Московский авиационный институт

(Национальный исследовательский университет)

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра вычислительной математики и программирования

Лабораторная работа №7

по курсу «Численные методы»

Студент: Мариечев К.Д.

Группа: М8О-409Б-20

Проверил: Пивоваров Д.Е.

Дата:

Оценка:

Москва, 2023

Задание

Решить краевую задачу для дифференциального уравнения эллиптического типа. Аппроксимацию уравнения произвести с использованием центрально-разностной схемы. Для решения дискретного аналога применить следующие методы: метод простых итераций (метод Либмана), метод Зейделя, метод простых итераций с верхней релаксацией. Вычислить погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением $U(x, y)$. Исследовать зависимость погрешности от сеточных параметров h_x, h_y.

Вариант 9:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2\frac{\partial u}{\partial y} - 3u,$$

$$u(0, y) = \exp(-y)\cos y,$$

$$u(\frac{\pi}{2}, y) = 0,$$

$$u(x,0) = \cos x,$$

$$u(x, \frac{\pi}{2}) = 0.$$

Аналитическое решение: $U(x, y) = \exp(-y)\cos x \cos y$.

Теория

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2\frac{\partial u}{\partial y} - 3u$$

$$u(0, y) = \varphi_1(y)$$

$$u(\pi/2, y) = \varphi_2(y)$$

$$u(x, 0) = \psi_1(x)$$

$$u(x, \pi/2) = \psi_2(x)$$

Пусть M разбиений по x, N разбиений по y. Тогда $h_x=l_x/M$ $h_y=l_y/N$, $x_i= l_x + i\, h_x$, $y_j= l_y + j\, h_y$

Во внутренних узлах конечно-разностной сетки аппроксимация происходит по следующей схеме:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} = -2\frac{u_{i+1,j} - u_{i-1,j}}{2h_x} - 3u_{i,j}$$

Выразим u_{ij}

$$u_{i,j} = \frac{h_x^2(u_{i+1,j} + u_{i-1,j}) + h_y^2\left(u_{i,j+1} + u_{i,j-1}\right) + h_y^2 h_x(u_{i+1,j} + u_{i-1,j})}{4h_y^2 - 3h_y^2 h_x^2}$$

Метод Либмана

$$u_{i,j}^{k+1} = \frac{h_x^2\left(u_{i+1,j}^k + u_{i-1,j}^k\right) + h_y^2\left(u_{i,j+1}^k + u_{i,j-1}^k\right) + h_y^2 h_x(u_{i+1,j}^k + u_{i-1,j}^k)}{4h_y^2 - 3h_y^2 h_x^2}$$

Метод Зейделя

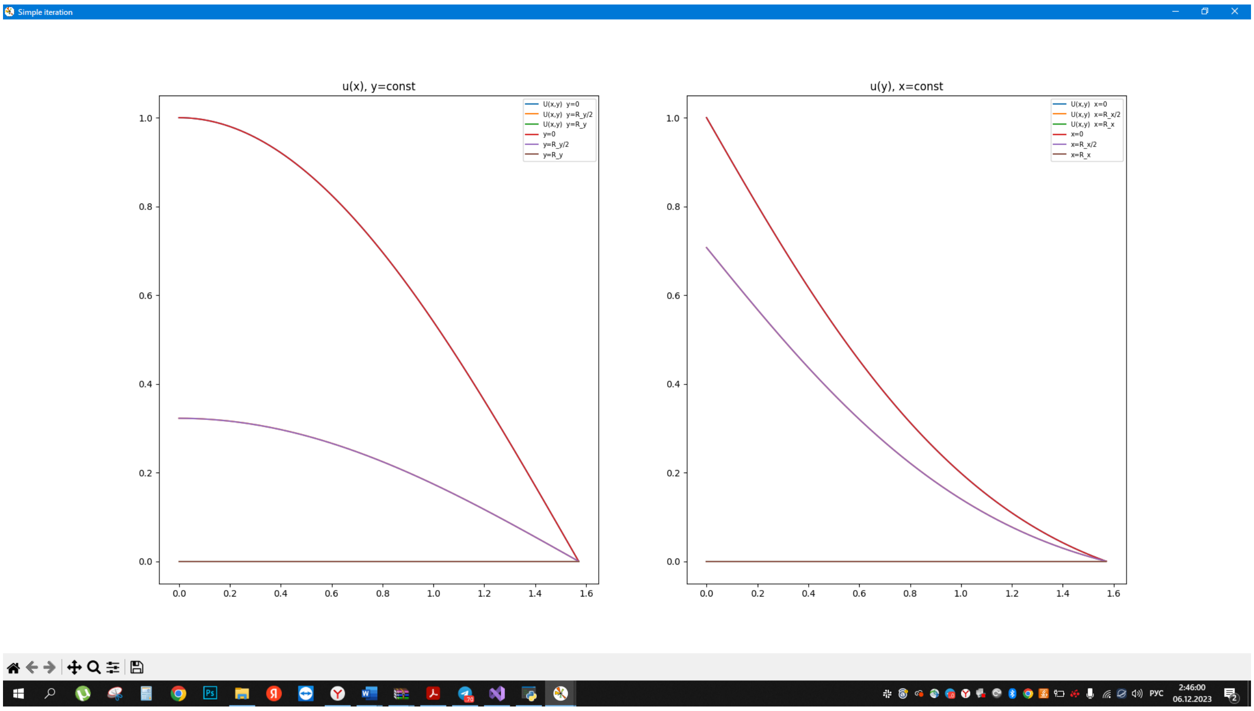
$$u_{i,j}^{k+1} = \frac{h_x^2\left(u_{i+1,j}^k + u_{i-1,j}^{k+1}\right) + h_y^2\left(u_{i,j+1}^k + u_{i,j-1}^{k+1}\right) + h_y^2 h_x(u_{i+1,j}^k + u_{i-1,j}^{k+1})}{4h_y^2 - 3h_y^2 h_x^2}$$

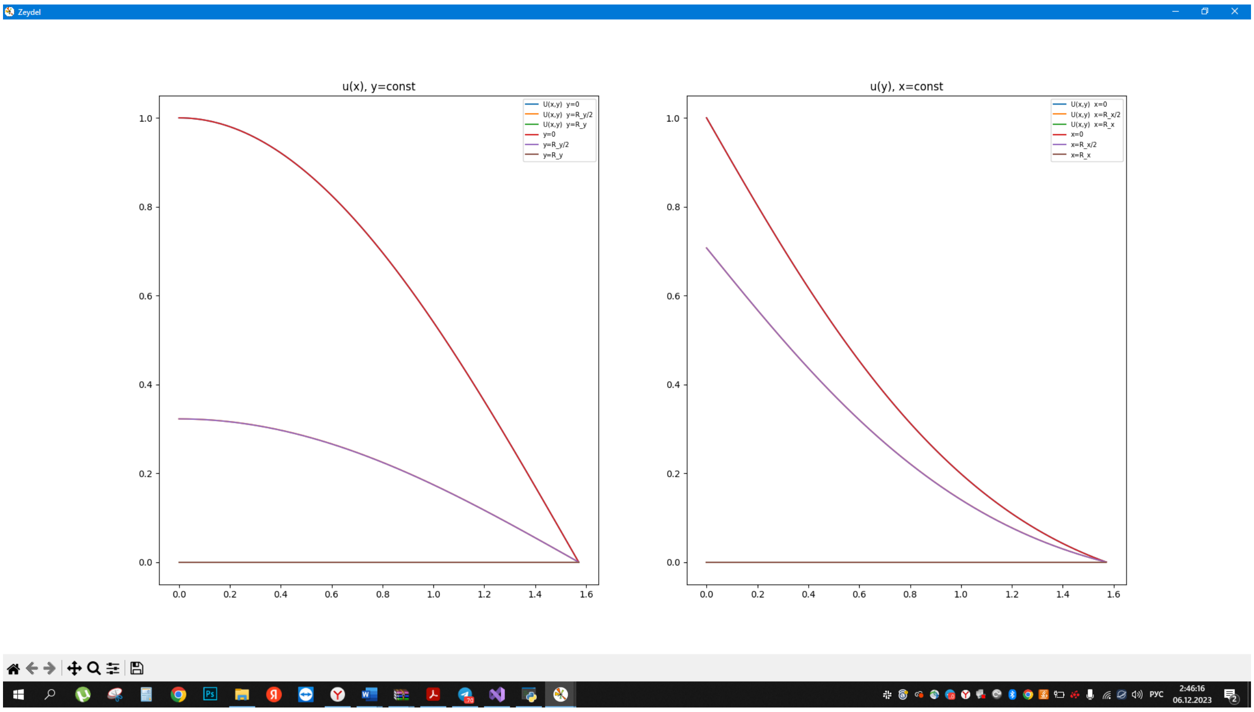
Для нахождения значение на 0-ой итерации применяем линейную интерполяцию

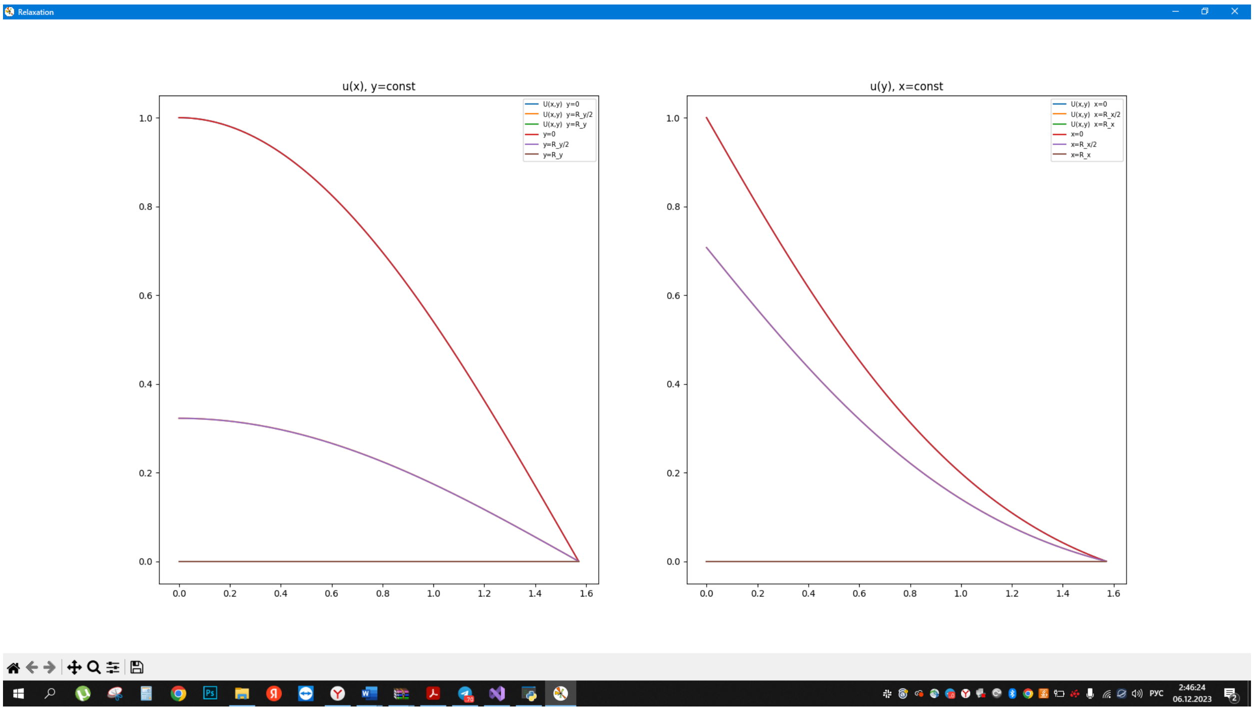
Метод Зейделя с верхней релаксацией

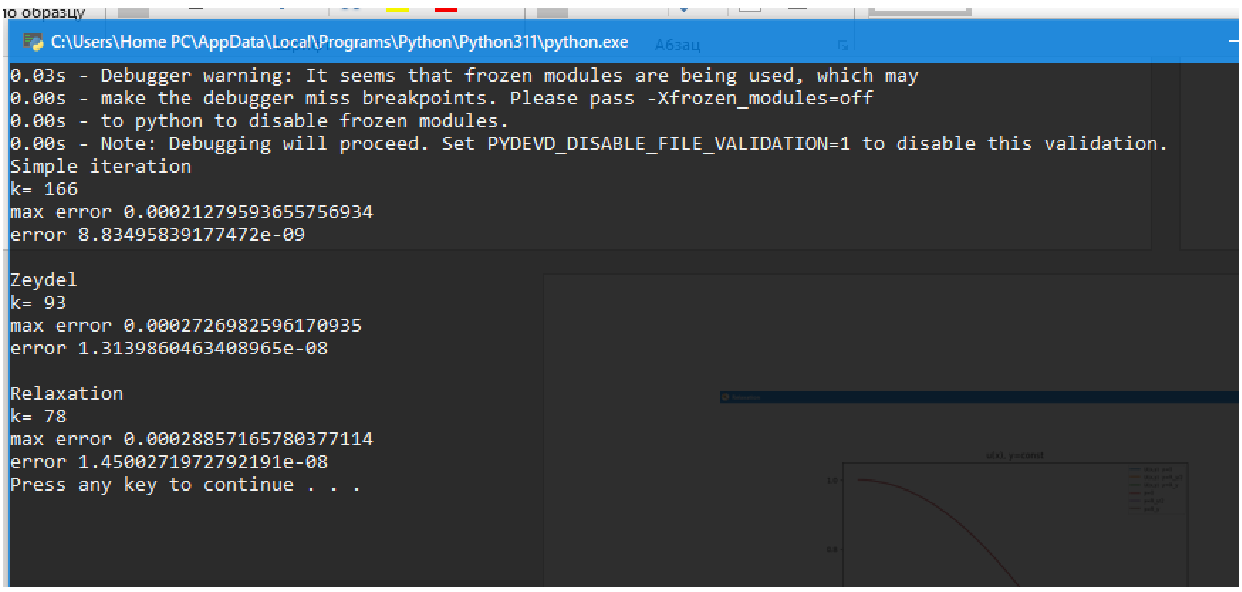
$$u_{i,j}^{k+1} = \text{c}\frac{h_x^2\left(u_{i+1,j}^k + u_{i-1,j}^{k+1}\right) + h_y^2\left(u_{i,j+1}^k + u_{i,j-1}^{k+1}\right) + h_y^2 h_x(u_{i+1,j}^k + u_{i-1,j}^{k+1})}{4h_y^2 - 3h_y^2 h_x^2} + (1 - \text{c})\, u_{i,j}^k$$

Решение


Simple iteration
u(x), y=const
U(x,y) y=0
U(x,y) y=R_y/2
U(x,y) y=R_y
y=0
y=R_y/2
y=R_y
u(y), x=const
U(x,y) x=0
U(x,y) x=R_x/2
U(x,y) x=R_x
x=0
x=R_x/2
x=R_x


Zeydel
u(x), y=const
U(x,y) y=0
U(x,y) y=R_y/2
U(x,y) y=R_y
y=0
y=R_y/2
y=R_y
u(y), x=const
U(x,y) x=0
U(x,y) x=R_x/2
U(x,y) x=R_x
x=0
x=R_x/2
x=R_x





```
0.03s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
Simple iteration
k= 166
max error 0.00021279593655756934
error 8.83495839177472e-09

Zeydel
k= 93
max error 0.0002726982596170935
error 1.3139860463408965e-08

Relaxation
k= 78
max error 0.00028857165780377114
error 1.4500271972792191e-08
Press any key to continue . . .
```

Фрагменты кода

```
for i in range(0,N+1):
    x=L_x+i*h_x
    u[i][0]=psi1(x)
    u[i][M]=psi2(x)


for j in range(0,M+1):
    y=L_y+j*h_y
    u[0][j]=phi1(y)
    u[N][j]=phi2(y)



for i in range(1,N):
    for j in range(1,M):
        x=L_x+i*h_x
        y=L_y+j*h_y
```

```
        Cx=phi1(x)*(R_x-x)/(R_x-L_x)+phi2(x)*(x-L_x)/(R_x-L_x)
        Cy=psi1(x)*(R_y-y)/(R_y-L_y)+psi2(x)*(y-L_y)/(R_y-L_y)
        u[i][j]=(Cy+Cx)/2

u_new=np.array(u)

error1=1




def solve(u,f):
    u_new=np.array(u)
    k=0
    error1=1
    while epsilon<error1:

        k+=1
        u_old=np.array(u_new)


        for i in range(1,N):
            if f==1:
                for j in range(1,M):
                    u_new[i][j]=(h_y**2*(u_old[i+1][j]+u_old[i-
1][j])+h_x**2*(u_old[i][j-1]+u_old[i][j+1])+h_x**2*h_y*(u_old[i][j+1]-u_old[i][j-
1]))/(2*h_y**2+2*h_y**2-3*h_x**2*h_y**2)
            elif f==2:
                for j in range(1,M):
                    u_new[i][j]=(h_y**2*(u_old[i+1][j]+u_new[i-
1][j])+h_x**2*(u_new[i][j-1]+u_old[i][j+1])+h_x**2*h_y*(u_old[i][j+1]-u_new[i][j-
1]))/(2*h_y**2+2*h_y**2-3*h_x**2*h_y**2)
            elif f==3:
                for j in range(1,M):
                    uk=(h_y**2*(u_old[i+1][j]+u_new[i-1][j])+h_x**2*(u_new[i][j-
1]+u_old[i][j+1])+h_x**2*h_y*(u_old[i][j+1]-u_new[i][j-1]))/(2*h_y**2+2*h_y**2-
3*h_x**2*h_y**2)
                    u_new[i][j]=c*uk+(1-c)*u_old[i][j]

        error1=error(u_new,u_old)

    error2=error(u_new,U)
    print("k=",k)
    print("max error",error2)
    error4=error3(u_new,U)
    print("error",error4)
    return u_new
```

Вывод

В данной работе была решена краевая задача для ДУ эллиптического типа с помощью конечно-разностной схемы.

После применения конечно-разностной схемы мы получаем систему уравнений, которую можно решать несколькими методами:

- метод простых итераций
- метод Зейделя

- метод верхних релаксаций

С помощью каждого из этих методов получилось решение, отличающееся от аналитического на ~1e-8. Метод релаксаций оказался наиболее точным и потребовал меньшее количество итераций