Московский Авиационный Институт

(национальный исследовательский университет)

Институт №8 “Информационные технологии и прикладная математика”

Кафедра 806 “ Вычислительная математика и программирование”

Лабораторная работа № 7 по курсу “Численные методы”

на тему

“ Численные методы решения ДУЧП эллиптического типа”

Студент: Четвергов А. О.

Группа: М8О-409Б-19

Вариант: 3

Преподаватель: Пивоваров Д. Е.

Москва

2022

1. 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.

2. Вариант

3.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,$$

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

$$u(1, y) = e \cos y,$$

$$u_y(x, 0) = 0,$$

$$u_y(x, \frac{\pi}{2}) = -\exp(x).$$

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

3. Теория

Дана краевая задача для дифференциального уравнения эллиптического типа:

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x, y), \quad x \in (0, l_1),\ y \in (0, l_2); \\ \alpha_1 \dfrac{\partial u}{\partial x}(0, y) + \beta_1 u(0, y) = \varphi_1(y), \\ \alpha_2 \dfrac{\partial u}{\partial x}(l_1, y) + \beta_2 u(l_1, y) = \varphi_2(y), \\ \alpha_3 \dfrac{\partial u}{\partial x}(x, 0) + \beta_3 u(x, 0) = \varphi_3(x), \\ \alpha_4 \dfrac{\partial u}{\partial x}(x, l_2) + \beta_4 u(x, l_2) = \varphi_4(x) \end{cases}$$

Сетка:

$$\omega_{h_1, h_2} = \{x_i = ih_1, i = \overline{0, N_1}; \quad y_j = jh_2, j = \overline{0, N_2}\}.$$

На этой сетке аппроксимируем дифференциальную задачу во внутренних узлах с помощью отношения конечных разностей по следующей схеме:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_1^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_2^2} + O\left(h_1^2 + h_2^2\right) = f(x_i, y_j), \tag{5.47}$$

$$i = \overline{1, N_1 - 1}, \quad j = \overline{1, N_2 - 1}$$

Граничные условия аппроксимируем с первым порядком с помощью направленных разностей:

$$\alpha_1 \frac{u_{1j} - u_{0j}}{h_1} + \beta_1 u_{0j} = \varphi_1(y_j),\ j = \overline{1, N_2 - 1},$$

$$\alpha_2 \frac{u_{N_1 j} - u_{N_1 j}}{h_1} + \beta_2 u_{N_1 j} = \varphi_2(y_j),\quad j = \overline{1, N_2 - 1},$$

$$\alpha_3 \frac{u_{i1} - u_{i0}}{h_2} + \beta_3 u_{i0} = \varphi_3(x_i),\ i = \overline{1, N_1 - 1}$$

$$\alpha_4 \frac{u_{iN_2} - u_{iN_2 - 1}}{h_2} + \beta_4 u_{iN_2} = \varphi_4(x_i),\ i = \overline{1, N_1 - 1}.$$

СЛАУ (5.47) имеет пяти-диагональный вид (каждое уравнение содержит пять неизвестных и при соответствующей нумерации переменных матрица имеет ленточную структуру).

Рассмотрим разностно-итерационный метод Либмана численного решения задачи Дирихле. Из схемы (5.47) получим (k-номер итерации)

$$u_{i,j}^{(k+1)} = \frac{1}{4}[u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)} - h^2 \cdot f_{i,j}],\quad f_{i,j} = f(x_i, y_j),$$
$$i = \overline{1, N_1 - 1},\quad j = \overline{1, N_2 - 1}. \qquad (5.48)$$

На каждой координатной линии с помощью линейной интерполяции граничных значений $\varphi(x, y)$ определим $u_{i,j}^{(0)}$ на нулевой итерации, подставив которые в (5.48), получим распределение $u_{i,j}^{(1)}$ на первой итерации

$$u_{i,j}^{(1)} = \frac{1}{4}[u_{i+1,j}^{(0)} + u_{i-1,j}^{(0)} + u_{i,j+1}^{(0)} + u_{i,j-1}^{(0)} - h^2 \cdot f_{i,j}.$$

Это распределение $u_{i,j}^{(1)}$ снова подставляется в (5.48), получаем распределение $u_{i,j}^{(2)}$ и т.д. Процесс Либмана прекращается, когда

$$\left\| u^{(k+1)} - u^{(k)} \right\| \le \varepsilon,\qquad \left\| u^{(k)} \right\| = \max_{i,j} \left| u_{i,j}^{(k)} \right|,$$

Метод верхней релаксации:

После вычисления очередной i-й компоненты (k+1)-го приближения по формуле метода Зейделя

$$\tilde{x}_i^{(k+1)} = b_{i1} x_1^{(k+1)} + b_{i2} x_2^{(k+1)} + \ldots + b_{i,i-1} x_{i-1}^{(k+1)} + b_{i,i+1} x_{i+1}^{(k)} +$$
$$+ \ldots + b_{im} x_m^{(k)} + c_i$$

Производят дополнительное смещение этой компоненты на величину $(\omega - 1)(\tilde{x}_i^{(k+1)} - x_i^{(k)})$

Где ω – параметр релаксации. Таким образом, i-я компонента (k+1)-го приближения вычисляется по формуле

$$x_i^{(k+1)} = \tilde{x}_i^{(k+1)} + (\omega - 1)(\tilde{x}_i^{(k+1)} - x_i^{(k)}) = \omega\tilde{x}_i^{(k+1)} + (1 - \omega)x_i^{(k)}.$$

При $\omega = 1$ метод релаксации совпадает с методом Зейделя. При $\omega > 1$ верхняя релаксация, а при $\omega < 1$ нижняя релаксация.

4. Программа

Параметры:

```
lx = 1
ly = np.pi/2
nx = 10
ny = 10
hx = lx/nx
hy = ly/ny
eps = 10**(-3)
```

Дано:

```
def phi1(y):
    return np.cos(y)

def phi2(y):
    return np.cos(y)*np.exp(1)

def phi3(x):
    return 0

def phi4(x):
    return -np.exp(x)

def exact(x,y):
    return np.exp(x) * np.cos(y)
```

Сетка:

```
x = np.arange(0, lx+hx, hx)
y = np.arange(0, ly+hy, hy)
```

Получаем u0:

```
def u_zero():
    u = np.zeros(shape=(len(x), len(y)))
    for j in range(0,len(y)):
        u[0][j] = phi1(y[j])
        u[-1][j] = phi2(y[j])
    for j in range(len(y) - 1, -1, -1):
        for i in range(len(x) - 2, 0, -1):
            u[i, j] = u[i + 1, j] * y[i] / (x[j] + y[i])
    for j in range(len(y) - 1, -1, -1):
        for i in range(1,len(x)-1):
            u[i, j] += u[i + 1, j] * y[i] / (x[j] + y[i])
            u[i, j] /= 2
    return u
```

Метод Либмана:

```
def Lib_method():
    u = u_zero()
    v = np.zeros(shape=(len(x), len(y)))
    k = 0
    m = np.abs(u-v).max()
    while m > eps:
        v = copy.deepcopy(u)
        for i in range(1, len(x)-1):
            u[i][0] = v[i][1]-phi3(x[i])*hy
            u[i][-1] = v[i][-2]+phi4(x[i])*hy
        for i in range(1, len(x)-1):
            for j in range(1, len(y)-1):
                u[i,j] = (v[i+1][j] + v[i-1][j] + v[i][j-1] + v[i][j+1]) / 4
        m = np.abs(u-v).max()
        k += 1
    print("Метод Либмана")
    print("Шагов: ",k)
    print("Погрешность: ",m)
    return u
```

Метод верхней релаксации:

```
def relax(w):
    u = u_zero()
    v = np.zeros(shape=(len(x), len(y)))
    k = 0
    m = np.abs(u-v).max()
    while m > eps:
        v = copy.deepcopy(u)
        for i in range(1, len(x)-1):
            u[i][0] = v[i][1]-phi3(x[i])*hy
            u[i][-1] = v[i][-2]+phi4(x[i])*hy
        for i in range(1, len(x)-1):
            for j in range(1, len(y)-1):
                u[i,j] += w * ((u[i-1][j] + v[i+1][j] + u[i][j-1] +
v[i][j+1]) / 4 - v[i,j])
        m = np.abs(u-v).max()
        k += 1
    if (w == 1):
        print("Метод Зейделя")
    else:
        print("Метод верхней релаксации")
    print("Шагов: ",k)
    print("Погрешность: ",m)
    return u
```

Аналитическое решение:

```
def analytic(x, y):
    ext = np.zeros((len(x), len(y)))
    for i in range(0, len(x)):
        for j in range(0, len(y)):
            ext[i, j] = exact(x[i], y[j])
    return ext
```

График:

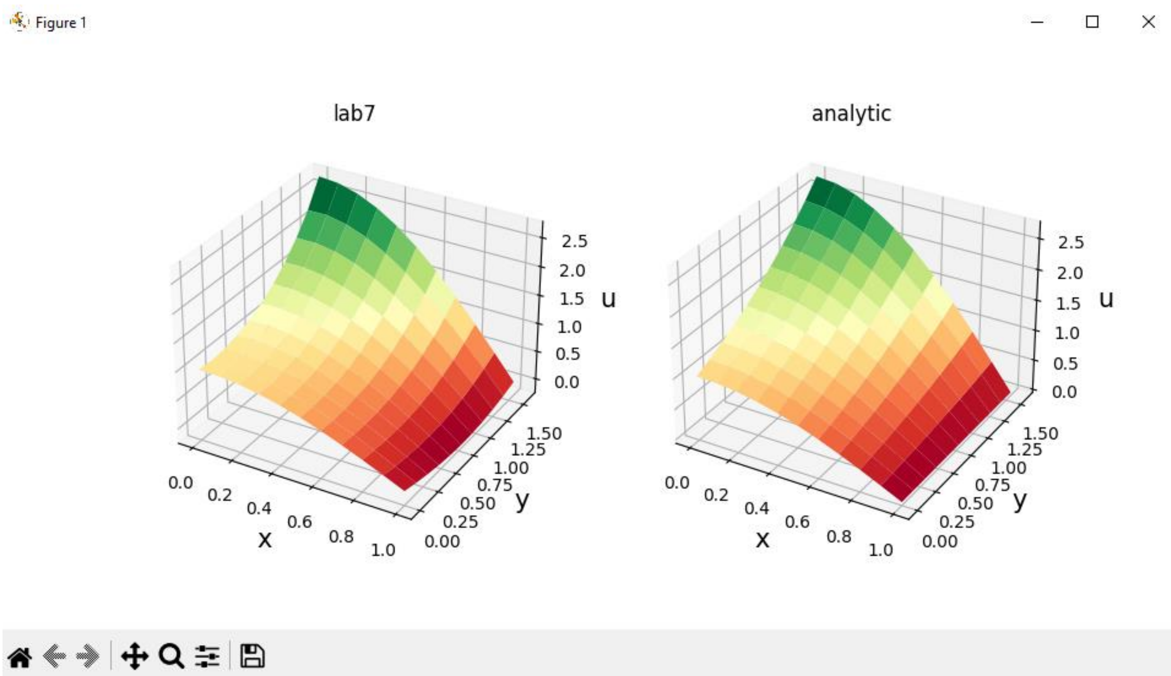
```
def plot(mtd, analytic, x, y):
    fig = plt.figure(figsize=plt.figaspect(0.5))
    x1, y1 = np.meshgrid(x, y)
    pnt = fig.add_subplot(1, 2, 1, projection='3d')
    plt.title('lab7')
    pnt.set_xlabel('x', fontsize=15)
    pnt.set_ylabel('y', fontsize=15)
    pnt.set_zlabel('u', fontsize=15)
    pnt.plot_surface(x1, y1, mtd, cmap=cm.RdYlGn)
    pnt = fig.add_subplot(1, 2, 2, projection='3d')
    plt.title('analytic')
    pnt.set_xlabel('x', fontsize=15)
    pnt.set_ylabel('y', fontsize=15)
    pnt.set_zlabel('u', fontsize=15)
    pnt.plot_surface(x1, y1, analytic, cmap=cm.RdYlGn)

    plt.show()
```

Main:

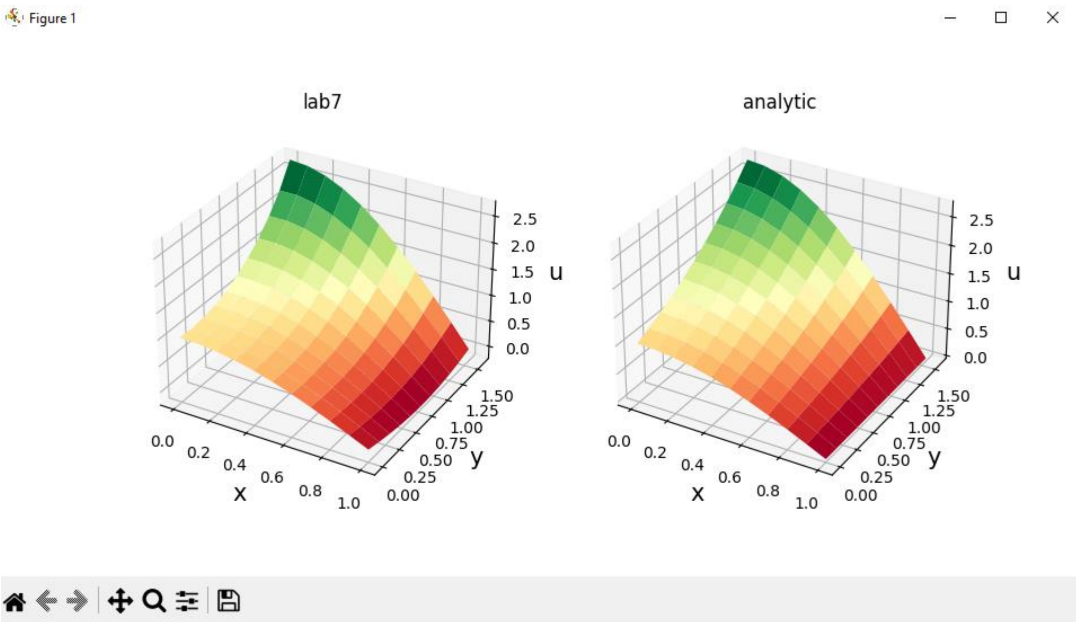
```
analytic = analytic(x, y)

lm = Lib_method() # Метод Либмана
plot(lm,analytic, x, y)

zeyd = relax(1) # Метод Зейделя
plot(zeyd,analytic, x, y)

rl = relax(1.5) # Метод верхней релаксации
plot(rl,analytic, x, y)
```

5. Результат

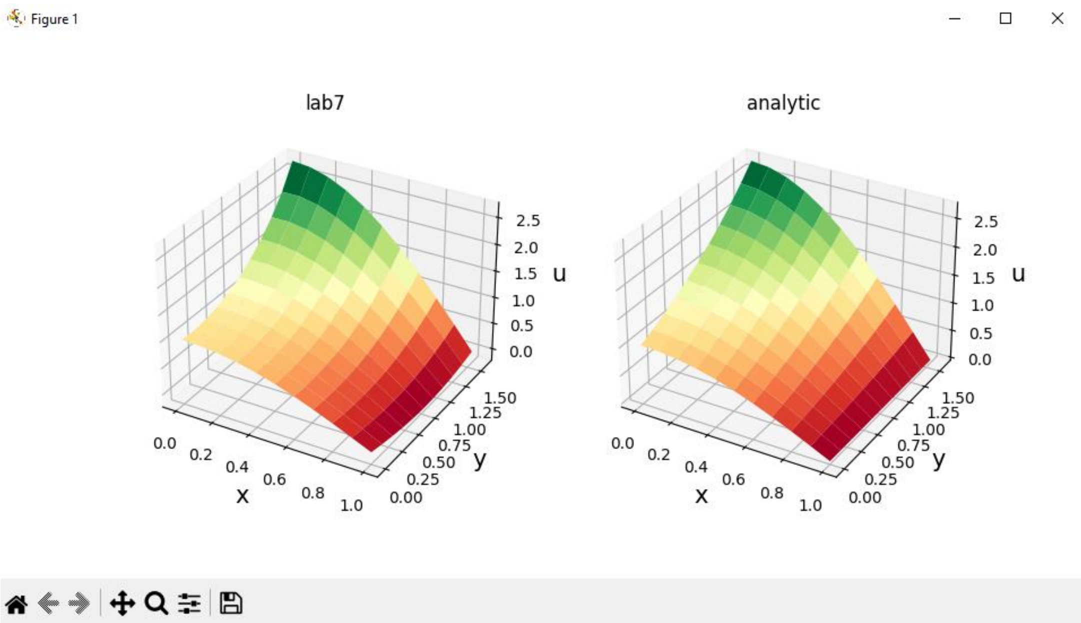
Метод Либмана:



Метод Зейделя:



Метод верхней релаксации:




```
Метод Либмана
Шагов:  105
Погрешность:  0.0009864968489001619
Метод Зейделя
Шагов:  65
Погрешность:  0.0009610149259369205
Метод верхней релаксации
Шагов:  31
Погрешность:  0.0009123901609551233

Process finished with exit code 0
```