Лабораторная работа № 7

Бабичева Анна М8О-404Б-17

Вариант № 2

Импортирую необходимые библиотеки, включая собственную matrix и graphics_for_labs.

In [1]:
```
import numpy as np
import pandas as pd
import math
import pylab
import matplotlib.pyplot as plt
import random
import matrix

from graphics_for_labs import Graphic
from numpy import arange
from numpy import meshgrid
from matplotlib import mlab
from sys import stdin
from copy import deepcopy

```

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.

Краевая задача:

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \\ u_x'(0, y) = 0 \\ u(1, y) = 1 - y^2 \\ u_y'(x, 0) = 0 \\ u(x, 1) = x^2 - 1 \end{cases}$$

Аналитическое решение:

$$U(x, y) = x^2 - y^2$$

In [2]:
```python
def U(x, y):
    return x * x - y * y

```

Центрально-разностная аппроксимация:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} + O(h_x^2 + h_y^2) = 0$$

где $i = 1, \ldots, N_x - 1, j = 1, \ldots, N_y - 1.$

Краевые условия аппроксимируются в виде:

$$u_{0,j} = u_{1,j}$$

$$u_{N_x,j} = 1 - y_j^2$$

$$u_{i,0} = u_{i,1}$$

$$u_{i,N_y} = x_i^2 - 1$$

In [3]:
```python
def u0j(u, j):
    return u[1][j]

def ui0(u, i):
    return u[i][1]

def uNj(y, j):
    return 1 - y[j] ** 2

def uiN(x, i):
    return x[i] ** 2 - 1

```

Такая аппроксимация требует решение СЛАУ, записанной в виде пятидиагональной матрицы.

Метод Либмана

Итерационно-разностный метод имеет вид:

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

In [4]:
```
def f(x, y, i, j):
    return 0

```

при $h_x = h_y = h$.

Критерий останова:

$||u^{(k+1)} - u^{(k)}|| \le \epsilon$

$$\left\|u^{(k)}\right\| = \max_{i,j} \left|u_{i,j}^{(k)}\right|$$

In [5]:
```
def norm(u):
    return u.Max()

```

На нулевой итерации значения определяются из краевых условий с помощью линейной интерполяции:

$$u_{N_x,j}^{(0)} = 1 - y_j^2$$

$$u_{i,N_y}^{(0)} = x_i^2 - 1$$

$$u_{0,j}^{(0)} = u_{1,j}^{(0)}$$

$$u_{i,0}^{(0)} = u_{i,1}^{(0)}$$

$$u_{i,j}^{(0)} = \frac{1 - x_i}{1 - x_i + 1 - y_j}(1 - y_j^2) + \frac{1 - y_j}{1 - x_i + 1 - y_j}(x_i^2 - 1)$$

In [19]:
```
def Libman(N, l=0, r=1, eps=0.01, method=1, p=1.5, u0j=u0j, ui0
    h = (r - l) / N

    x = [l + i * h for i in range(N + 1)]
    y = [l + i * h for i in range(N + 1)]

    u = []

    row_x = []

    for i in range(N):
        row_y = [0] * (N + 1)
        row_y[N] = uiN(x, i)
        row_x.append(row_y)
    row_y = []
    for j in range(N + 1):
        row_y.append(uNj(y, j))
    row_x.append(row_y)

    for i in range(1, N):
```

```
        for j in range(1, N):
            row_x[i][j] = (r - x[i]) * uNj(y, j) / (r - x[i] +

    for i in range(N):
        row_x[i][0] = ui0(row_x, i)
        row_x[0][i] = u0j(row_x, i)

    u.append(row_x)

    k = 0
    e = eps * 2

    while e > eps:

        row_x = np.array([[0.] * (N + 1)] * (N + 1))
        for i in range(N):

            row_x[i][N] = uiN(x, i)
            row_x[N][i] = uNj(y, i)

        if method == 1:
            for i in range(1, N):
                for j in range(1, N):
                    row_x[i][j] = (u[k][i + 1][j] + u[k][i - 1]
        elif method == 2:
            for i in range(1, N):
                for j in range(1, N):
                    row_x[i][j] = (u[k][i + 1][j] + row_x[i - 1
        elif method == 3:
            for i in range(1, N):
                for j in range(1, N):
                    row_x[i][j] = ((1 - p) * u[k][i][j] + u[k][
        else:
            raise Exception("1 - метод простых итераций, 2 - ме

        for i in range(N):
            row_x[i][0] = ui0(row_x, i)
            row_x[0][i] = u0j(row_x, i)
        u.append(row_x)

        e = norm(matrix.Matrix(u[k + 1]).Sum(matrix.Matrix(u[k]
        k += 1
    res = u[k]
    return x, y, res, k, e

```

Метод простых итераций

In [7]:

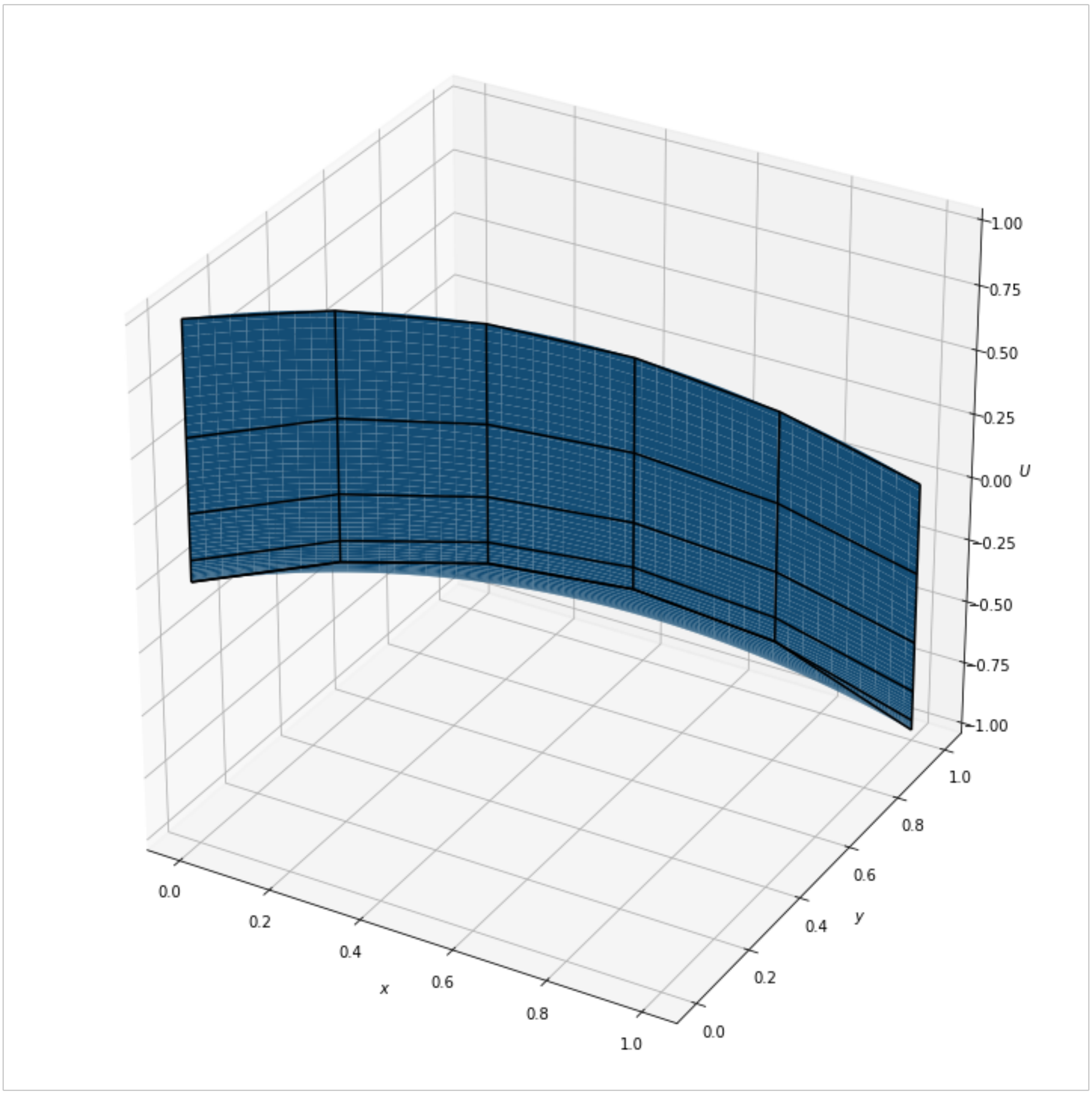
```
x, y, u, k, e = Libman(5)
print("Кол-во итераций: {}".format(k))
print("Эпсилон: {}".format(e))

```

```
Кол-во итераций: 10
Эпсилон: 0.008469118935721276
```

In [8]:

```
Graphic.draw_surface(x, y, u, U, 0, 1, 0, 1)

```

```
/home/ann/Документы/ЧМ/graphics_for_labs.py:142: UserWarning: This
figure includes Axes that are not compatible with tight_layout, so
results might be incorrect.
  fig.tight_layout()
```

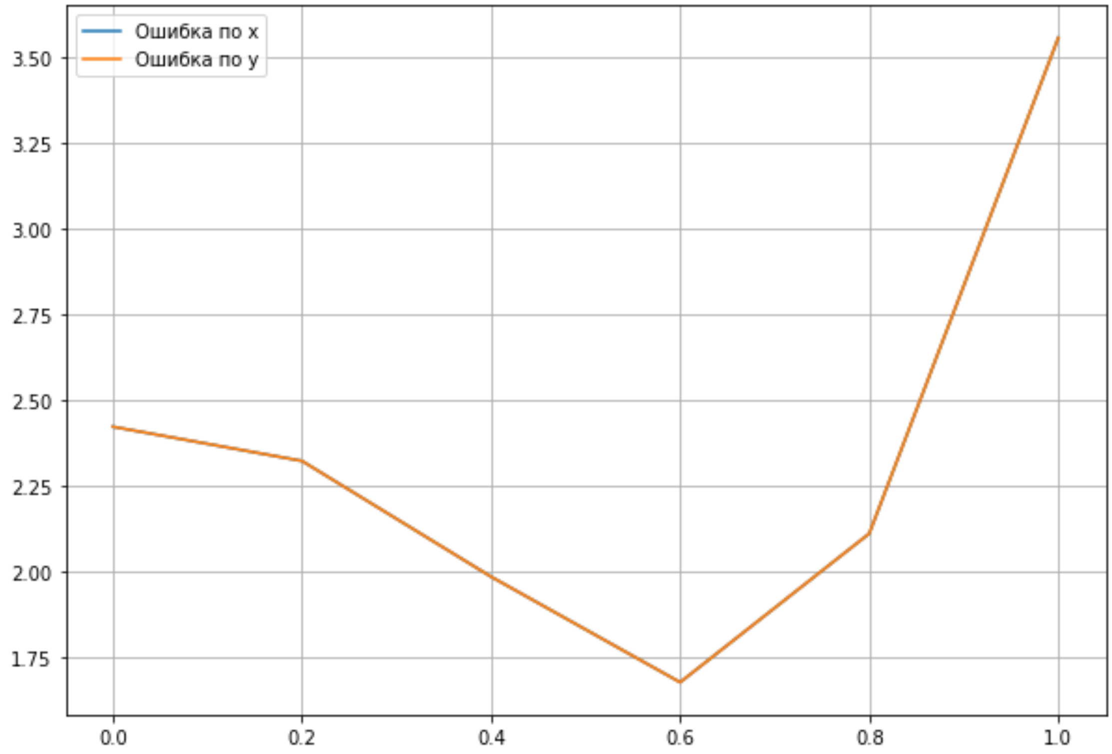


In [9]:

```
Graphic.draw_3d(x, y, u, U, 0, 1, 0, 1, 'Метод Либмана')

```

Метод Либмана

In [10]:
```
Graphic.draw_variance_xy(x, y, u, U)

```



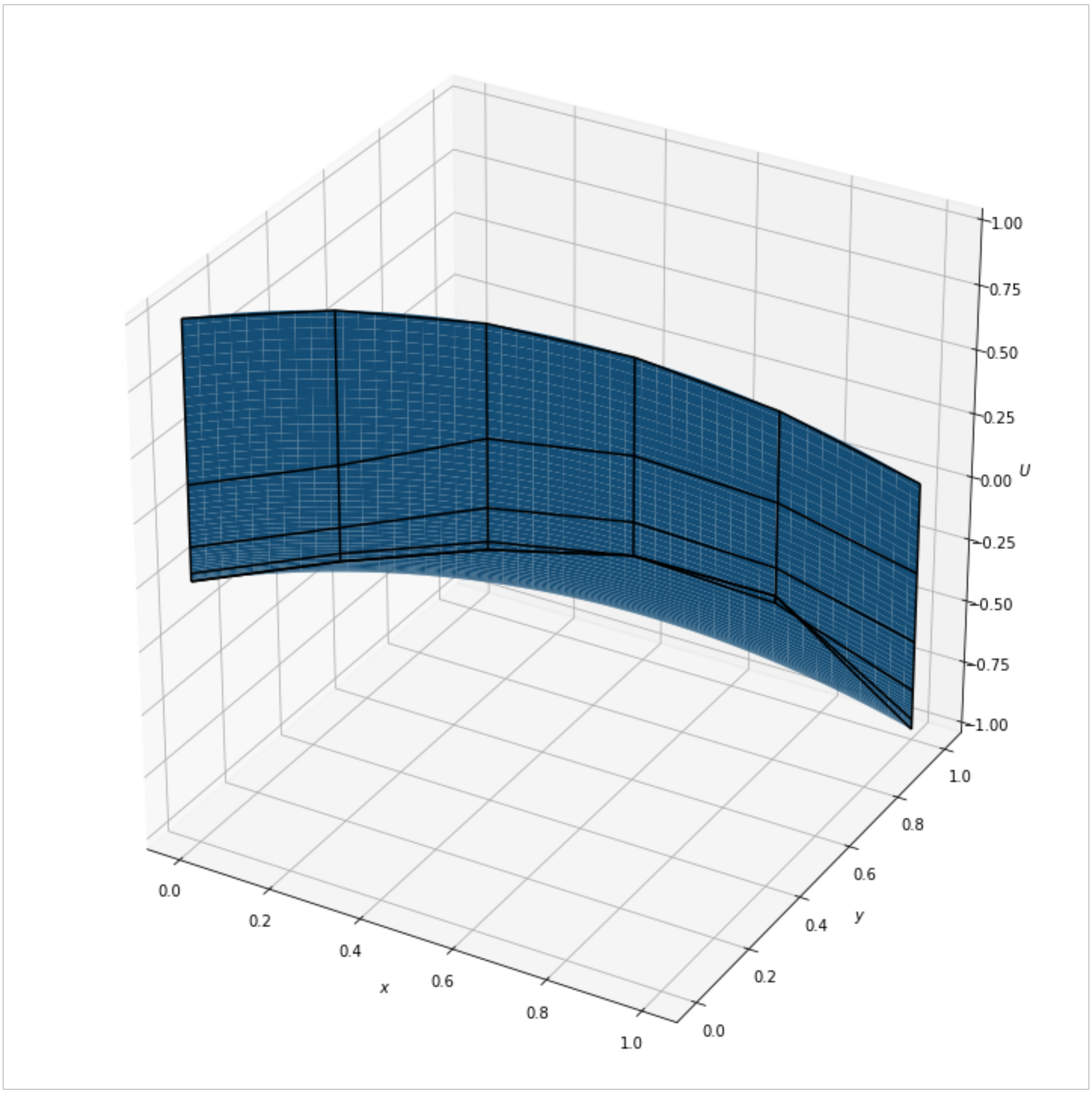
Метод Зейделя

In [11]:
```
x, y, u, k, e = Libman(5, method=2)
print("Кол-во итераций: {}".format(k))
print("Эпсилон: {}".format(e))

```

```
Кол-во итераций: 5
Эпсилон: 0.008052368164062504
```

In [12]:
```
Graphic.draw_surface(x, y, u, U, 0, 1, 0, 1)

```

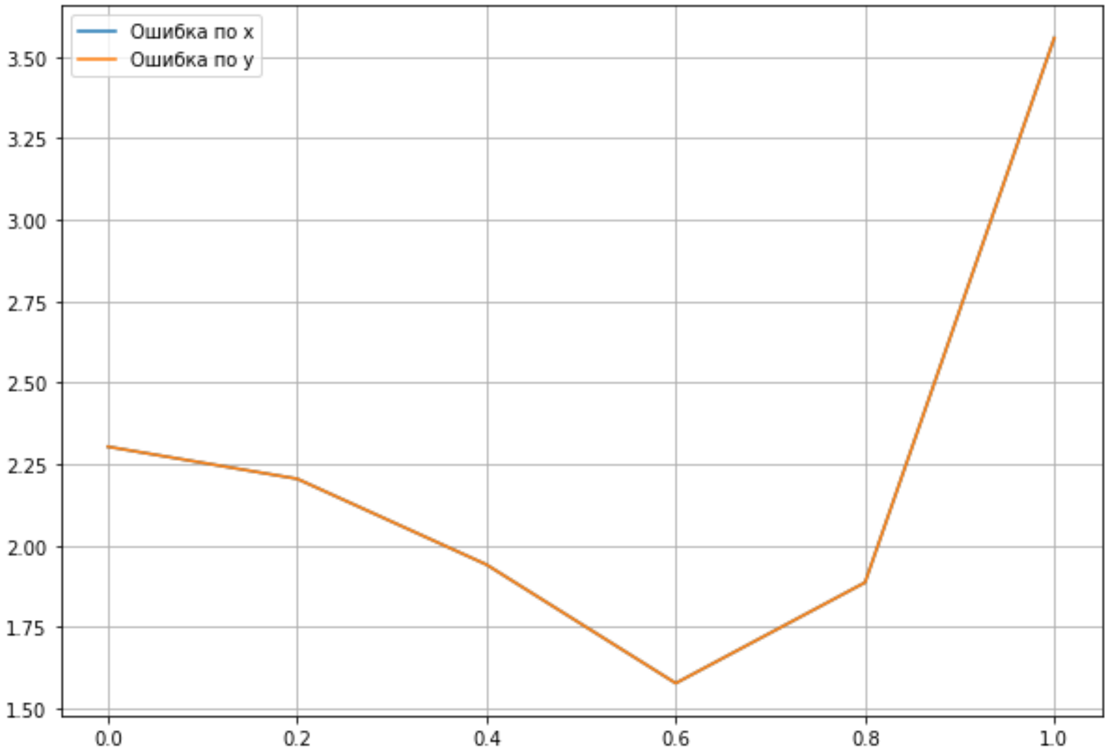
```
/home/ann/Документы/ЧМ/graphics_for_labs.py:142: UserWarning:

This figure includes Axes that are not compatible with tight_layou
t, so results might be incorrect.
```



In [13]:
```
Graphic.draw_3d(x, y, u, U, 0, 1, 0, 1, 'Метод Зейделя')

```

Метод Зейделя

In [14]:
```
Graphic.draw_variance_xy(x, y, u, U)

```

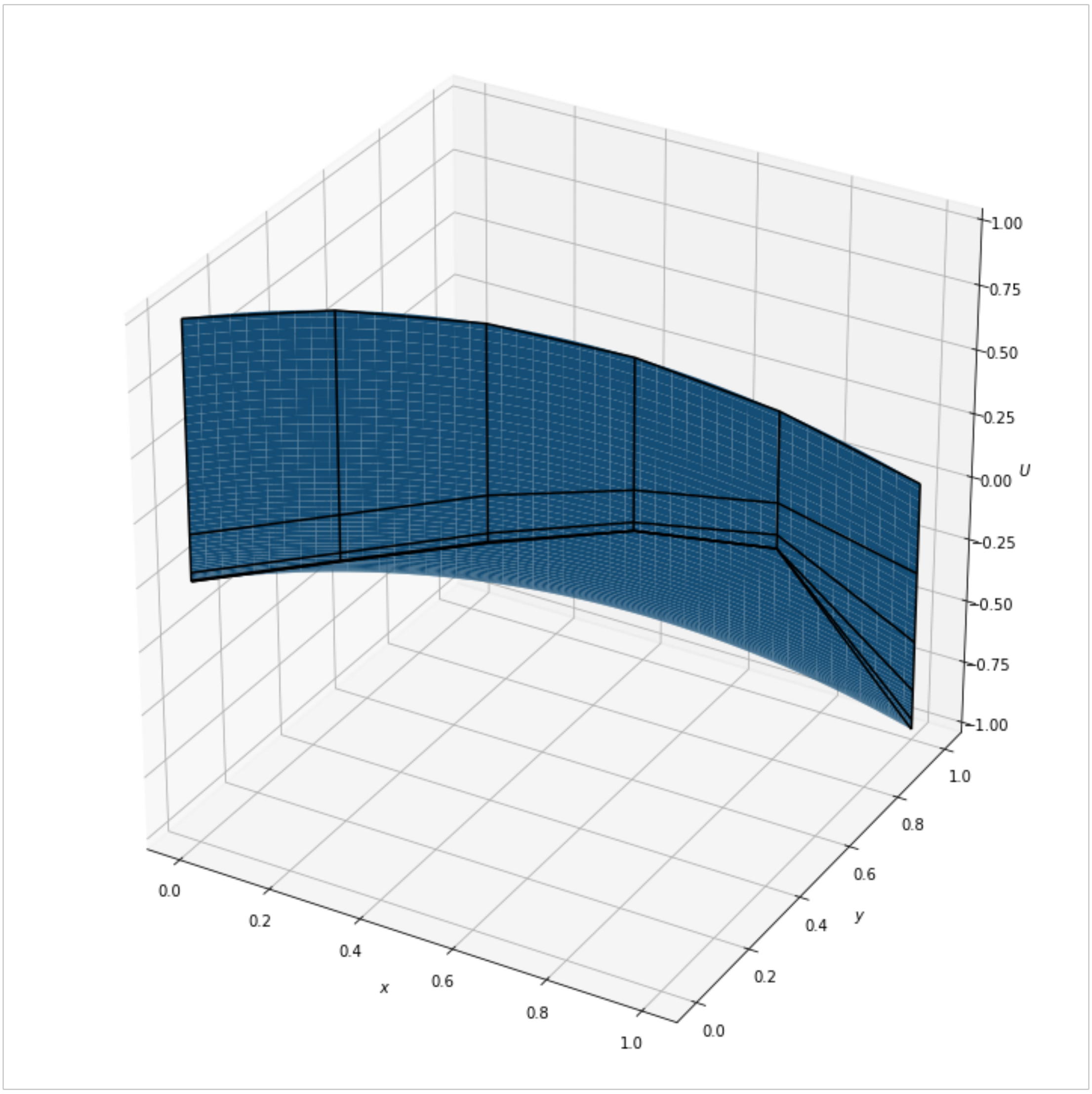


Релаксация

In [20]:
```
x, y, u, k, e = Libman(5, method=3)
print("Кол-во итераций: {}".format(k))
print("Эпсилон: {}".format(e))

```

```
Кол-во итераций: 3
Эпсилон: 0.008322816643804291
```

In [21]:
```
Graphic.draw_surface(x, y, u, U, 0, 1, 0, 1)

```

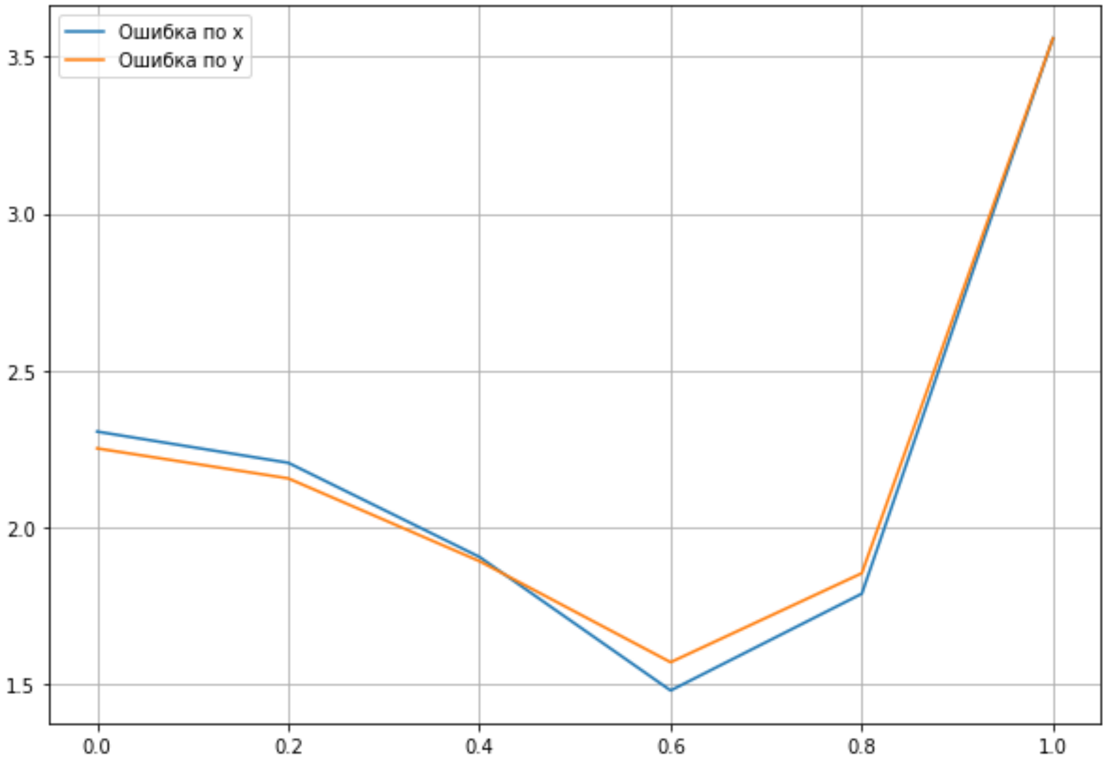


In [22]:
```
Graphic.draw_3d(x, y, u, U, 0, 1, 0, 1, 'Релаксация')

```

Релаксация

In [18]:

```
Graphic.draw_variance_xy(x, y, u, U)

```



In [ ]:

```

```