**ОТЧЁТ**

**ЛАБОРАТОРНАЯ РАБОТА №6**

**Линейное уравнение переноса**

**(Вариант 9)**

Выполнил студент 3 курса МОиАИС

Сагитов Александр

**Задание**

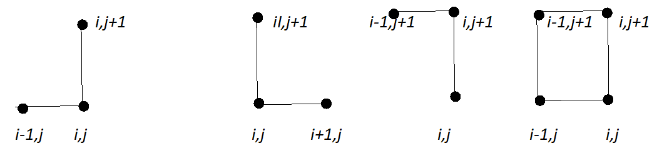
Численно решить уравнение переноса



1) Для полуплоскости −∞<𝑥<∞;𝑡≥0

2) В прямоугольнике 0≤𝑥≤1;0≤𝑡≤10

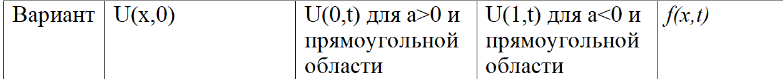
Во всех случаях a – const. Применить следующие шаблоны для явных и неявных схем.

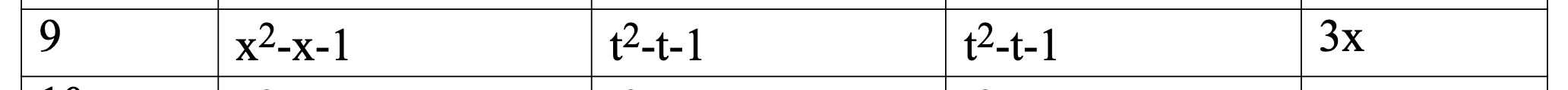


Схемы выбирать в зависимости от знака a. Причем параметр a принимает два значения: a=2 и a= -2

Для полуплоскости и для прямоугольной области решить задачу от 0 до 1 с шагом 0.1 по х и от 0 до 10 по времени с шагом, отвечающим условиям устойчивости. Применить все возможные схемы.

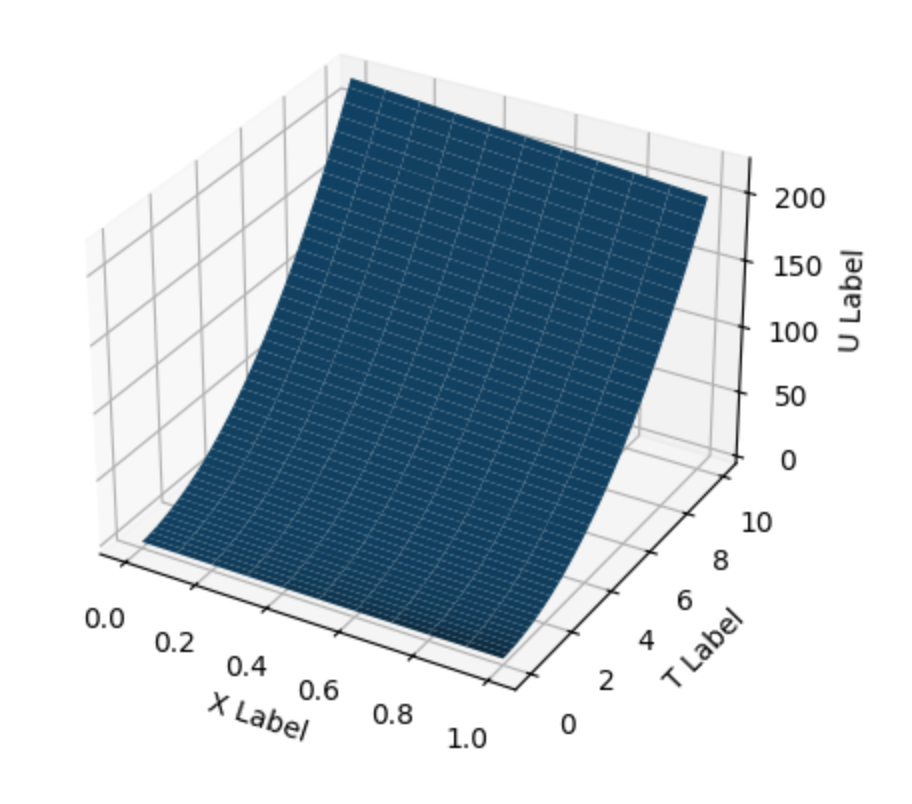
Рассмотреть и решить задачу для всех возможных схем для двух случаев: однородного уравнения с нулевой правой частью и правой частью, указанной в варианте задания.



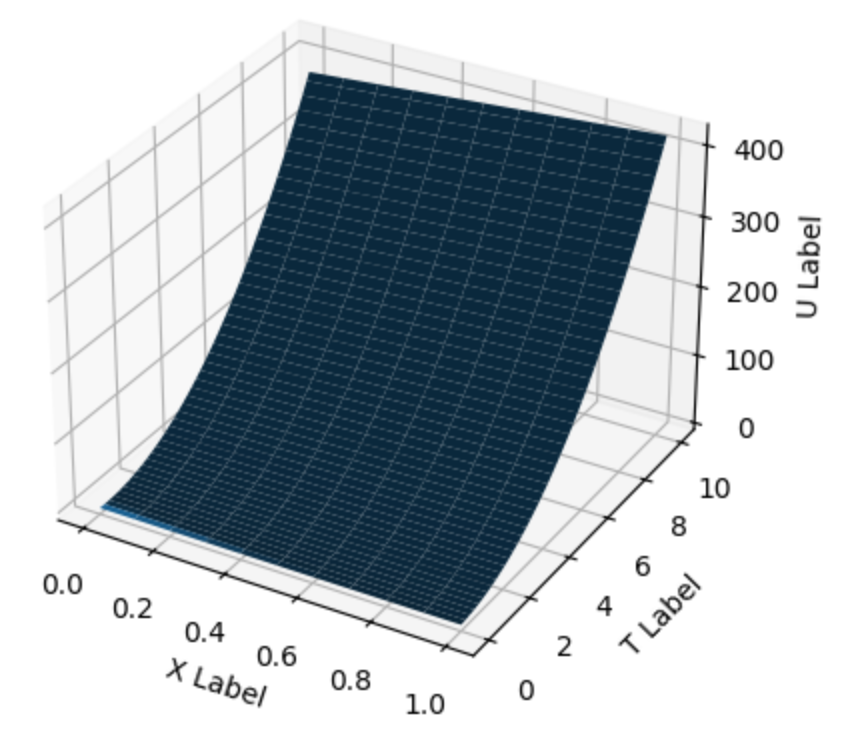


**Решение**

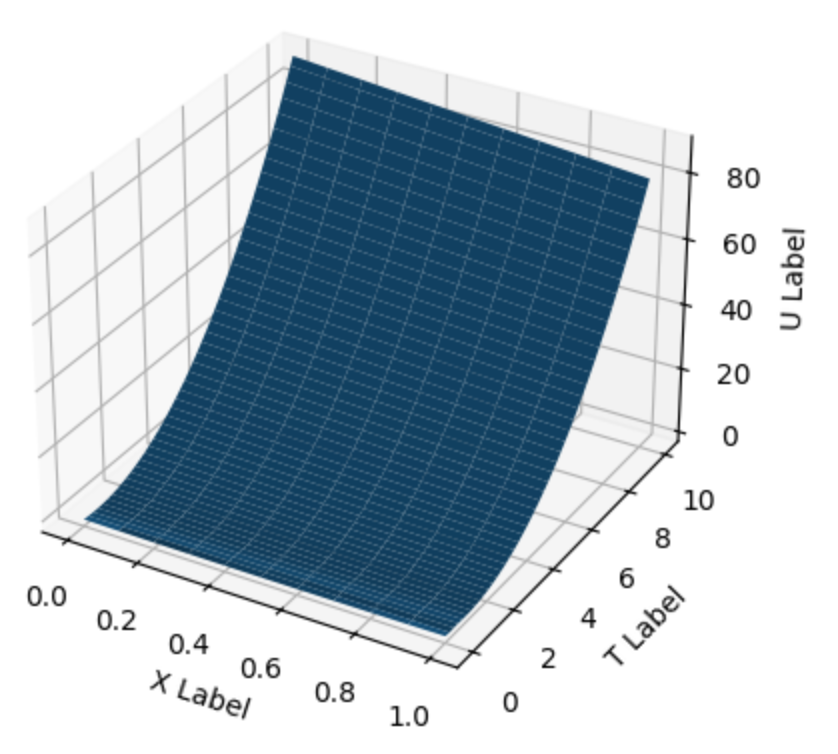
**1.f=0**

Полуплоскость, a>0, правый нижний угол

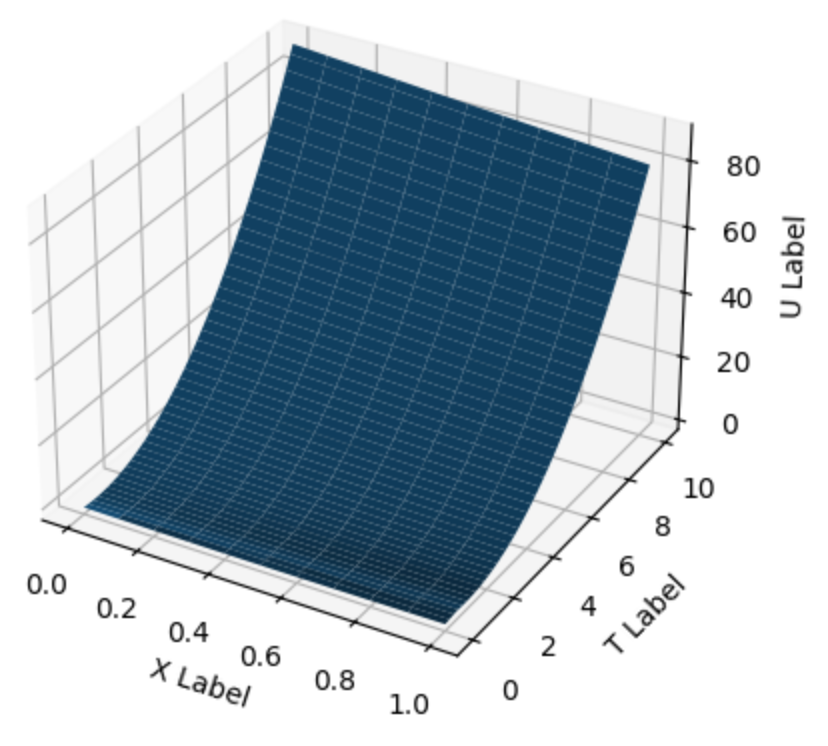
Полуплоскость, a<0, левый нижний угол



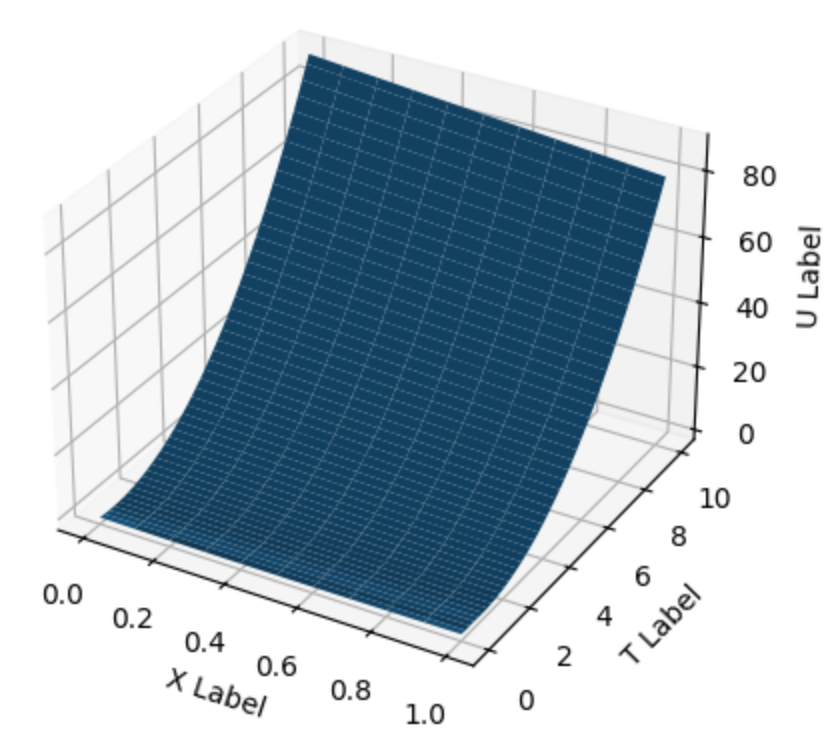
Прямоугольник, a>0, правый нижний угол



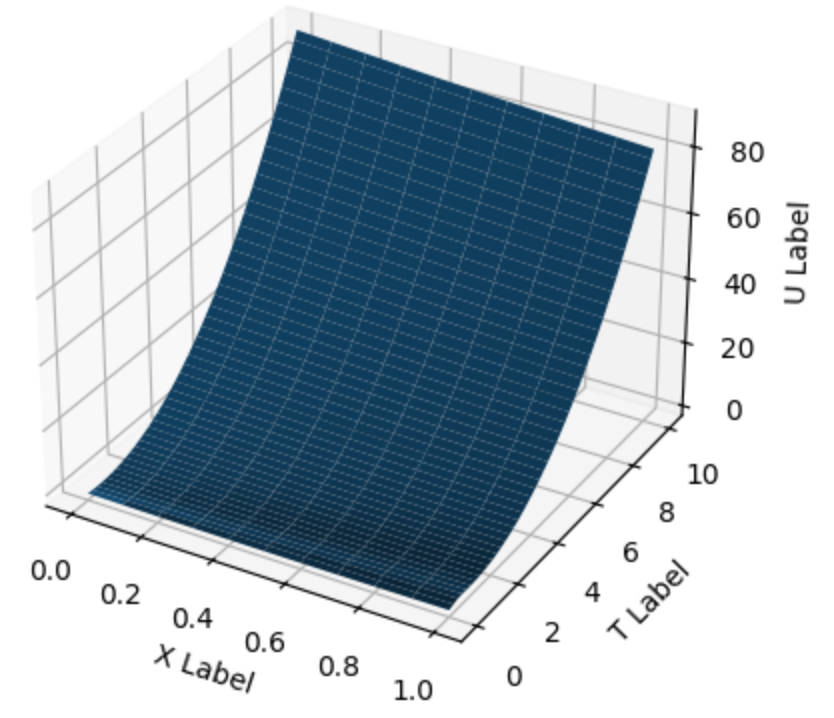
Прямоугольник, а>0, правый верхний угол



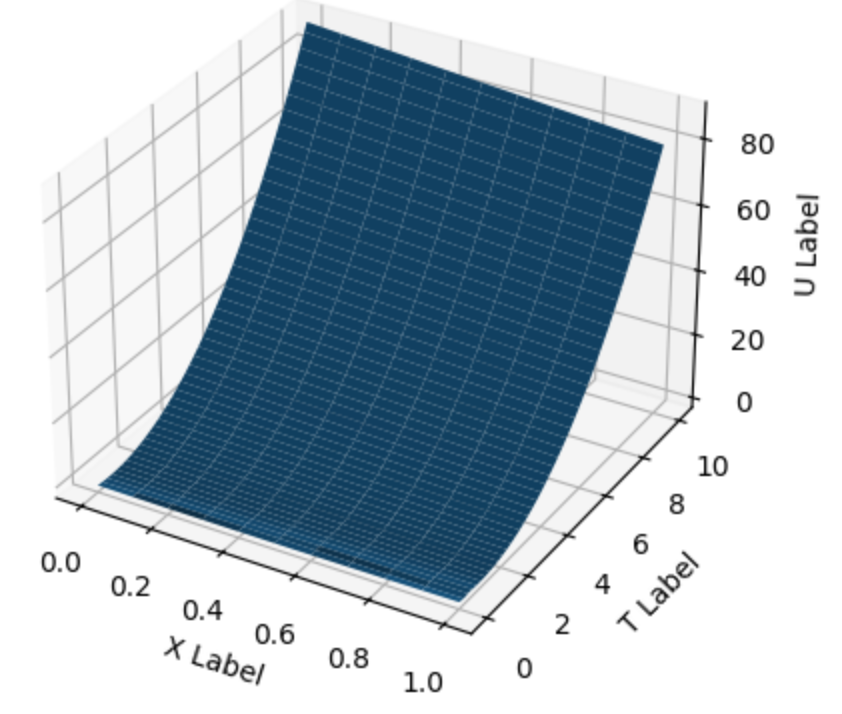
Прямоугольник, a>0, прямоугольник



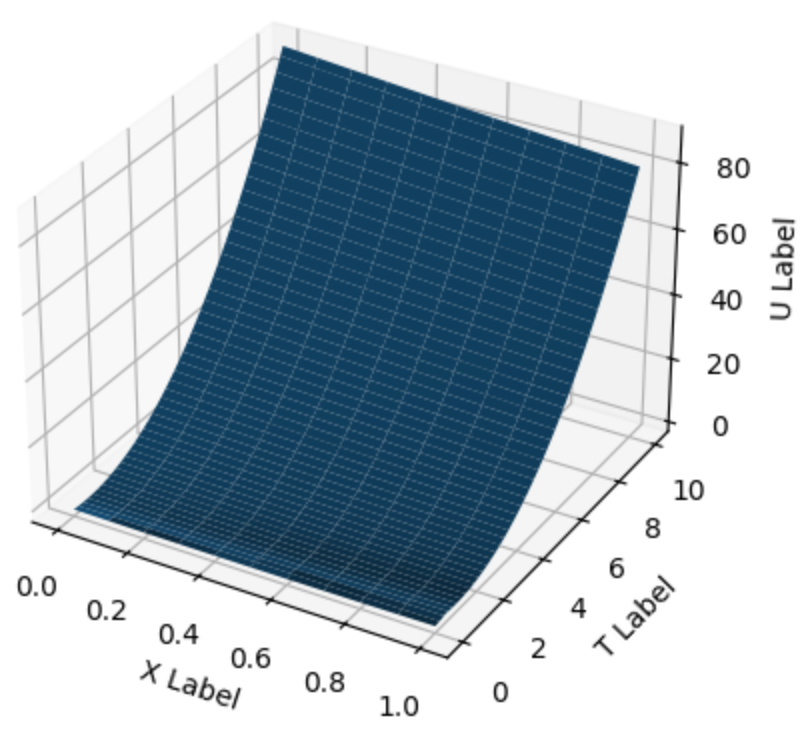
Прямоугольник, a<0, левый нижний угол



Прямоугольник, a<0, левый верхний угол

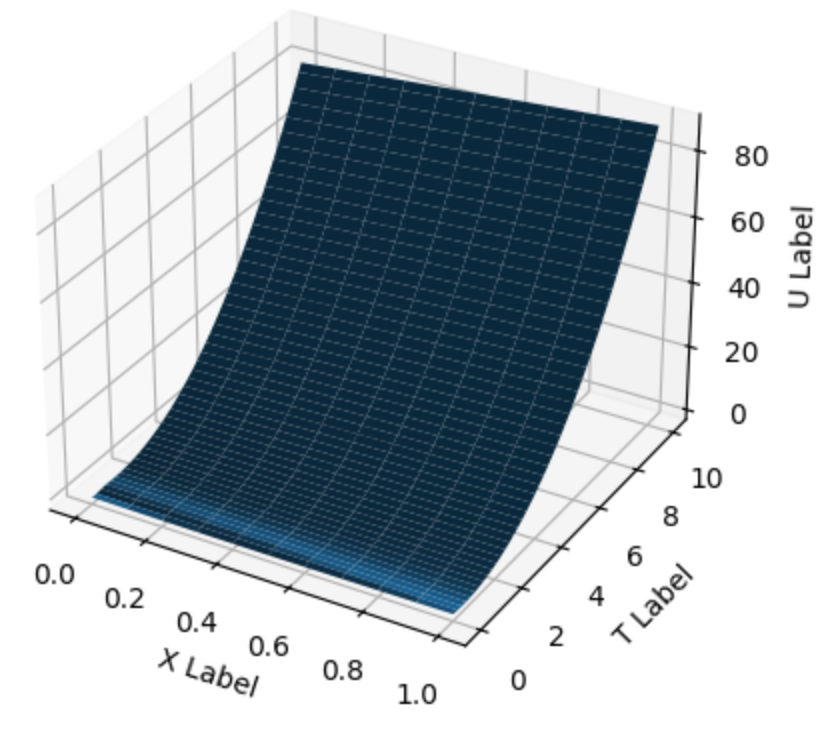


Прямоугольник, a<0, прямоугольник

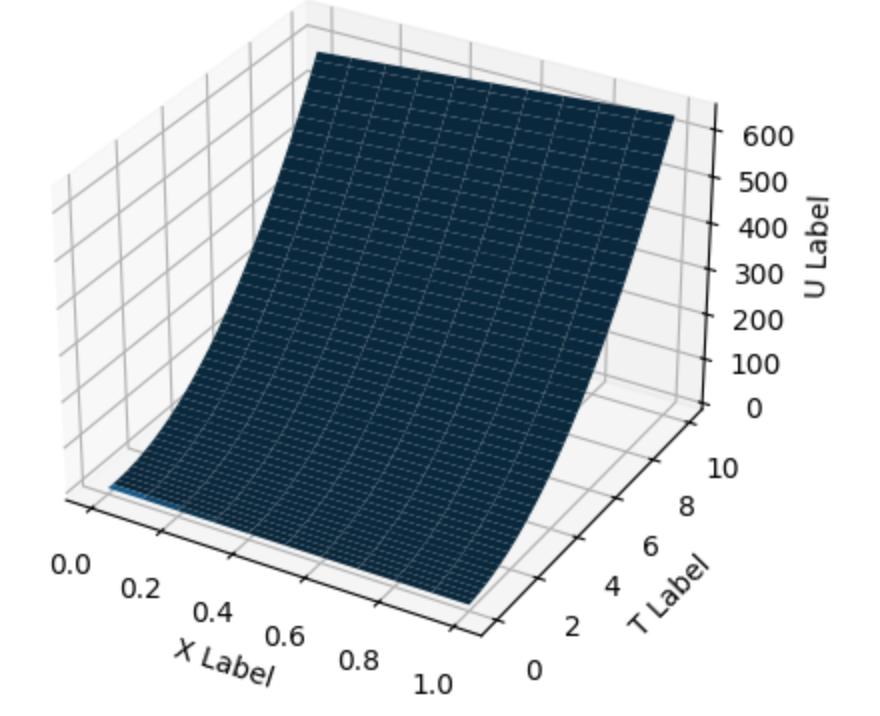


**2.f=x**

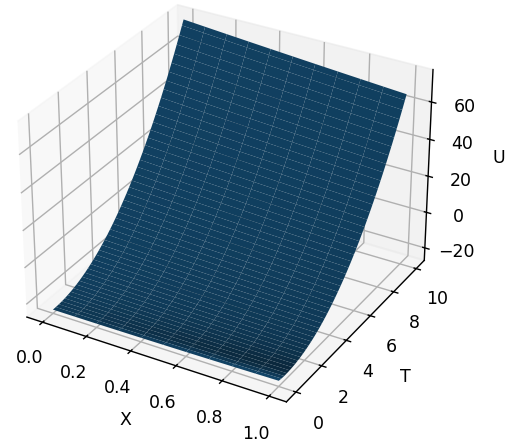
Полуплоскость, a>0, правый нижний угол



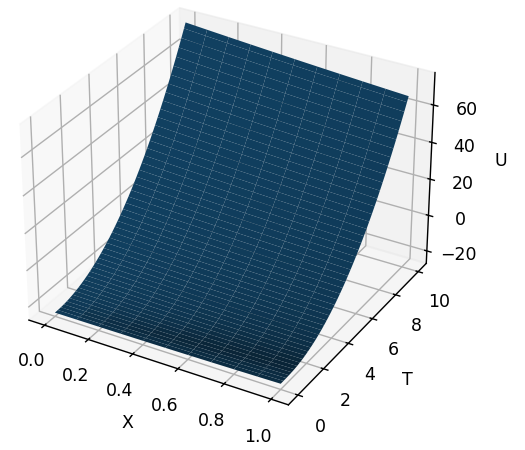
Полуплоскость, a<0, левый нижний угол



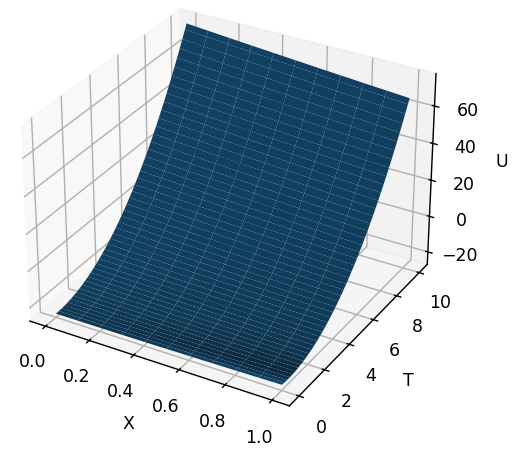
Прямоугольник, a>0, правый нижний угол



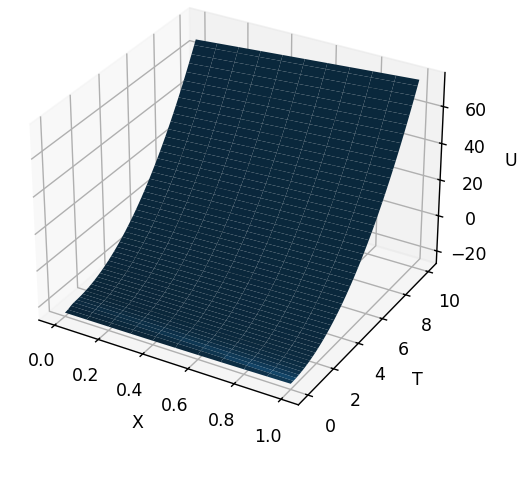
Прямоугольник, a>0, правый верхний угол



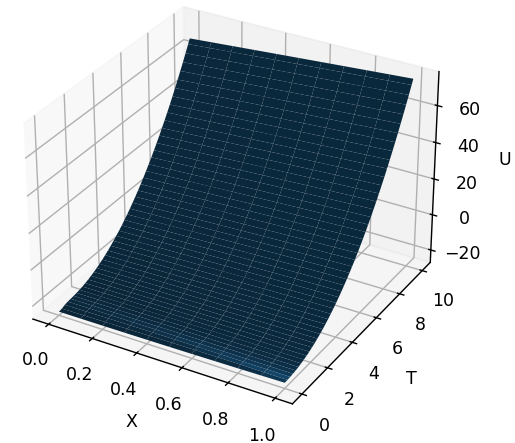
Прямоугольник, a>0, прямоугольник



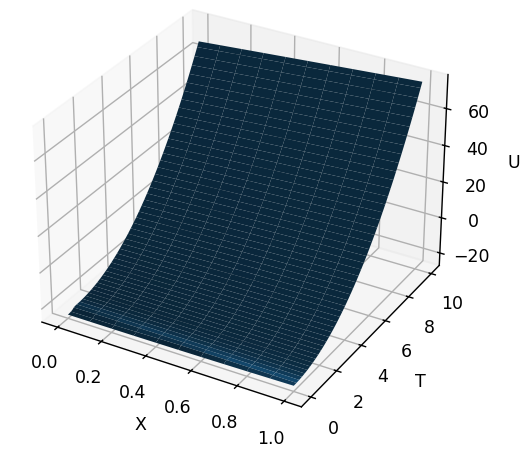
Прямоугольник, a<0, левый нижний угол



Прямоугольник, a<0, левый верхний угол



Прямоугольник, a<0, прямоугольник



**Приложение**

*main.py*

import numpy as np  
import matplotlib.pyplot as plt  
  
  
def ux0(x):  
 return x\*\*2 - x - 1  
  
  
def u0t(t):  
 return t\*\*2 - t - 1  
  
  
def u1t(t):  
 return t\*\*2 - t - 1  
  
  
# a>0 f=x правый нижний угол  
def fun2():  
 u = np.zeros((1011, 1001))  
 x = np.zeros(1011)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = 2  
 for i in range(1010, -1, -1):  
 x[i] = (i \* h) - 100  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(j, 1011):  
 u[i][j] = lam \* u[i - 1][j - 1] + (1 - lam) \* u[i][j - 1] + 2 \* tau \* x[i]  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i + 1000][j]  
 xres[i][j] = x[i + 1000]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=0 левый нижний угол  
def fun3():  
 u = np.zeros((1011, 1001))  
 x = np.zeros(1011)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 1011, 1):  
 x[i] = i \* h  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(0, 1011 - j):  
 u[i][j] = (lam + 1) \* u[i][j - 1] - lam \* u[i + 1][j - 1]  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=x левый нижний угол  
def fun4():  
 u = np.zeros((1011, 1001))  
 x = np.zeros(1011)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 1011, 1):  
 x[i] = i \* h  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(0, 1011 - j):  
 u[i][j] = (lam + 1) \* u[i][j - 1] - lam \* u[i + 1][j - 1] + 2 \* tau \* x[i]  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# решение в прямоугольнике  
# для решения в прямоугольнике никаких фиктивных элементов не требуется  
  
# a>0 f=0 правый нижний угол  
def fun5():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = 2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[0][i] = u0t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(1, 11):  
 u[i][j] = lam \* u[i - 1][j - 1] + (1 - lam) \* u[i][j - 1]  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a>0 f=x правый нижний угол  
def fun6():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = 2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[0][i] = u0t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(1, 11):  
 u[i][j] = lam \* u[i - 1][j - 1] + (1 - lam) \* u[i][j - 1] + 2 \* tau \* x[i]  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a>0 f=0 правый верхний угол  
def fun7():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = 2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[0][i] = u0t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(1, 11):  
 u[i][j] = (lam \* u[i - 1][j] + u[i][j - 1]) / (lam + 1)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a>0 f=x правый верхний угол  
def fun8():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = 2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[0][i] = u0t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(1, 11):  
 u[i][j] = (lam \* u[i - 1][j] + u[i][j - 1] + 4 \* tau \* x[i]) / (lam + 1)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a>0 f=0 правый прямоугольник  
def fun9():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = 2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[0][i] = u0t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(1, 11):  
 u[i][j] = ((lam + 1) \* u[i - 1][j - 1] + (1 - lam) \* (u[i][j - 1] - u[i - 1][j])) / (lam + 1)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a>0 f=x правый прямоугольник  
def fun10():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = 2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[0][i] = u0t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(1, 11):  
 u[i][j] = ((lam + 1) \* u[i - 1][j - 1] + (1 - lam) \* (u[i][j - 1] - u[i - 1][j]) + 4 \* tau \* (  
 x[i] + h / 2)) / (lam + 1)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=0 левый нижний угол  
def fun11():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[10][i] = u1t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(0, 10):  
 u[i][j] = (lam + 1) \* u[i][j - 1] - lam \* u[i + 1][j - 1]  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=x левый нижний угол  
def fun12():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[10][i] = u1t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(0, 10):  
 u[i][j] = (lam + 1) \* u[i][j - 1] - lam \* u[i + 1][j - 1] + 2 \* tau \* x[i]  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=0 левый верхний угол  
def fun13():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[10][i] = u1t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(9, -1, -1):  
 u[i][j] = (u[i][j - 1] - lam \* u[i + 1][j]) / (1 - lam)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=x левый верхний угол  
def fun14():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[10][i] = u1t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(9, -1, -1):  
 u[i][j] = (u[i][j - 1] - lam \* u[i + 1][j] + 2 \* tau \* x[i]) / (1 - lam)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=0 левый прямоугольник  
def fun15():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[10][i] = u1t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(9, -1, -1):  
 u[i][j] = ((1 + lam) \* (u[i][j - 1] - u[i + 1][j]) + u[i + 1][j - 1] \* (1 - lam)) / (1 - lam)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
# a<0 f=x левый прямоугольник  
def fun16():  
 u = np.zeros((11, 1001))  
 x = np.zeros(11)  
 t = np.zeros(1001)  
 h = 0.1  
 tau = 0.01  
 a = -2  
 for i in range(0, 11):  
 x[i] = (i \* h)  
 u[i][0] = ux0(x[i])  
  
 for i in range(1001):  
 t[i] = i \* tau  
 u[10][i] = u1t(t[i])  
  
 lam = a \* tau / h  
 for j in range(1, 1001):  
  
 for i in range(9, -1, -1):  
 u[i][j] = ((1 + lam) \* (u[i][j - 1] - u[i + 1][j]) + u[i + 1][j - 1] \* (1 - lam) + 4 \* tau \* (  
 x[i] + h / 2)) / (1 - lam)  
  
 ures = np.zeros((11, 1001))  
 xres = np.zeros((11, 1001))  
 tres = np.zeros((11, 1001))  
 for i in range(0, 11):  
  
 for j in range(0, 1001):  
 ures[i][j] = u[i][j]  
 xres[i][j] = x[i]  
 tres[i][j] = t[j]  
  
 fig = plt.figure()  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot\_surface(xres, tres, ures)  
  
 ax.set\_xlabel('X Label')  
 ax.set\_ylabel('T Label')  
 ax.set\_zlabel('U Label')  
  
 plt.show()  
 return  
  
  
fun2()  
fun3()  
fun4()  
fun5()  
fun6()  
fun7()  
fun8()  
fun9()  
fun10()  
fun11()  
fun12()  
fun13()  
fun14()  
fun15()  
fun16()