Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»
Кафедра информатики

Отчет по лабораторной работе №5
Метод сеток решения задачи Дирихле для уравнения Пуассона

Выполнил:
студент гр. 953505
Басенко К. А.

Руководитель:
доцент
Анисимов В. Я.

Минск 2021

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

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

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

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

Задача Дирихле (2.48), (2.49) имеет единственное решение $u\ (x,\ y)$. Положим $h = l\,/N$, $x_k = k\,h$, $y_m = m\,h$, $f_{km} = f(x_k, y_m)$. Построим сетки

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

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

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

$\|\upsilon\|_h = \max\limits_{\omega_h} |\upsilon_{km}|$, $\|\upsilon\|'_h = \max\limits_{\omega_h'} |\upsilon_{km}|$.

Разностная схема:

$$\Lambda \upsilon_{km} = f_{km}, \quad k, m = 1, 2, \ldots, N\text{-}1, \tag{2.50}$$

$$\upsilon_{km} = 0 \quad \text{на} \quad \omega_h^*, \tag{2.51}$$

Разностное уравнение (2.50) в более подробной записи имеет вид

$$-\frac{\upsilon_{k-1,m} - 2\upsilon_{km} + \upsilon_{k+1,m}}{h^2} - \frac{\upsilon_{k,m-1} - 2\upsilon_{km} + \upsilon_{k,m+1}}{h^2} = f_{km}. \tag{2.52}$$

Его шаблон изображен на рис. 2.6

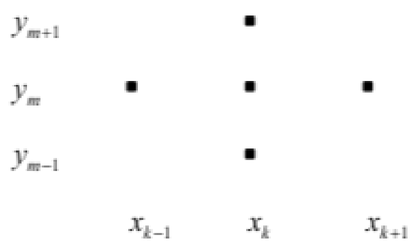


Рис. 2.6

Решение υ разностной задачи Дирихле находится методом последовательных приближений по схеме переменных направлений, где $f_{km}^{\nu-1/2} = f(x_k, y_m)$, υ_{km}^0 – произвольные. Можно доказать, что $\lim\limits_{\nu\to\infty} \upsilon_{km}^{\nu} = \upsilon_{km}$, $k, m =$ 1, 2, . . . , N-1, при любых начальных приближениях υ_{km}^0, причем наибольшая скорость сходимости достигается при $\tau \approx h/\pi$. Здесь положена в основу идея о стабилизации при $t \to +\infty$ 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

$$\Lambda z_{km} = \xi_{km}, \quad k, m = 1, 2, \ldots, N-1,$$

$$z_{km} = 0 \ \text{ на } \ \omega^*$$

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

$$\|z\|_h \leq c\|\xi\|'_h, \tag{2.53}$$

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 $\overline{D}$, а именно, $u(x, y) \in C_4(\overline{D})$. Тогда разностное уравнение (2.51) аппроксимирует дифференциальное уравнение (2.48) на решение задачи (2.48), (2.49) со вторым порядком относительно h, т. е.

$$\|\psi\|'_h = O(h^2), \tag{2.54}$$

где

$$\psi_{km} = \Lambda u_{km} - f_{km}, \tag{2.55}$$

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 υ к решению $u \in C_4(\overline{D})$ задачи (2.48), (2.49) со вторым порядком относительно h, т. е.

$$\|u - \upsilon\|_h = O(h^2). \tag{2.56}$$

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

$$\Lambda r_{km} = \psi_{km}, \qquad k, m = 1, 2, \ldots, N-1,$$

$$r_{km} = 0 \text{ на } \omega_h^*.$$

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

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

где

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

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:

$$\Lambda \upsilon_{km} = f_{km}, \qquad (2.59)$$

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

$$\upsilon_0 = \frac{\rho_1}{\rho_1 + \rho_2} \nu_2 + \frac{\rho_2}{\rho_1 + \rho_2} \varphi_1, \qquad (2.60)$$

где ρ_1 – расстояние от точки 0 до точки 1 на границе Γ, в которой берется заданное значение функции φ, обозначенное через φ_1; υ_0, υ_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.

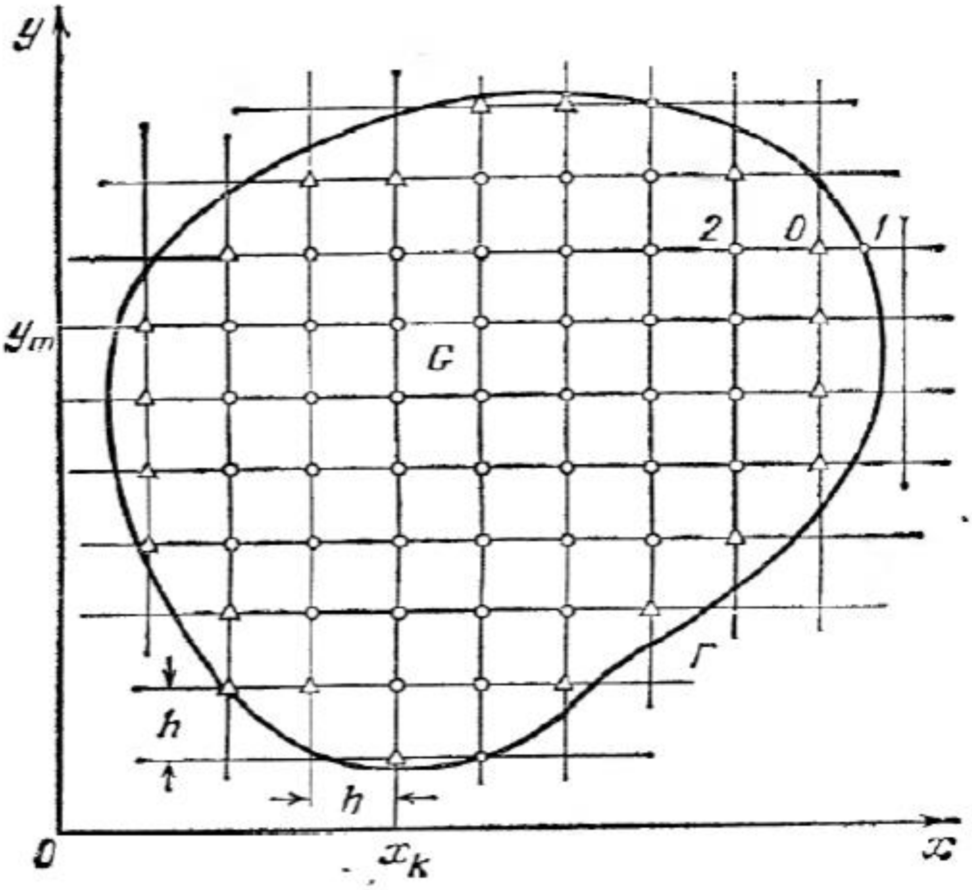


Рис. 2.7.

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 υ_{km}:

$$\upsilon_{km} = \frac{\upsilon_{k-1,m} + \upsilon_{k+1,m} + \upsilon_{k,m-1} + \upsilon_{k,m+1}}{4} + \frac{h^2}{4} f_{km}. \tag{2.61}$$

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

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

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

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

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

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.

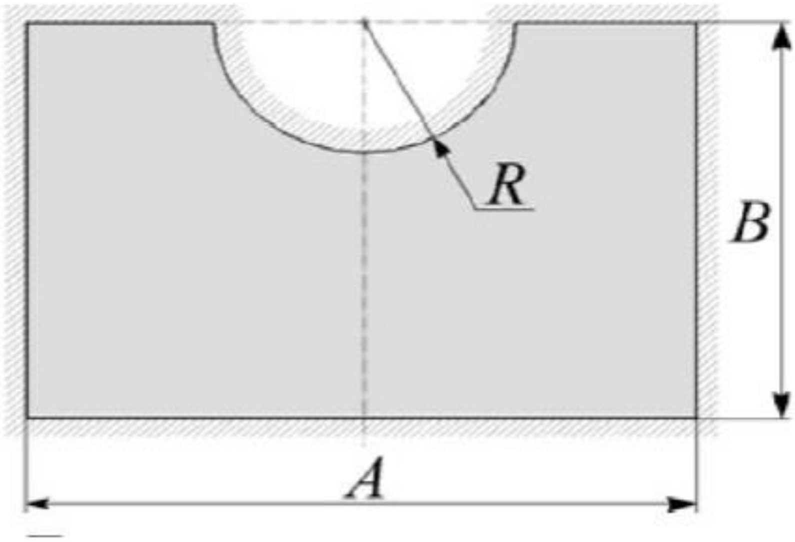


Рис 2.8

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

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

Программная реализация:

Зависимости:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly
```

Нач данные:

```
A = 150
B = 75
R = 15
d = 5
P = 55
E = 70
v = 0.3
```

```
D = 10 * E * d**3 / (12 * (1 - v**2))
P / D
```

```
h = 1
nx = len(np.arange(0, A + h, h))
ny = len(np.arange(0, B + h, h))
```

Записываем:

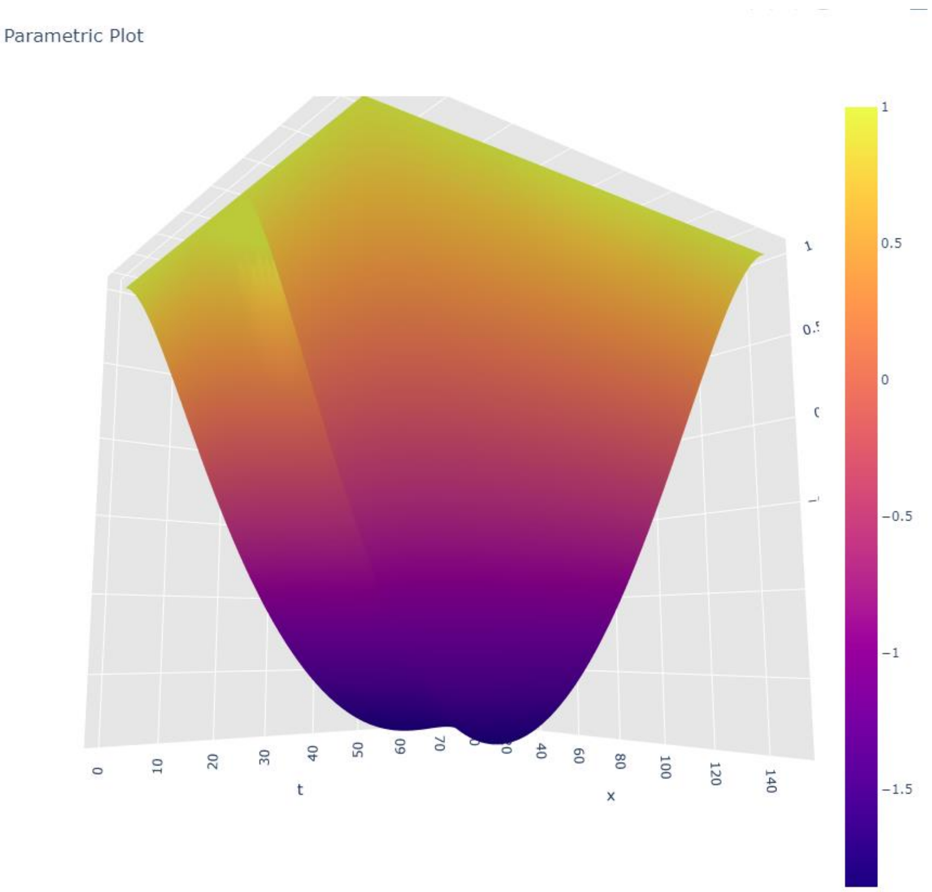
```python
def plot(nx, ny, h, u):
    s = np.arange(0, A + h, h)
    t = np.arange(0, B + h, h)
    tGrid, sGrid = np.meshgrid(t, s)

    surface = go.Surface(x=sGrid, y=tGrid, z=u.T)
    data = [surface]

    layout = go.Layout(
        title='Parametric Plot',
        scene=dict(
            xaxis=dict(
                gridcolor='rgb(255, 255, 255)',
                showbackground=True,
                backgroundcolor='rgb(230, 230,230)'
            ),
            yaxis=dict(
                title='t',
                gridcolor='rgb(255, 255, 255)',
                showbackground=True,
                backgroundcolor='rgb(230, 230,230)'
            ),
            zaxis=dict(
                title='u(x, t)',
                gridcolor='rgb(255, 255, 255)',
                showbackground=True,
                backgroundcolor='rgb(230, 230,230)'
            )
        )
    )

    fig = go.Figure(data=data, layout=layout)
    plotly.offline.plot(fig, auto_open=True)
```

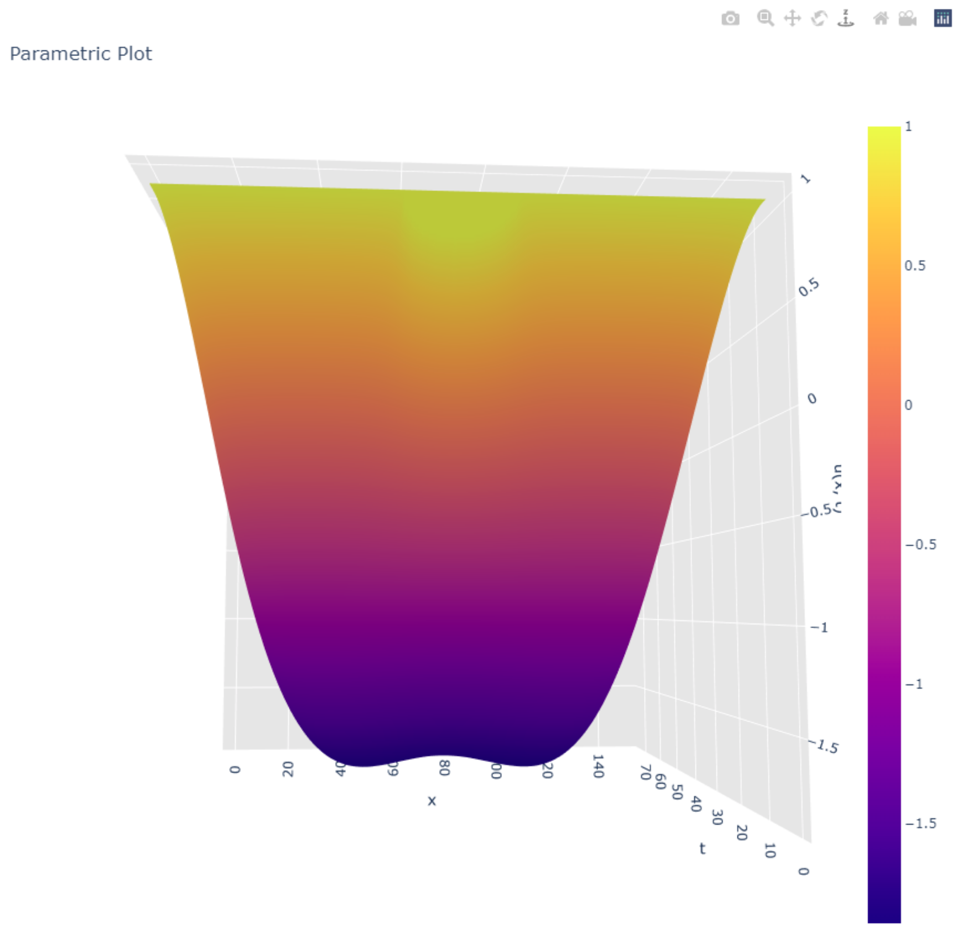
Дописываем:

```python
def get_u(eps):
    cur_error = 1

    iters = 0
    while cur_error > eps:
        cur_error = 0
        iters += 1
        for i in range(1, ny - 1):
            for j in range(1, nx - 1):
                if np.sqrt((A / 2 - h * j)**2 + (i * h)**2) <= R:
                    u[i][j] = 1
                else:
                    prev = u[i][j]
                    u[i][j] = (u[i, j + 1] + u[i + 1, j] + u[i - 1, j] + u[i,
                    error = abs(prev - u[i][j])
                    cur_error = max(error, cur_error)
        # print(cur_error)

    return iters, u
```

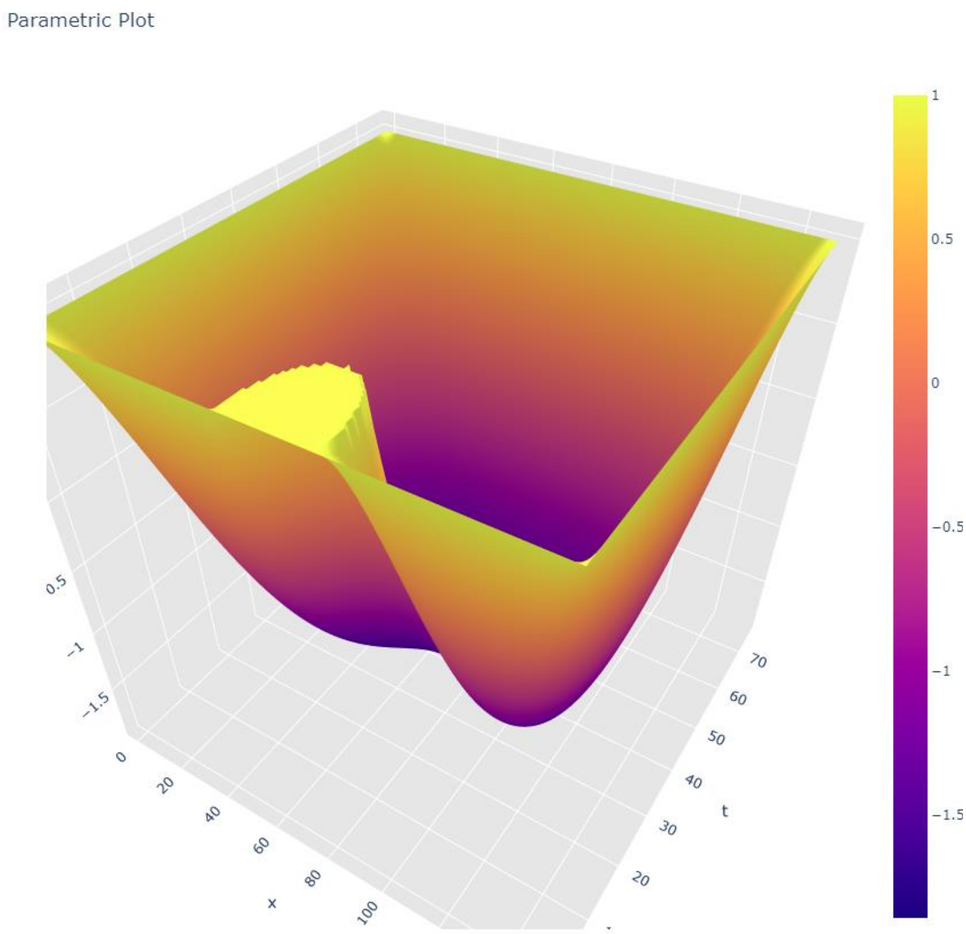
Фиксируем

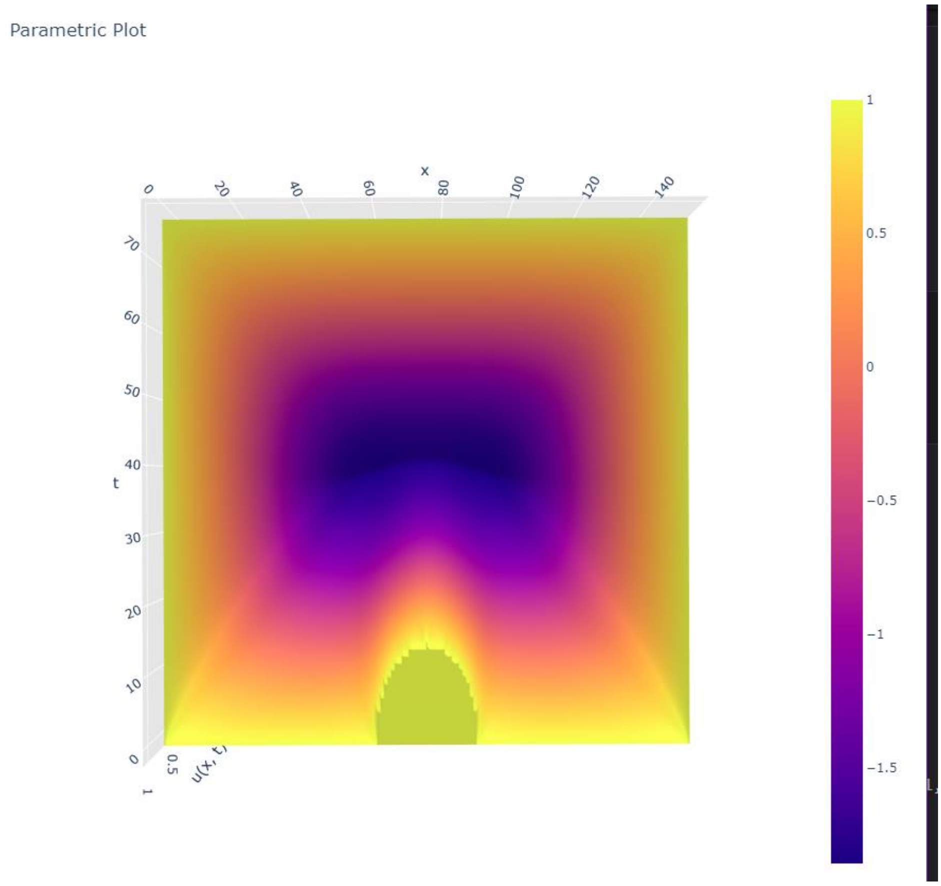
```
iters, u = get_u(0.0001)
plot(nx, ny, h, u)
```





Parametric Plot
x
t
u(x, t)


Parametric Plot
x
t
1
0.5
0
−0.5
−1
−1.5


Parametric Plot
x
t
u(x, t)
1
0.5
0
−0.5
−1
−1.5

Наибольшие прогибы находятся в центре, по бокам от выреза.

Заключение:

В результате данной работы была написана программная реализация решения задачи Дирихле.