**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 9**

**Параболические уравнения**

**(Вариант 9)**

*Выполнил студент 3 курса ПМиИ*

*Кондратьев Виталий*

***Цель работы***: усвоить сущность и методы решения ***линейного дифференциального уравнения 2-го порядка параболического типа***.

Численное решение дифференциального уравнения в частных производных предполагает получение двумерной числовой таблицы приближенных значений *Uij* искомой функции *U*(*t,x)* с заданной точностью для некоторых значений аргументов

*xj ∈* [*a*, *b*], *ti ∈* [*c*, *d*]

Численное решение таких дифференциальных уравнений возможно методами конечных разностей.

Погрешность решения, найденного этими методами, оценивается величиной O(*τp,hq*)*,* где *p*, *q* - порядок метода.

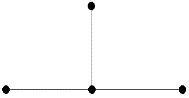
***Задание.***

Решить параболическое уравнение

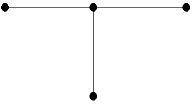


явным методом и неявным методом.

Шаблон для явного метода:



Шаблон для неявного метода:



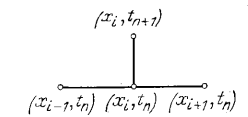
Вывести результаты в виде графиков U(x) для разных значений t от 0 до 10 c шагом 1

Для всех вариантов [*a*, *b*] = [0; 1], [*c*, *d*] = [0; 10], D=1. Погрешность решения 0,01.

Материал стр. 243 и 278 Турчак ЛИ «Основы численных методов»

|  |  |  |
| --- | --- | --- |
| № вар. | Начальные условия | Граничные условия |
| 9 |  |  |

1. ***ЯВНАЯ СХЕМА***



Уравнение теплопроводности:

Введём схему, состоящую из четырёх узлов. Разностное уравнение, построенное по данной схеме:

Разрешая уравнение относительно :

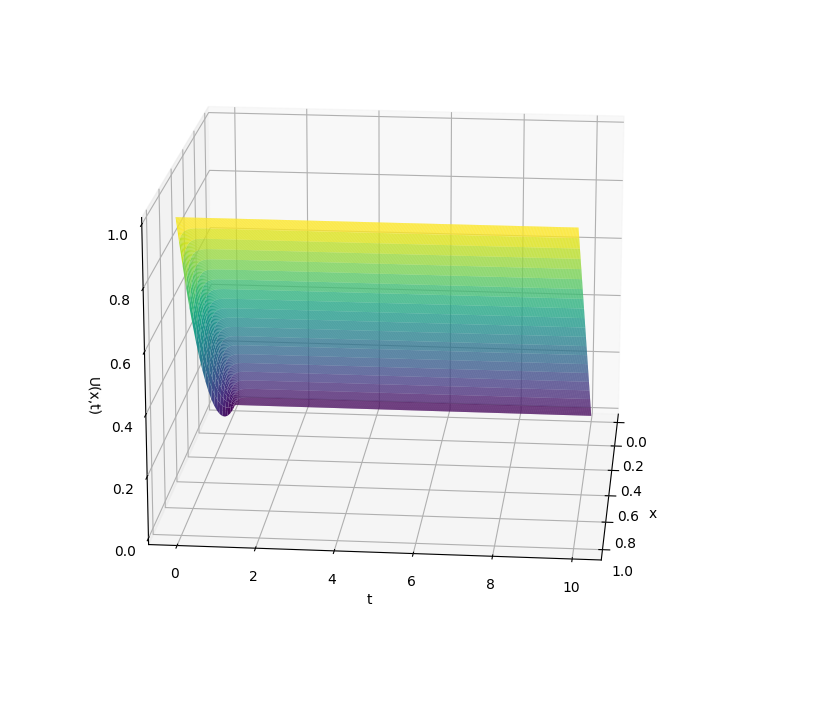
Начальные условия:

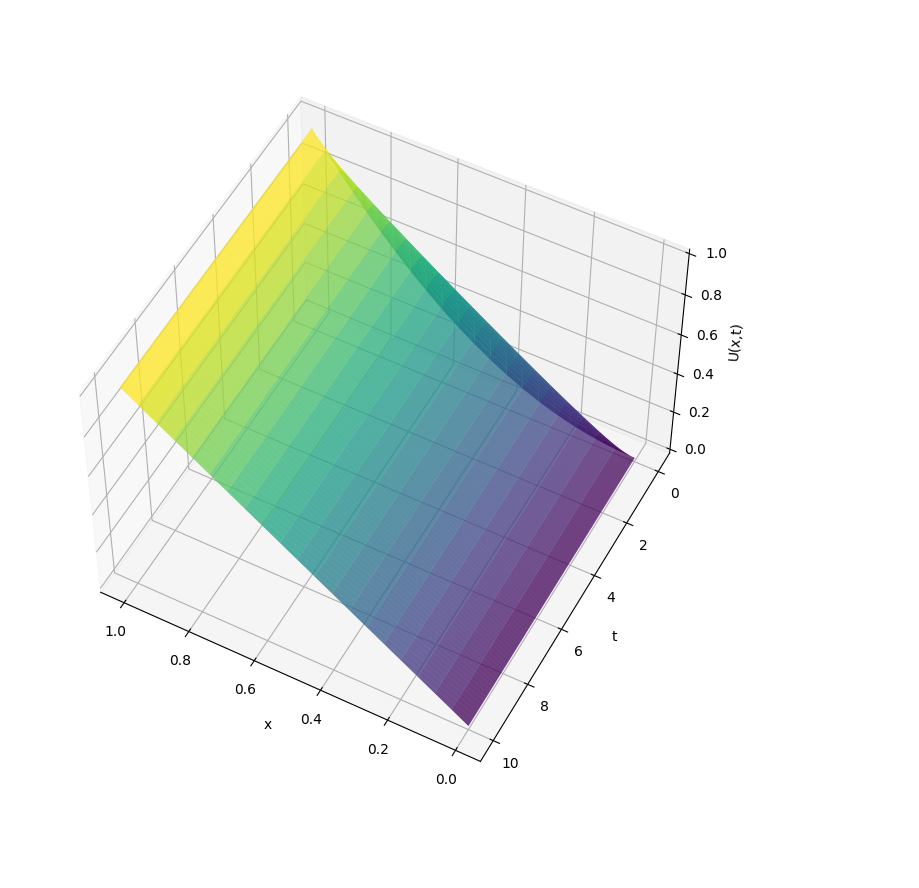
Граничные условия в сеточном виде:

Рассмотренная разностная схема условно устойчива. Необходимое и достаточное условие устойчивости:

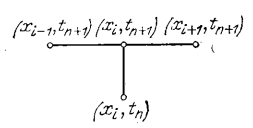
Возьмем в качестве первого шага разбиения: и . Итерации завершаются при выполнении условия:

**Графики:**





***2. НЕЯВНАЯ СХЕМА***



Построим простейшую неявную схему. Разностное уравнение, построенное по данной схеме:

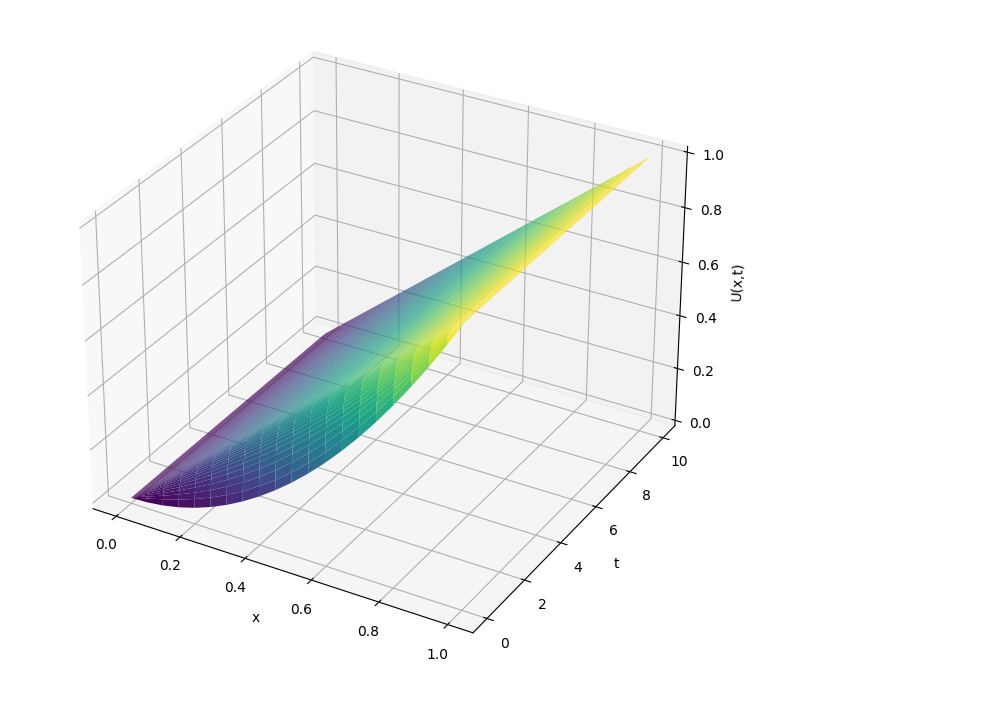
Из этого разностного соотношения можно получить систему уравнений относительно неизвестных значений сеточной функции на слое:

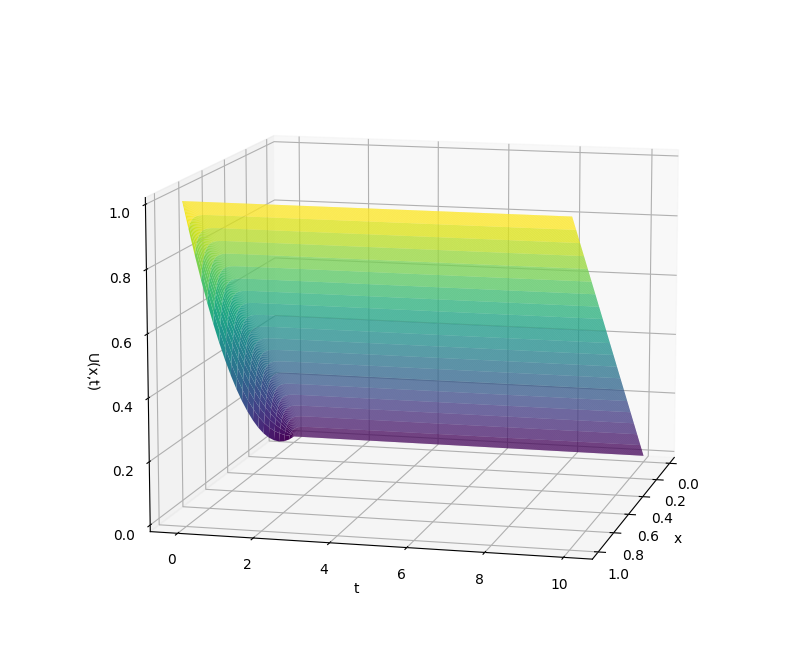
Полученная схема устойчива и сходится со скоростью . Граничные и начальные условия аналогичны явной схеме.

Возьмем в качестве первого шага разбиения: и . Итерации завершаются при выполнении условия:

Систему линейных уравнений решаем методом прогонки.

**Графики:**





**ПРИЛОЖЕНИЕ**

*"""  
Лабораторная работа №9  
Студент ОНК «ИВТ» ВШ КНиИИ направления ПМиИ 3 курса  
Кондратьев Виталий  
Вариант 9  
"""*import numpy  
import matplotlib.pyplot as plt  
  
  
def Ux0(x):  
 return x\*\*2  
  
  
def U0t(t):  
 return 0  
  
  
def Ult(t):  
 return 1  
  
  
h = 0.05  
r = 0.00125  
a = 1  
p = int(1 / h) + 1  
q = int(10 / r) + 1  
l = a \* r / (h \* h)  
  
U = [0] \* p  
for i in range(p):  
 U[i] = [0] \* q  
  
for i in range(0, p):  
 x = h \* i  
 U[i][0] = Ux0(x)  
  
for j in range(1, q):  
 t = r \* j  
 U[0][j] = U0t(t)  
 U[p - 1][j] = Ult(t)  
  
for j in range(0, q - 1):  
 for i in range(1, p - 1):  
 U[i][j + 1] = l \* (U[i + 1][j] + U[i - 1][j]) + (1 - 2 \* l) \* U[i][j]  
  
u, v = numpy.mgrid[0:p, 0:q]  
x = h \* u  
y = r \* v  
z = x - x  
for i in range(0, p):  
 for j in range(0, q):  
 z[i][j] = U[i][j]  
  
fig = plt.figure(figsize=plt.figaspect(0.5))  
axes = fig.add\_subplot(1, 2, 1, projection='3d')  
axes.set\_xlabel("x")  
axes.set\_ylabel("t")  
axes.set\_zlabel("U(x,t)")  
suf = axes.plot\_surface(x, y, z, rstride=1, cstride=15, cmap='viridis', edgecolor='none')  
plt.show()  
  
U = [0] \* p  
for i in range(p):  
 U[i] = [0] \* q  
  
for i in range(0, p):  
 x = h \* i  
 U[i][0] = Ux0(x)  
  
for j in range(1, q):  
 t = r \* j  
 U[0][j] = U0t(t)  
 U[p - 1][j] = Ult(t)  
  
for j in range(0, q - 1):  
 mb = [0] \* p  
 for i in range(0, p - 2):  
 mb[i] = -l  
  
 mc = [0] \* p  
 for i in range(1, p - 1):  
 mc[i] = 1 + 2 \* l  
  
 ma = [0] \* p  
 for i in range(2, p - 1):  
 ma[i] = -l  
  
 mf = [0] \* p  
 for i in range(1, p - 1):  
 mf[i] = U[i][j]  
 mf[1] = mf[1] + l \* U[0][j]  
 mf[p - 2] = mf[p - 2] + l \* U[p - 1][j]  
  
 for i in range(2, p - 1):  
 m = ma[i] / mc[i - 1]  
 mc[i] = mc[i] - m \* mb[i - 1]  
 mf[i] = mf[i] - m \* mf[i - 1]  
 U[p - 2][j + 1] = mf[p - 2] / mc[p - 2]  
  
 for i in range(p - 3, 0, -1):  
 U[i][j + 1] = (mf[i] - mb[i] \* U[i + 1][j + 1]) / mc[i]  
  
u, v = numpy.mgrid[0:p, 0:q]  
x = h \* u  
y = r \* v  
z = x - x  
for i in range(0, p):  
 for j in range(0, q):  
 z[i][j] = U[i][j]  
  
fig = plt.figure(figsize=plt.figaspect(0.5))  
axes = fig.add\_subplot(1, 2, 1, projection='3d')  
axes.set\_xlabel("x")  
axes.set\_ylabel("t")  
axes.set\_zlabel("U(x,t)")  
suf = axes.plot\_surface(x, y, z, rstride=1, cstride=15, cmap='viridis', edgecolor='none')  
plt.show()