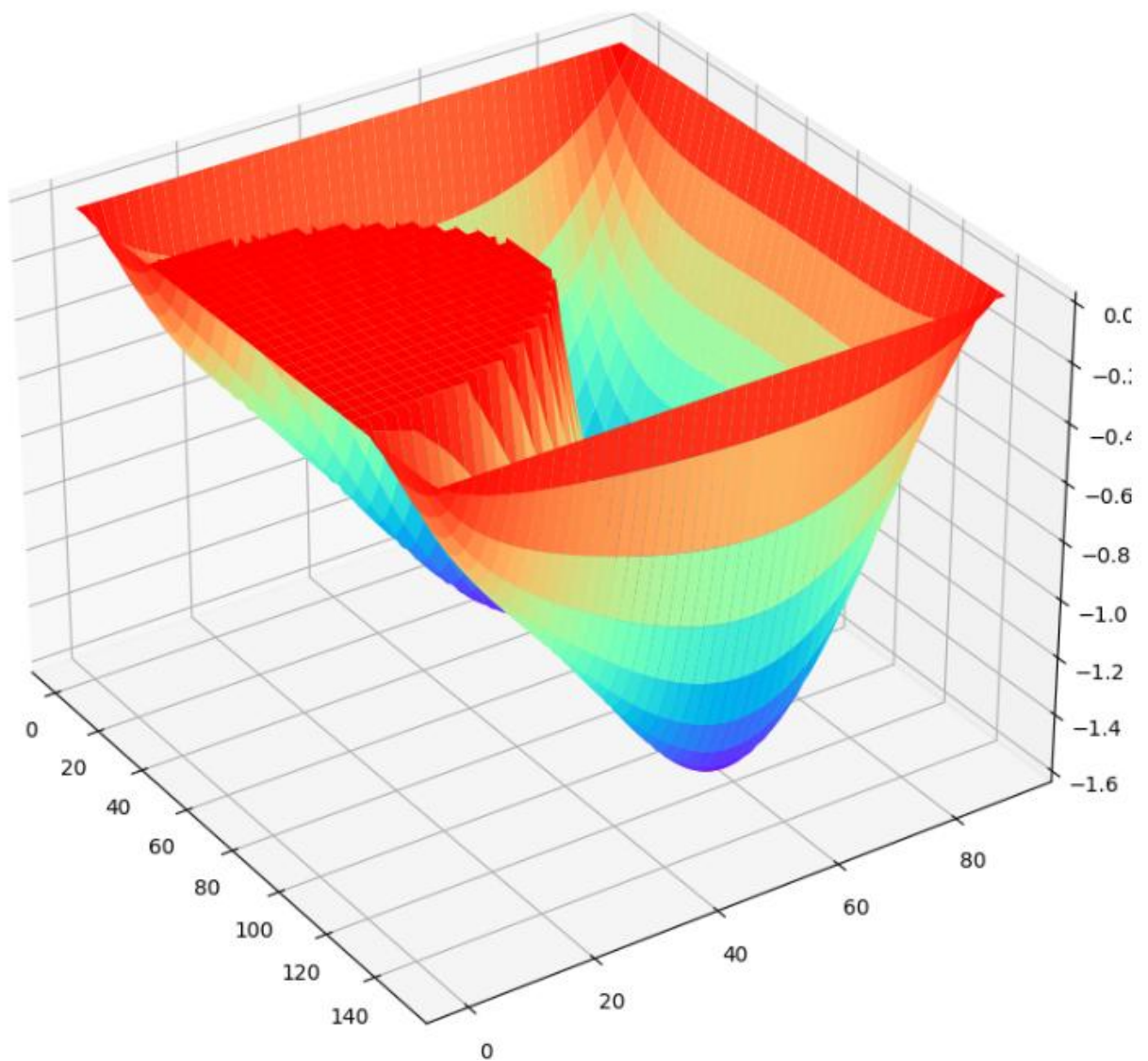
Номер варианта	Параметры						
	A , мм	B , мм	R , мм	h , мм	P ,	E , Н/м ²	ν
5	180	90	35	5	$110 \cdot 10^9$	120	0.3

Тестовое задание

$A = 150$; $B = 90$; $R = 45$; $h = 6$; $P = 65 \cdot 10^9$; $E = 140$; $\nu = 0.28$

Шаг равен единице.





Программная реализация

```
import numpy as np
import matplotlib.pyplot as plt
import typing as tp
import scipy.sparse.linalg as spa_linalg

from math import *

VARIANT_NUMBER = 5
width = 180
height = 90
radius = 35
thickness = 5
P = 110*10^9
E = 120
v = 0.3
step = 1

function = lambda x, y: P / (E * thickness**3 / (12 * (1 - v**2)))

def solve(width: float, height: float, radius: float,
```

```

        function: tp.Callable[[float, float], float], step: float) ->
np.array:
    cols_amount = int(ceil(width / step)) + 1
    rows_amount = int(ceil(height / step)) + 1

    total_equations_amount = cols_amount * rows_amount
    A = np.zeros((total_equations_amount, total_equations_amount))

    b = np.zeros(total_equations_amount)

    def get_mapped_index(i: int, j: int) -> int:
        return cols_amount * i + j

    def is_bound_point(i: int, j: int) -> bool:
        x, y = i * step, j * step
        if np.isclose(x, 0) or np.isclose(y, 0):
            return True
        elif x >= height or y >= width:
            return True
        elif x ** 2 + (y - width / 2) ** 2 <= radius ** 2:
            return True
        else:
            return False

    for i in range(rows_amount):
        for j in range(cols_amount):
            if not is_bound_point(i, j):
                A[get_mapped_index(i, j), get_mapped_index(i, j - 1)] = 1
                A[get_mapped_index(i, j), get_mapped_index(i, j + 1)] = 1
                A[get_mapped_index(i, j), get_mapped_index(i - 1, j)] = 1
                A[get_mapped_index(i, j), get_mapped_index(i + 1, j)] = 1
                A[get_mapped_index(i, j), get_mapped_index(i, j)] = -4

                b[get_mapped_index(i, j)] = function(i * step, j * step) *
step ** 2
            else:
                A[get_mapped_index(i, j), get_mapped_index(i, j)] = 0

    solution = spa_linalg.cg(A, b)[0]
    return solution.reshape((rows_amount, cols_amount))

def plot_solution(solution: np.array, width: float, height: float, step:
float) -> None:
    y = np.arange(0, height + step, step)
    x = np.arange(0, width + step, step)
    X, Y = np.meshgrid(x, y)
    # print('X:', X, 'Y:', Y)
    fig = plt.figure(figsize=(10, 10))
    ax = plt.axes(projection='3d')
    ax.plot_surface(X, Y, solution, cmap='rainbow')
    plt.show()

if __name__ == '__main__':
    # solution = solve(width, height, radius, function, thickness)
    # plot_solution(solution, width, height, thickness)

    solution = solve(width, height, radius, function, step)
    print(solution)
    plot_solution(solution, width, height, step)

```

```
# solution = solve(width, height, radius, function, 8)
# plot_solution(solution, width, height, 8)
#
# solution = solve(width, height, radius, function, 16)
# plot_solution(solution, width, height, 16)
```

Полученные результаты

Требуется промоделировать следующий процесс: пластина прямоугольной формы с вырезом на одной из сторон жестко закреплена по краям и равномерно нагружена по площади рис 2.8. Прогиб пластины определяется из уравнения Пуассона как функции $W(x, y)$.

$\frac{\delta^2 w(x,y)}{\delta x^2} + \frac{\delta^2 w(x,y)}{\delta y^2} = \frac{P}{D}$, где $D = \frac{Eh^3}{12(1-\nu^2)}$ – изгибая жёсткость, E – модуль упругости, h – толщина пластины, ν – коэффициент Пуассона.

Также, A, B – размеры пластины; R – радиус выреза; P – нагрузка; E – модуль упругости; граничное условие $W = 0$.

Для решения данного дифференциального уравнения аппроксимируем вторые производные как:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2}$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u(x,y+h) - 2u(x,y) + u(x,y-h)}{y^2}$$

Тогда, для каждого внутреннего узла составим разностную схему вида:

$$\frac{W_{i+1,j} - 2W_{i,j} + W_{i-1,j}}{h^2} + \frac{W_{i,j+1} - 2W_{i,j} + W_{i,j-1}}{h^2} = \frac{P}{D}$$

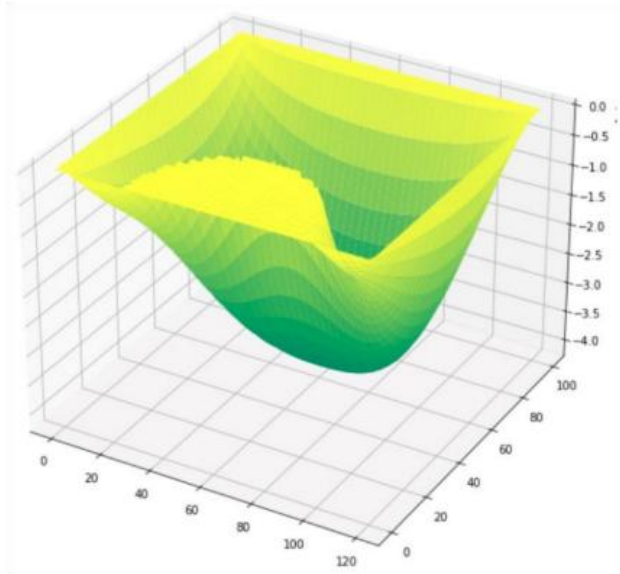
Упростим выражение:

$$W_{i+1,j} + W_{i-1,j} + W_{i,j+1} + W_{i,j-1} - 4W_{i,j} = \frac{P}{D}$$

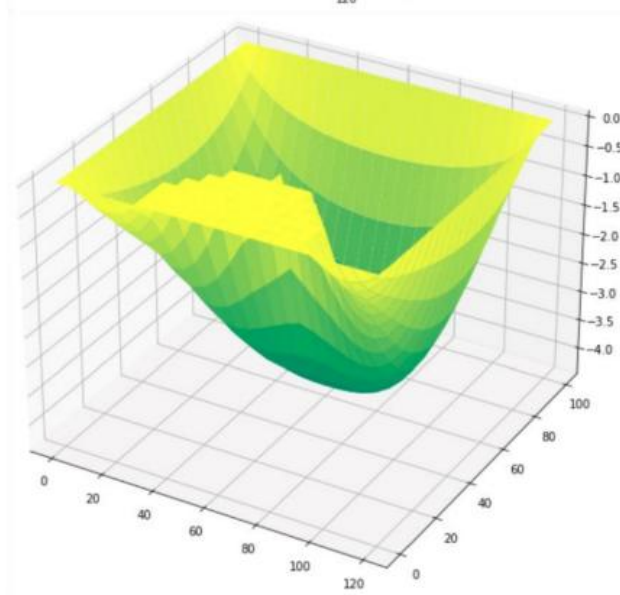
Для остальных (граничных и внешних точек) значение $W_{i,j}=0$.

Данную разностную схему можно решать как систему линейных уравнений. Также будем использовать разреженные матрицы т.к. на одной строке матрицы ненулевыми будут только 5 значений.

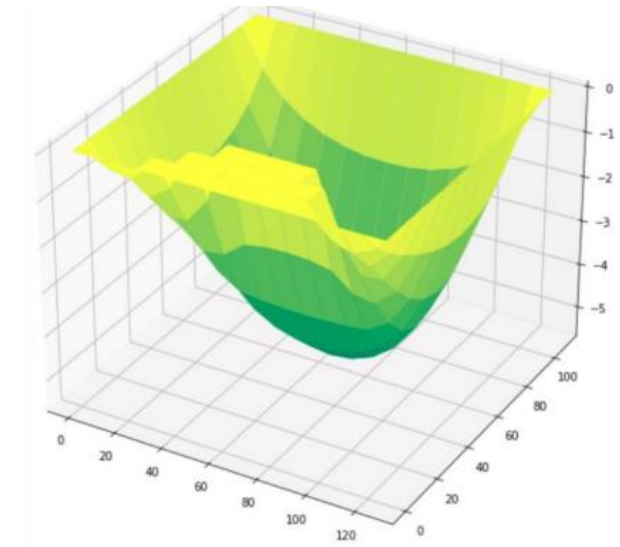
Вычислим решение разностной
схемы с шагом $h=1$.



Вычислим решение разностной
схемы с шагом $h=5$.



Вычислим решение разностной
схемы с шагом $h=9$.



Выводы

В данной лабораторной работе был разработан программный продукт для решения задачи Дирихле для уравнения Пуассона и проверена корректность работы на примере реального физического процесса.

Также, в ходе работы я изучил зависимость качества аппроксимации функции от шага разностной сетки, а также получил визуальное подтверждение гипотезы в виде графика: при увеличении шага качество аппроксимации функции снижается.

Стоит отметить, что для улучшения скорости выполнения программы, использовались разреженные матрицы.