Московский авиационный институт

(Национальный исследовательский университет)

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

**Лабораторная работа №8**

«ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ»

Вариант 4

Выполнил: Железнов Д.Е.

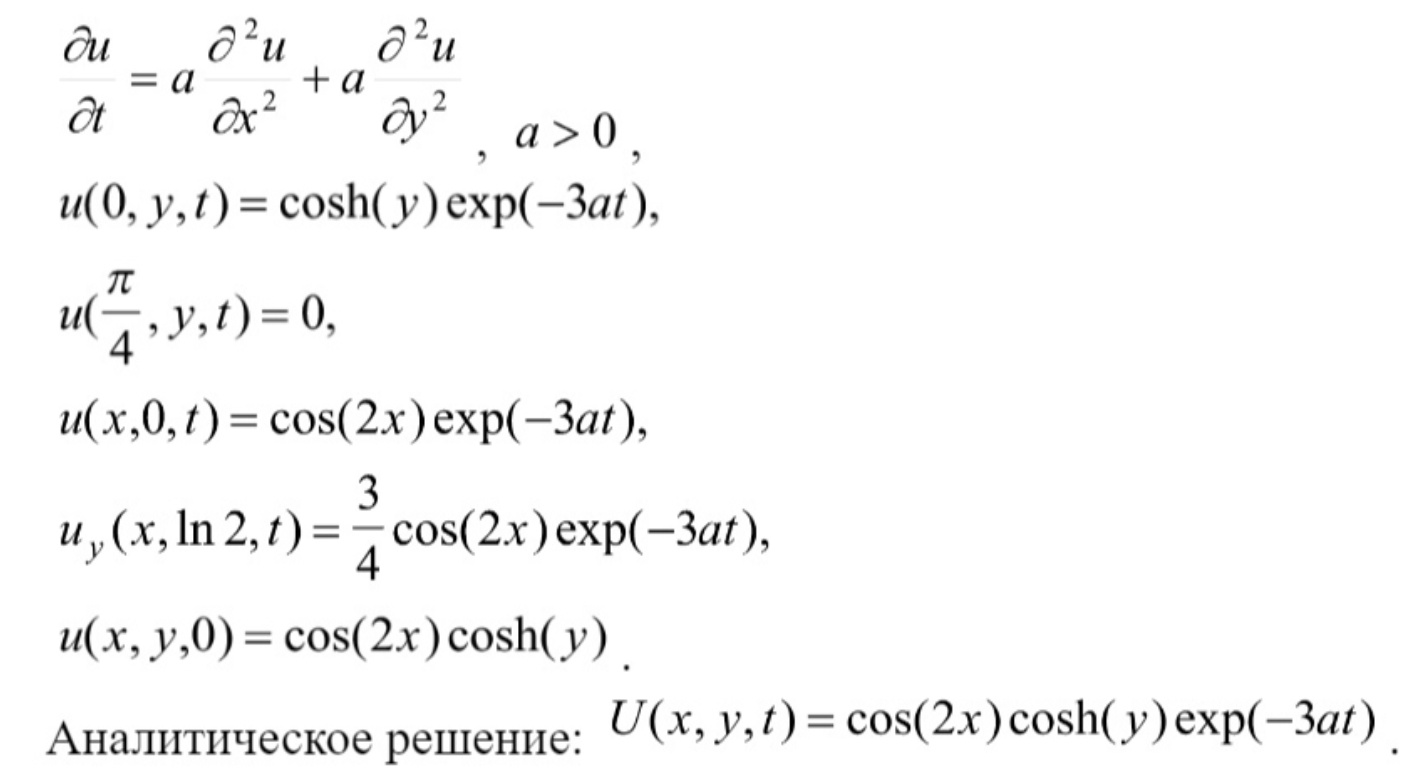
Группа: М8О-409Б-20

Проверил: Пивоваров Д.Е.

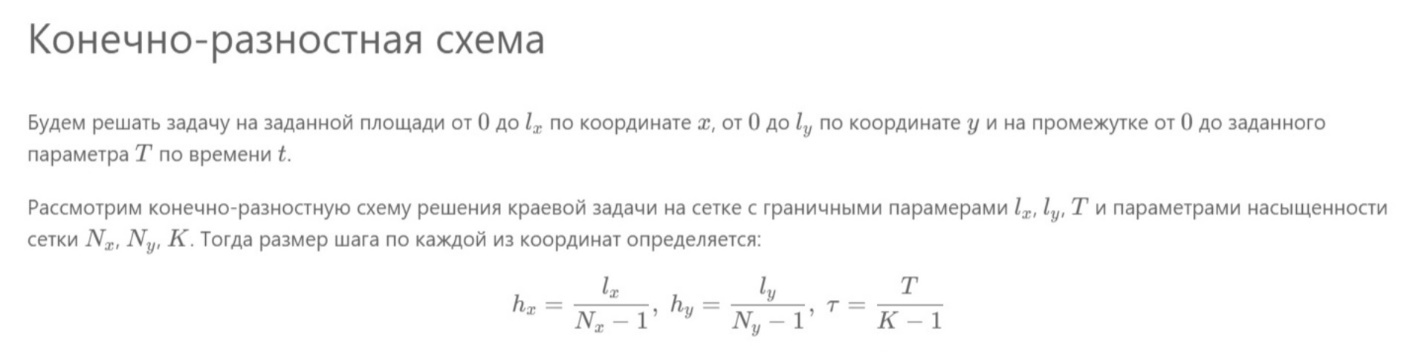
Дата:

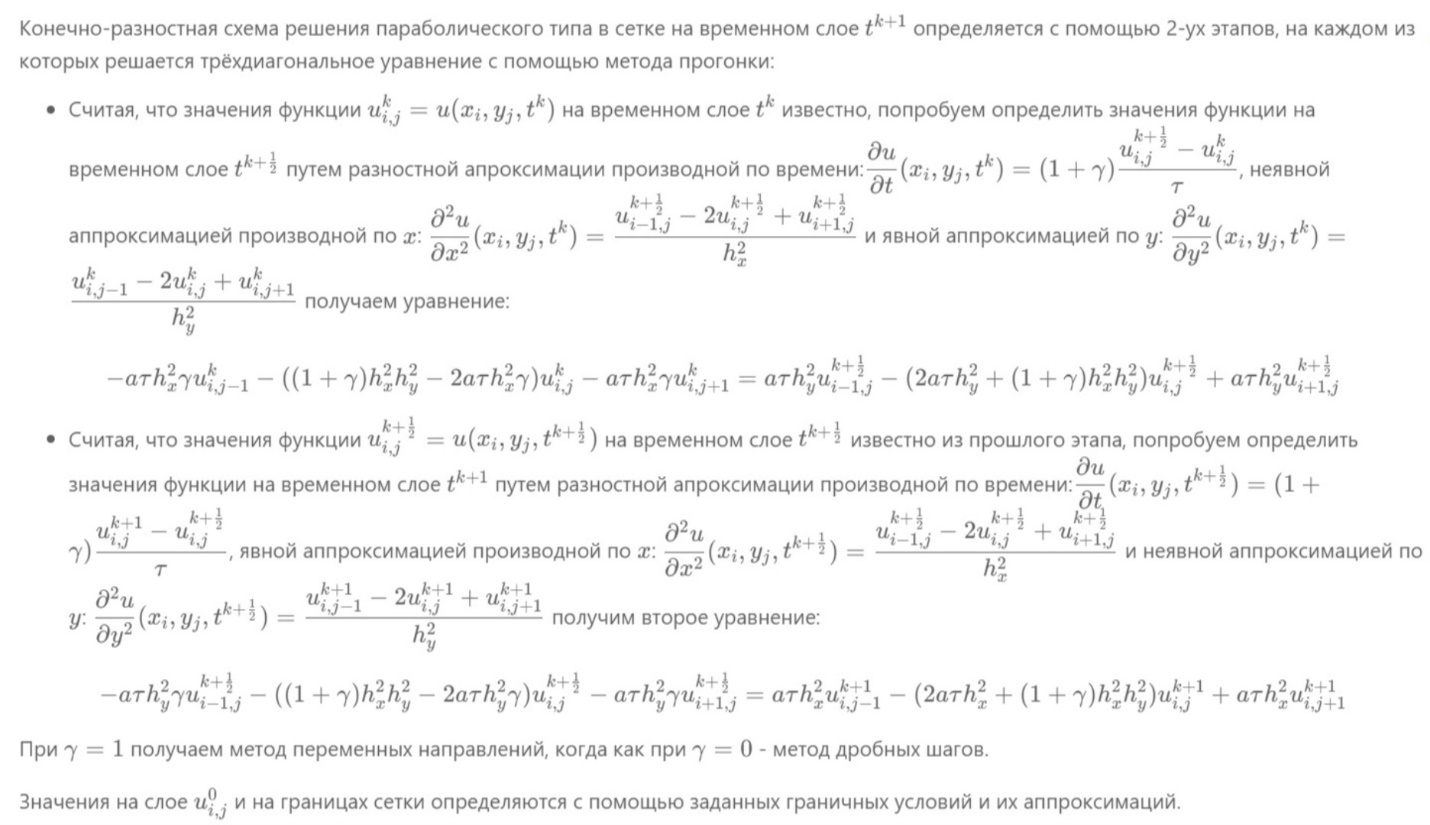
Оценка:

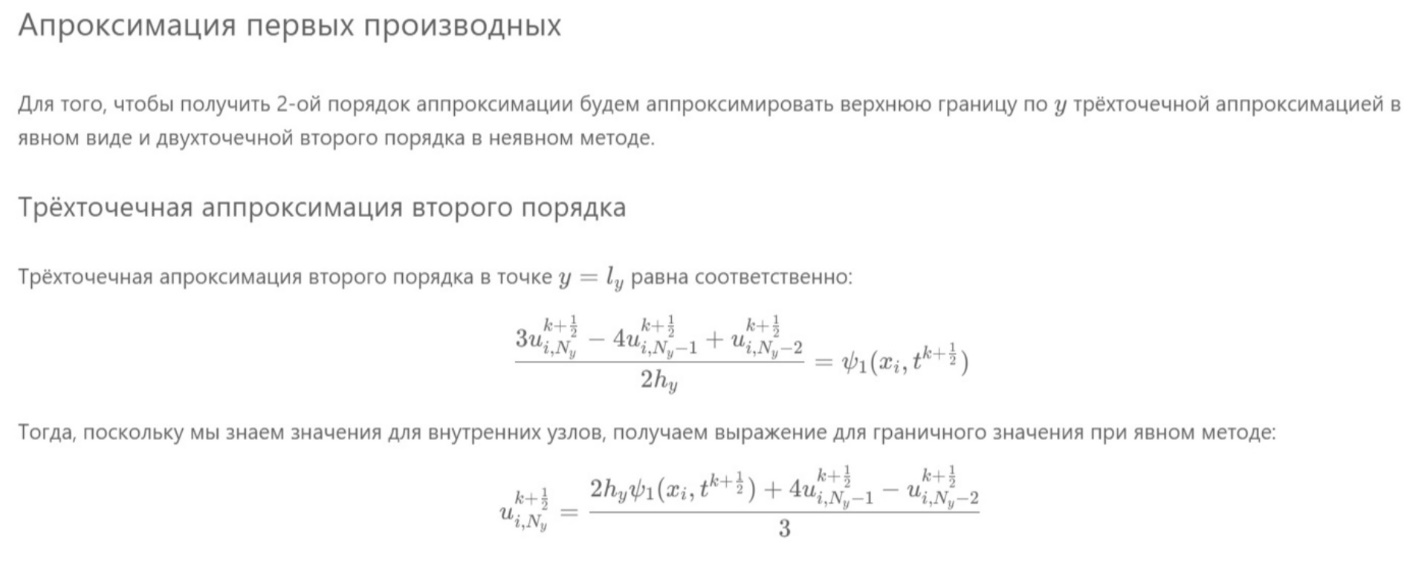
Задание: Используя схемы переменных направлений и дробных шагов, решить двумерную начально-краевую задачу для дифференциального уравнения параболического типа. В различные моменты времени вычислить погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением *u (x, t)*. Исследовать зависимость погрешности от сеточных параметров τ, hx,hy.

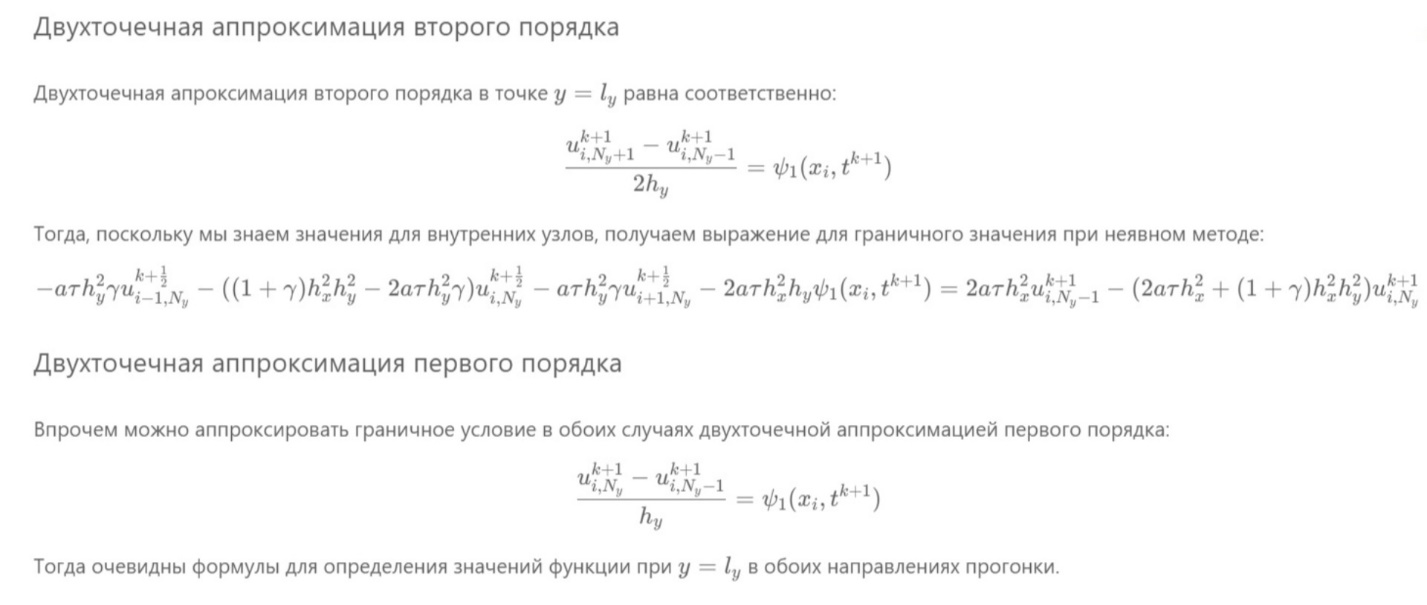


Теоретическая часть:







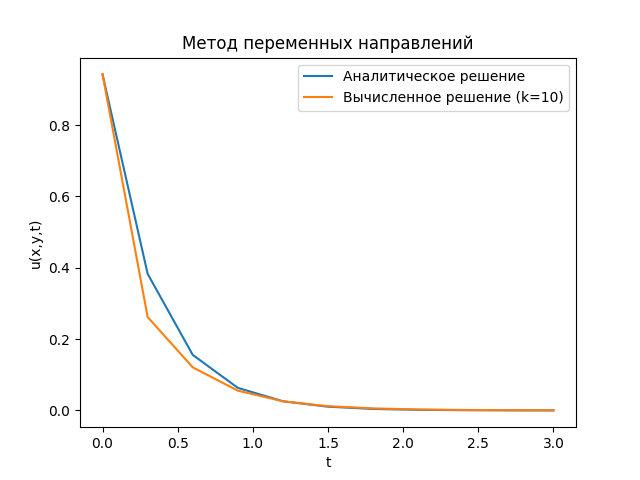


**Код программы:**

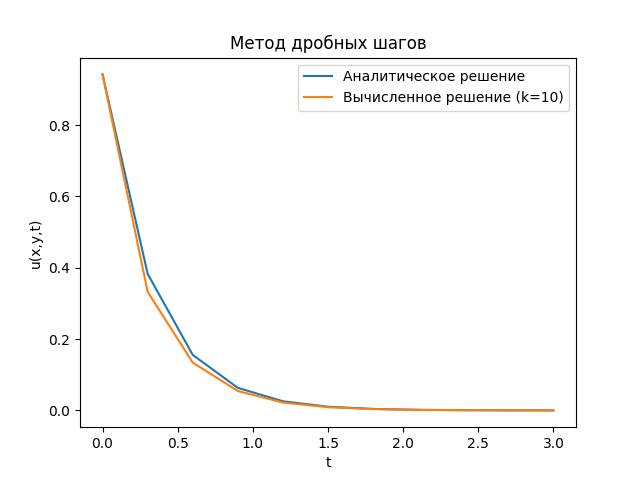
import matplotlib.pyplot as plt  
import numpy as np  
from sympy import \*  
from functions import \*  
import math  
import logging  
  
a = 1  
T = 3  
n = 15  
k = 10  
m = 10  
hx = math.pi/(4\*n)  
hy = math.log(2)/m  
tau = T/k  
  
t\_array = [0] \* (k+1)  
x\_array = [0] \* (n+1)  
y\_array = [0] \* (m+1)  
solution\_array\_t = [0] \* (k+1)  
  
for i in range(k+1):  
 t\_array[i] = tau \* i  
for i in range(n+1):  
 x\_array[i] = hx \* i  
for i in range(m+1):  
 y\_array[i] = hy \* i  
def T\_y\_back(t,y):  
 return math.cosh(y) \* math.exp(-3 \* a \* t)  
def T\_y\_front(t,y):  
 return 0  
def X\_t\_left(x,t):  
 return math.cos(2 \* x) \* math.exp(-3 \* a \* t)  
def X\_t\_right(x,t):  
 return (3/4) \* math.cos(2 \* x) \* math.exp(-3 \* a \* t)  
def X\_y\_bottom(x,y):  
 return math.cos(2 \* x) \* math.cosh(y)  
def solution(x,y,t):  
 return math.cos(2 \* x) \* math.cosh(y) \* math.exp(-3 \* a \* t)  
  
*# сетка*array = [0] \* (n+1)  
for i in range(n+1):  
 array[i] = [0]\*(m+1)  
 for j in range(m+1):  
 array[i][j] = [0] \* (k+1)  
  
  
for i in range(m+1):  
 for j in range(n+1):  
 array[j][i][0] = X\_y\_bottom(j \* hx, i \* hy)  
  
for i in range(m+1):  
 for j in range(k+1):  
 array[0][i][j] = T\_y\_back(tau \* j, hy \* i)  
  
for i in range(n+1):  
 for j in range(k+1):  
 array[i][0][j] = X\_t\_left(i \* hx, j \* tau)  
  
for i in range(n+1):  
 for j in range(k+1):  
 array[i][-1][j] = X\_t\_right(i \* hx, j \* tau)  
  
*#Переменные направления*for iter in range(1,k+1):  
 A = [0] \* (n - 1)  
 for i in range(n - 1):  
 A[i] = [0] \* (n - 1)  
  
 C = [0] \* (m - 1)  
 for i in range(m - 1):  
 C[i] = [0] \* (m - 1)  
  
 for i in range(n - 1):  
 A[i][i] = 1 + ((a \* tau) / (hx\*\*2))  
 count = 0  
 for i in range(1, n - 1):  
 A[i][count] = - (a \* tau) / (2 \* hx\*\*2)  
 count += 1  
 count = 1  
 for i in range(n - 2):  
 A[i][count] = - (a \* tau) / (2 \* hx\*\*2)  
 count += 1  
  
 for i in range(m - 1):  
 C[i][i] = 1 + (a \* tau) / (hy\*\*2)  
 count = 0  
 for i in range(1, m - 1):  
 C[i][count] = - (a \* tau) / (2 \* hy\*\*2)  
 count += 1  
 count = 1  
 for i in range(m - 2):  
 C[i][count] = - (a \* tau) / (2 \* hy\*\*2)  
 count += 1  
  
 semi\_array = [0] \* (n + 1)  
 for i in range(n + 1):  
 semi\_array[i] = [0] \* (m + 1)  
  
 for i in range(n+1):  
 semi\_array[i][0] = X\_t\_left(hx \* i,tau \* ((iter - 1) + 0.5)) *#iter с 1 начинается,проверь!* for i in range(m+1):  
 semi\_array[0][i] = T\_y\_back(tau \* ((iter - 1) + 0.5),i \* hy)  
 for i in range(n+1):  
 semi\_array[i][-1] = X\_t\_right(hx \* i,tau \* ((iter - 1) + 0.5))  
  
 B = [0] \* (n - 1)  
 D = [0] \* (m - 1)  
  
 for j in range(1,m-1):  
 for i in range(1,n-2):  
 B[i] = array[i+1][j][iter-1] + ((a \* tau) / (2 \* hy\*\*2)) \* (array[i+1][j-1][iter-1] - (2 \* array[i+1][j][iter-1]) + array[i+1][j+1][iter-1])  
 B[0] = ((a \* tau) / (2 \* hx\*\*2)) \* semi\_array[0][j] + array[1][j][iter-1] + ((a \* tau) / (2 \* hy\*\*2)) \* (array[1][j-1][iter-1] - (2 \* array[1][j][iter-1]) + array[1][j+1][iter-1])  
 B[-1] = ((a \* tau) / (2 \* hx\*\*2)) \* semi\_array[-1][j] + array[n-1][j][iter-1] + ((a \* tau) / (2 \* hy\*\*2)) \* (array[n-1][j-1][iter-1] - (2 \* array[n-1][j][iter-1]) + array[n-1][j+1][iter-1])  
 x = progonka(A, B, n-1)  
 for i in range(len(x)):  
 semi\_array[i+1][j] = x[i]  
  
 for j in range(1,n-1):  
 for i in range(1,m-2):  
 D[i] = semi\_array[j][i+1] + ((a \* tau) / (2 \* hx\*\*2)) \* (semi\_array[j+1][i+1] - 2 \* semi\_array[j][i+1] + semi\_array[j-1][i+1])  
 D[0] = ((a \* tau) / (2 \* hy\*\*2)) \* array[j][0][iter] + semi\_array[j][1] + ((a \* tau) / (2 \* hx\*\*2)) \* (semi\_array[j+1][1] - 2 \* semi\_array[j][1] + semi\_array[j-1][1])  
 D[-1] = ((a \* tau) / (2 \* hy\*\*2)) \* array[j][-1][iter] + semi\_array[j][m-1] + ((a \* tau) / (2 \* hx\*\*2)) \* (semi\_array[j+1][m-1] - 2 \* semi\_array[j][m-1] + semi\_array[j-1][m-1])  
 y = progonka(C, D, m-1)  
 for i in range(len(y)):  
 array[j][i+1][iter] = y[i]  
  
  
ax = plt.axes()  
ax.set\_xlabel("t")  
ax.set\_ylabel("u(x,y,t)")  
for i in range(k+1):  
 solution\_array\_t[i] = solution(x\_array[5],y\_array[6],t\_array[i])  
print(solution\_array\_t)  
plt.plot(t\_array, solution\_array\_t, label="Аналитическое решение")  
plt.plot(t\_array, array[5][6], label=f"Вычисленное решение (k={k})")  
plt.legend(loc='best')  
plt.title("Метод переменных направлений")  
plt.show()  
  
  
*#дробные шаги*for iter in range(1,k+1):  
 A = [0] \* (n - 1)  
 for i in range(n - 1):  
 A[i] = [0] \* (n - 1)  
  
 C = [0] \* (m - 1)  
 for i in range(m - 1):  
 C[i] = [0] \* (m - 1)  
  
 for i in range(n - 1):  
 A[i][i] = 1 + ((2 \* a \* tau) / (hx\*\*2))  
 count = 0  
 for i in range(1, n - 1):  
 A[i][count] = - (a \* tau) / (hx\*\*2)  
 count += 1  
 count = 1  
 for i in range(n - 2):  
 A[i][count] = - (a \* tau) / (hx\*\*2)  
 count += 1  
  
 for i in range(m - 1):  
 C[i][i] = 1 + (2 \* a \* tau) / (hy\*\*2)  
 count = 0  
 for i in range(1, m - 1):  
 C[i][count] = - (a \* tau) / (hy\*\*2)  
 count += 1  
 count = 1  
 for i in range(m - 2):  
 C[i][count] = - (a \* tau) / (hy\*\*2)  
 count += 1  
  
 semi\_array = [0] \* (n + 1)  
 for i in range(n + 1):  
 semi\_array[i] = [0] \* (m + 1)  
  
 for i in range(n+1):  
 semi\_array[i][0] = X\_t\_left(hx \* i,tau \* ((iter - 1) + 0.5)) *#iter с 1 начинается,проверь!* for i in range(m+1):  
 semi\_array[0][i] = T\_y\_back(tau \* ((iter - 1) + 0.5),i \* hy)  
 for i in range(n+1):  
 semi\_array[i][-1] = X\_t\_right(hx \* i,tau \* ((iter - 1) + 0.5))  
  
 B = [0] \* (n - 1)  
 D = [0] \* (m - 1)  
  
 for j in range(1,m-1):  
 for i in range(1,n-2):  
 B[i] = array[i+1][j][iter-1]  
 B[0] = ((a \* tau) / (hx\*\*2)) \* semi\_array[0][j] + array[1][j][iter-1]  
 B[-1] = ((a \* tau) / (hx\*\*2)) \* semi\_array[-1][j] + array[n-1][j][iter-1]  
 x = progonka(A, B, n-1)  
 for i in range(len(x)):  
 semi\_array[i+1][j] = x[i]  
  
 for j in range(1,n-1):  
 for i in range(1,m-2):  
 D[i] = semi\_array[j][i+1]  
 D[0] = ((a \* tau) / (hy\*\*2)) \* array[j][0][iter] + semi\_array[j][1]  
 D[-1] = ((a \* tau) / (hy\*\*2)) \* array[j][-1][iter] + semi\_array[j][m-1]  
 y = progonka(C, D, m-1)  
 for i in range(len(y)):  
 array[j][i+1][iter] = y[i]  
  
ax = plt.axes()  
ax.set\_xlabel("t")  
ax.set\_ylabel("u(x,y,t)")  
for i in range(k+1):  
 solution\_array\_t[i] = solution(x\_array[5],y\_array[6],t\_array[i])  
print(solution\_array\_t)  
plt.plot(t\_array, solution\_array\_t, label="Аналитическое решение")  
plt.plot(t\_array, array[5][6], label=f"Вычисленное решение (k={k})")  
plt.legend(loc='best')  
plt.title("Метод дробных шагов")  
plt.show()

**Результат:**

[0.9420063685734295, 0.38299120874437664, 0.155712605422842, 0.06330802100408704, 0.025739120558482372, 0.010464745487484434, 0.004254648012118519, 0.0017298107946030405, 0.0007032885861773745, 0.0002859358011815821, 0.00011625282139405139]



[0.9420063685734295, 0.38299120874437664, 0.155712605422842, 0.06330802100408704, 0.025739120558482372, 0.010464745487484434, 0.004254648012118519, 0.0017298107946030405, 0.0007032885861773745, 0.0002859358011815821, 0.00011625282139405139]



**Вывод:**

В ходе лабораторной работы, используя *схемы переменных направлений* и *дробных шагов*, решил двумерную начально-краевую задачу для дифференциального уравнения параболического типа. В различные моменты времени вычислил погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением *u (x, t)*. Исследовал зависимость погрешности от сеточных параметров τ, hx,hy.