Московский государственный университет имени М.В.Ломоносова

Казахстанский филиал

Факультет вычислительной математики и кибернетики

Отчёт по практикуму по специализации

Численное решение задачи Дирихле для уравнения Пуассона

Составил: студент Калдаров Б.М.

Проверил: преподаватель Нетесов В.В.

Нур-Султан, 2021

Содержание

1 Постановка задачи

Рассматривается задача Дирихле для эллиптического уравнения

$$-Lu = f(x,y), \quad (x,y) \in G, \tag{1}$$

$$u = \mu(x,y), \quad (x,y) \in H. \tag{2}$$

Пусть $\overline{G} = G + H = 0 \leq x \leq l_x, 0 \leq y \leq l_y$ - прямоугольник, а

$$Lu = \frac{\partial}{\partial x}\left(p(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right) \tag{3}$$

Здесь $p(x,y), q(x,y)$ - достаточно гладкие функции такие, что $0 < c_1 \leq p(x,y) \leq c_2, 0 < d_1 \leq q(x,y) \leq d_2$, где c_1, c_2, d_1, d_2 - постоянные. Обозначим $A = max(c_2, d_2)$. Разобьём отрезок $[0, l_x]$ на N_x равных частей. Обозначим $h_x = \frac{l_x}{N_x}, x_i = ih_x, 0 \leq i \leq N_x$.
Разобьём отрезок $[0, l_y]$ на N_y равных частей. Обозначим $h_y = \frac{l_y}{N_y}, y_j = jh_y, 0 \leq j \leq N_y$.

2 Основные результаты

2.1 Выбор функций и параметров

В качестве пробной функции взята функция

$$\mu(x, y) = x^2y + y^2x$$

Правая часть

$$f(x, y) = xy$$

Норма $||A|| = max|a_{ij}|$, а точность $eps = 1e - 2$.

Количество итераций считалось по формулам

Для метода простой итерации:

$$m \geq 2\frac{ln(1/eps)}{(\pi h)^2}$$

Для метода Зейделя:

$$m \geq \frac{ln(1/eps)}{(\pi h)^2}$$

Для метода верхней релаксации:

$$m \geq 2\frac{ln(1/eps)}{\pi h}$$

2.2 Метод простой итерации с оптимальным параметром

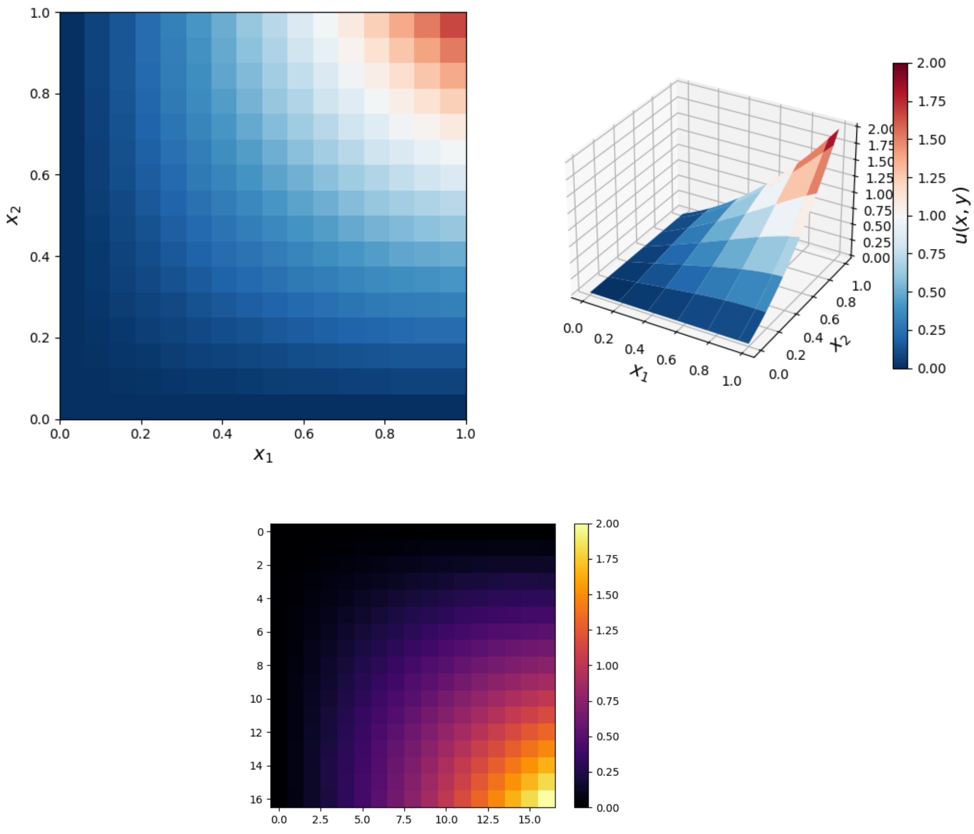

График для $n = 16, m_{opt} = 239$.

N	$\|\|u^k - u^*\|\|$	$\frac{\|\|u^k - u^*\|\|}{\|\|u^0 - u^*\|\|}$	$\|\|u^k - u^{k-1}\|\|$
1	1.647	0.823	2.0
10	0.712	0.356	0.0583
30	0.26	0.13	0.013
m_{opt}	0.025	0.0126	0.004

2.3 Метод Зейделя

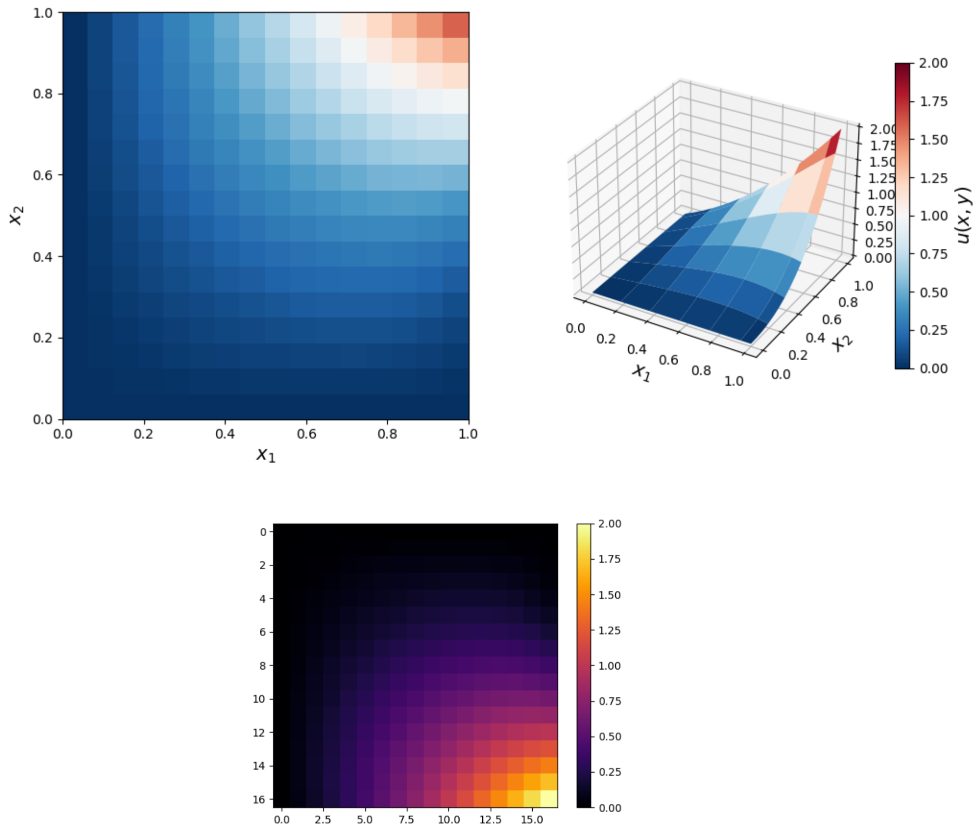

График для $n = 16, m_{opt} = 120.$

N	$\|\|u^k - u^*\|\|$	$\frac{\|\|u^k - u^*\|\|}{\|\|u^0 - u^*\|\|}$	$\|\|u^k - u^{k-1}\|\|$
1	1.544	0.77	2.0
10	0.340	0.17	0.049
30	0.091	0.045	0.009
m_{opt}	0.238	0.119	0.0002

2.4 Метода верхней релаксации

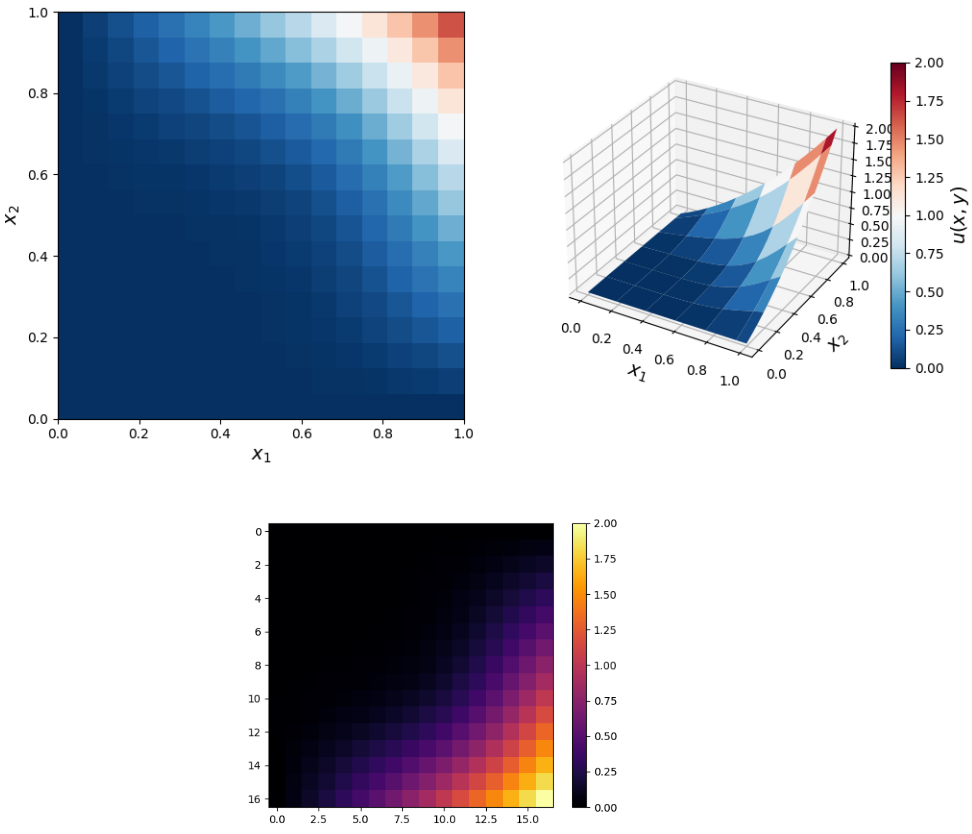

График для $n = 16, m_{opt} = 47$.

N	$\|\|u^k - u^*\|\|$	$\frac{\|\|u^k - u^*\|\|}{\|\|u^0 - u^*\|\|}$	$\|\|u^k - u^{k-1}\|\|$
1	1.647	0.823	2.0
10	0.861	0.43	0.057
30	0.438	0.219	0.015
m_{opt}	0.272	0.136	0.008

3 Листинг программы

3.1 Общий код

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr  4 17:55:38 2021

@author: Kaldarov
"""

import numpy as np
import time

import matplotlib as mpl
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.axes3d import Axes3D

def norm(a):
        return np.max(np.abs(a))

def mu(x, y):
        return x*x*y + x*y*y

def f(x, y):
        return x * y

N = 16
N1 = N+1
```

```
L = 1.0
h = L/N
x = np.linspace(0.0, L, N1)
y = np.linspace(0.0, L, N1)

ua = np.zeros((N1,N1))
for i in range(N1):
        for j in range(N1):
                ua[i, j] = mu(x[i], y[j])

u = np.zeros((N1, N1))

u0 = u.copy()
u_old = u.copy()

delta = 8/(h*h) * np.sin(np.pi*h/2)**2
delta2 = 8/(h*h) * np.cos(np.pi*h/2)**2
xi = delta/delta2

roH = (1-xi)/(1+xi)
print('roH = ', roH)

err = norm(u_old-ua)
eps = 1e-2
```

3.2 Метод простой итерации с оптимальным параметром

```
m = int(2*(np.log(1/eps))/(np.pi*h)**2+1)
while m > 0:
        for i in range(N1):
                u_old[i, 0] = mu(x[i], 0.0)
                u_old[i, N] = mu(x[i], 1.0)
                u_old[0, i] = mu(0.0, y[i])
                u_old[N, i] = mu(1.0, y[i])

        for i in range(1, N):
                for j in range(1, N):
                        u[i, j] = (u_old[i-1, j] + u_old[i+1, j] +
                        u_old[i, j-1] + u_old[i, j+1] +
                        h*h*f(x[i], y[j])) / 4


        u, u_old = u_old, u
        err = norm(u - ua)
        print(err, err / norm(u0-ua), norm(u_old-u))
        m -= 1
```

3.3 Метод Зейделя

```
m = int((np.log(1/eps))/(np.pi*h)**2+1)
while m > 0:
        for i in range(N1):
                u_old[i, 0] = mu(x[i], 0.0)
                u_old[i, N] = mu(x[i], 1.0)
                u_old[0, i] = mu(0.0, y[i])
                u_old[N, i] = mu(1.0, y[i])



        for i in range(1, N):
                for j in range(1, N):
                        u[i, j] = (u[i-1, j] + u_old[i+1, j] +
                        u[i, j-1] + u_old[i, j+1] +
                        h*h*f(x[i], y[i])) / 4


        u, u_old = u_old, u
        err = norm(u - ua)
        m -= 1
        print(err, err / norm(u0-ua), norm(u_old-u))
```

3.4 Метода верхней релаксации

```
m = int(2*(np.log(1/eps))/(np.pi*h)+1)
while m > 0:
        for i in range(N1):
                u_old[i, 0] = mu(x[i], 0.0)
                u_old[i, N] = mu(x[i], 1.0)
                u_old[0, i] = mu(0.0, y[i])
                u_old[N, i] = mu(1.0, y[i])

        for i in range(1, N):
                for j in range(1, N):
                        u[i, j] = u_old[i, j] + w * (u[i-1, j] +
                        u_old[i+1, j] + u[i, j-1] + u_old[i, j+1] +
                        h*h*f(x[i], y[i]) - 4 * u_old[i, j]) / 4

        u, u_old = u_old, u
        err = norm(u - ua)
        m -= 1
        print(err, err / norm(u0-ua), norm(u_old-u))
```

4 Список литературы

1. **Пакулина А.Н.** Практикум по методам вычислений. Часть 2. СПб., СПбГУ, 2019. – 113 с.

2. **Самарский А.А, Гулин А.В.** Численные методы математической физики. Наука, 2000. - 310 с.