**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10**

**ПРИБЛИЖЕННЫЕ МЕТОДЫ РЕШЕНИЯ**

**НЕЛИНЕЙНЫХ УРАВНЕНИЙ**

**(Вариант 10)**

*Выполнил студент 3 курса МОиАИС*

*Соколов Арсений*

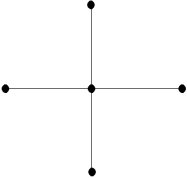
**Задание:**

Решить эллиптическое уравнение

методами 2-го порядка точности.

Сетки по x и по y взять равномерные.

Шаблон для разностной схемы:



Для решения разностных уравнений применить:

А) метод простой итерации

Б) метод Зейделя

Оценивать погрешность решения с помощью сравнения двух последовательных итераций.

Взять сетки размерами 5×5 ячеек и 10×10 ячеек и сравнить полученные решения.

**Дано:**

[a, b] = [0; 10],

[c, d] = [0; 10],

Погрешность решения = 0,01,

Граничные условия

|  |  |
| --- | --- |
| № вар. | Правая часть |
| 10 |  |

**Решение:**

Разностная схема второго порядка точности, шаблон крест

Метод итераций

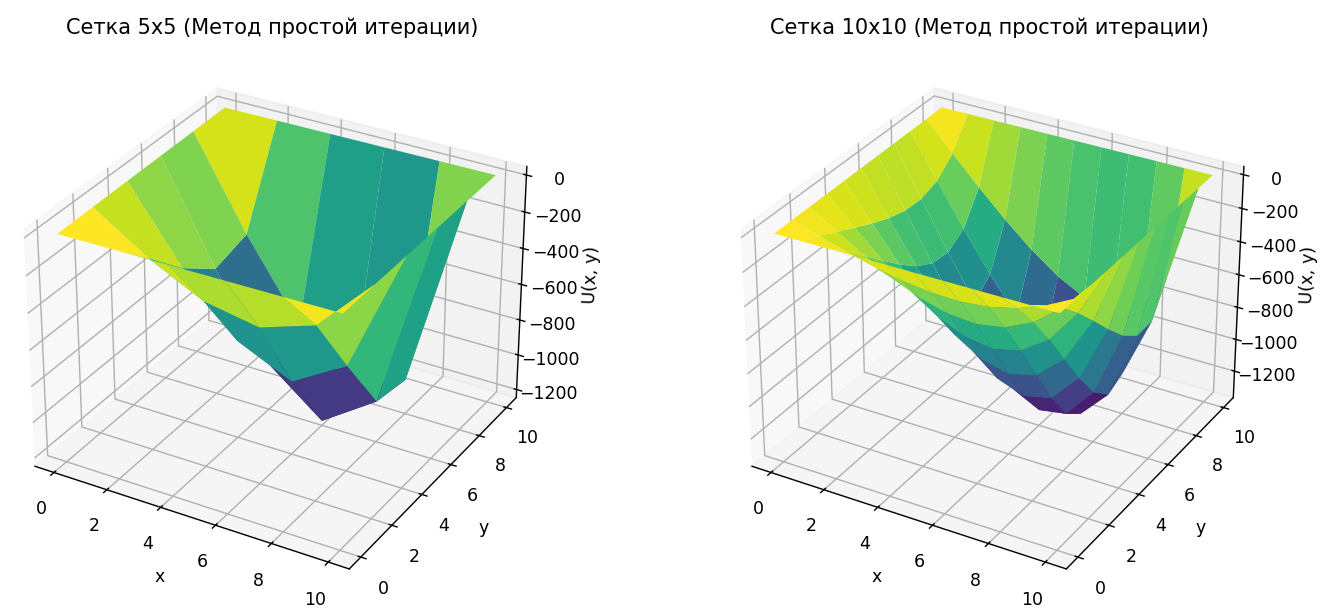
Обновляем значения на сетке на каждой итерации, используя значения из предыдущей

Итерационный процесс продолжается пока

Разбиение области

​

Тогда



Метод Зейделя

– модификация метода итераций, когда в правой части уже посчитанные значения

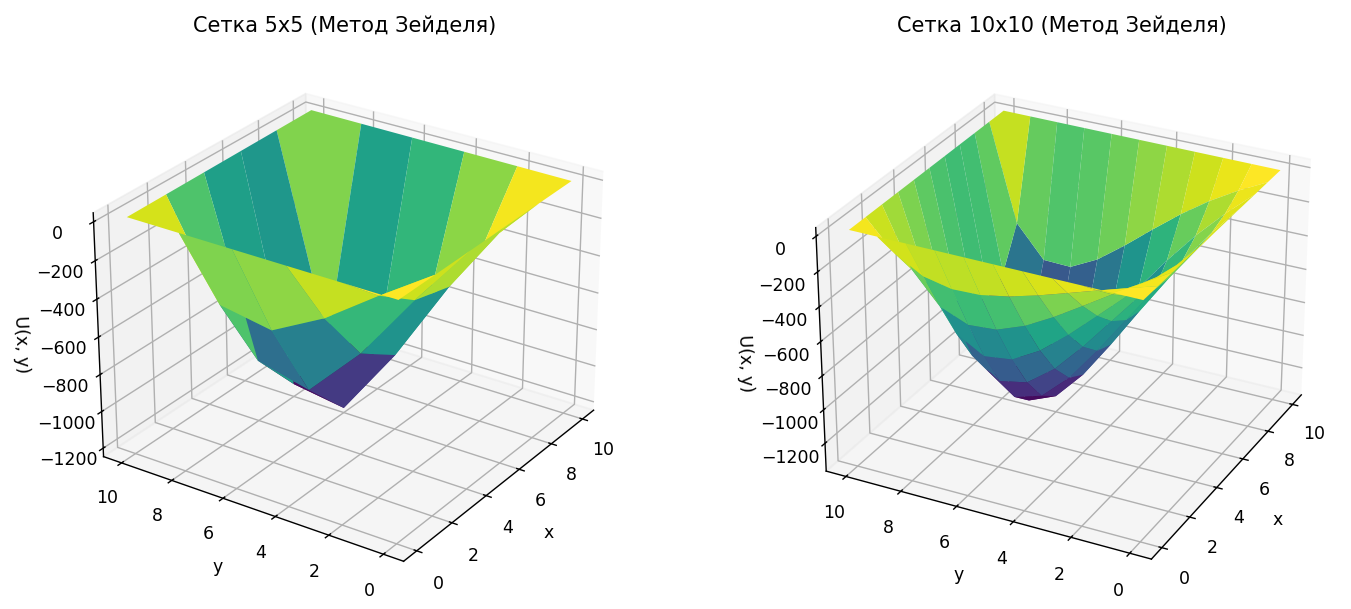
Итерационный процесс продолжается пока

Начальные значения *U* задаются на основе граничных условий

Разбиение области

​

Тогда



**ПРИЛОЖЕНИЕ 1**

***Программа решения эллиптического уравнения методом итераций***

import numpy as np  
import matplotlib.pyplot as plt  
  
# Параметры  
a, b = 0, 10  
c, d = 0, 10  
epsilon = 0.01  
  
  
# Правая часть  
def f(x, y):  
 return x \* y \*\* 2  
  
  
# Граничные условия  
def boundary\_conditions(U, x, y):  
 U[:, 0] = x + c  
 U[:, -1] = x + d  
 U[0, :] = a + y  
 U[-1, :] = b + y  
 return U  
  
  
# Метод простой итерации  
def simple\_iteration(N, M):  
 h = (b - a) / N  
 k = (d - c) / M  
  
 x = np.linspace(a, b, N + 1)  
 y = np.linspace(c, d, M + 1)  
  
 U = np.zeros((N + 1, M + 1))  
 U = boundary\_conditions(U, x, y)  
 U\_new = np.copy(U)  
  
 max\_iterations = 10000  
 for iteration in range(max\_iterations):  
 for i in range(1, N):  
 for j in range(1, M):  
 U\_new[i, j] = 0.25 \* (U[i + 1, j] + U[i - 1, j] + U[i, j + 1] + U[i, j - 1] - h \*\* 2 \* f(x[i], y[j]))  
  
 if np.linalg.norm(U\_new - U, ord=np.inf) < epsilon:  
 break  
  
 U = np.copy(U\_new)  
  
 return x, y, U  
  
  
# Решение для сеток  
x5, y5, U5 = simple\_iteration(5, 5)  
x10, y10, U10 = simple\_iteration (10, 10)  
  
# Визуализация  
X5, Y5 = np.meshgrid(x5, y5)  
X10, Y10 = np.meshgrid(x10, y10)  
  
fig = plt.figure(figsize=(12, 5))  
  
ax1 = fig.add\_subplot(121, projection='3d')  
ax1.plot\_surface(X5, Y5, U5.T, cmap='viridis')  
ax1.set\_title('Сетка 5x5 (Метод простой итерации)')  
ax1.set\_xlabel('x')  
ax1.set\_ylabel('y')  
ax1.set\_zlabel('U(x, y)')  
  
ax2 = fig.add\_subplot(122, projection='3d')  
ax2.plot\_surface(X10, Y10, U10.T, cmap='viridis')  
ax2.set\_title('Сетка 10x10 (Метод простой итерации)')  
ax2.set\_xlabel('x')  
ax2.set\_ylabel('y')  
ax2.set\_zlabel('U(x, y)')  
  
plt.tight\_layout()  
plt.show()

**ПРИЛОЖЕНИЕ 2**

***Программа решения эллиптического уравнения методом Зейделя***

import numpy as np  
import matplotlib.pyplot as plt  
  
# Параметры  
a, b = 0, 10  
c, d = 0, 10  
epsilon = 0.01  
  
  
# Правая часть  
def f(x, y):  
 return x \* y \*\* 2  
  
  
# Граничные условия  
def boundary\_conditions(U, x, y):  
 U[:, 0] = x  
 U[:, -1] = x + (d - c)  
 U[0, :] = y  
 U[-1, :] = (b - a) + y  
 return U  
  
  
# Метод Зейделя  
def seidel\_method(N, M):  
 h = (b - a) / N  
 k = (d - c) / M  
  
 x = np.linspace(a, b, N + 1)  
 y = np.linspace(c, d, M + 1)  
  
 U = np.zeros((N + 1, M + 1))  
 U = boundary\_conditions(U, x, y)  
  
 max\_iterations = 10000  
 for iteration in range(max\_iterations):  
 U\_old = np.copy(U)  
 for i in range(1, N):  
 for j in range(1, M):  
 U[i, j] = 0.25 \* (U[i + 1, j] + U[i - 1, j] + U[i, j + 1] + U[i, j - 1] - h \*\* 2 \* f(x[i], y[j]))  
  
 if np.linalg.norm(U - U\_old, ord=np.inf) < epsilon:  
 break  
  
 return x, y, U  
  
  
# Решение для сеток  
x5, y5, U5 = seidel\_method(5, 5)  
x10, y10, U10 = seidel\_method(10, 10)  
  
# Визуализация  
X5, Y5 = np.meshgrid(x5, y5)  
X10, Y10 = np.meshgrid(x10, y10)  
  
fig = plt.figure(figsize=(12, 5))  
  
ax1 = fig.add\_subplot(121, projection='3d')  
ax1.plot\_surface(X5, Y5, U5.T, cmap='viridis')  
ax1.set\_title('Сетка 5x5 (Метод Зейделя)')  
ax1.set\_xlabel('x')  
ax1.set\_ylabel('y')  
ax1.set\_zlabel('U(x, y)')  
  
ax2 = fig.add\_subplot(122, projection='3d')  
ax2.plot\_surface(X10, Y10, U10.T, cmap='viridis')  
ax2.set\_title('Сетка 10x10 (Метод Зейделя)')  
ax2.set\_xlabel('x')  
ax2.set\_ylabel('y')  
ax2.set\_zlabel('U(x, y)')  
  
plt.tight\_layout()  
plt.show()