Федеральное государственное бюджетное
образовательное учреждение высшего
профессионального образования «Московский
государственный университет имени М.В.Ломоносова»

Задание №2, вариант №16 по курсу "Основы математического моделирования"

Михайлов Павел Олегович
студент 316 группы
Кафедра физики полупроводников и криоэлектроники

Преподаватель:
Буткарев Иван Андреевич

Физический факультет, 2021 г.

Содержание

Постановка задачи

Используя схему бегущего счета и итерационные методы, решить задачу:

$$\begin{cases} \dfrac{\partial U}{\partial t} - \dfrac{\partial U}{\partial x} = 1 - e^{t+U}, \quad -1 \leq x < 0, \ t > 0 \\ U(x,0) = \ln\left(\dfrac{2}{1-2x}\right), \\ U(0,t) = \ln\left(\dfrac{2}{e^t - 2t\, e^{-t}}\right) \end{cases} \tag{1}$$

Аналитическое решение

1. Исследование характеристик

Сразу можно отметить, что при значениях $x = 0$, $t = 0$ из $U(0,0) = ln\,(2)$ для ГУ и НУ. Таким образом, начальное и граничное условия согласованы.
Составим уравнение характеристик и проверим, будут ли пересекаться характеристики:

$$dt = -dx = \frac{dU}{1 - e^{t+U}} \tag{2}$$

Проинтегрировав это уравнение, приходим к семейству характеристик:

$$x + t = C_1 \tag{3}$$

где C - произвольная постоянная. Таким образом, видно, что семейство характеристик представляет собой множество параллельных прямых. Графическое представление семейства характеристик представлено на рис.1.

Поскольку пересечений нет, то в каждой точке рассматриваемой области решение единственно.

2. Нахождение аналитического решения

Для нахождения аналитического решения нужно найти второй первый интеграл.

$$dt = \frac{dU}{1 - e^{t+U}} \tag{4}$$

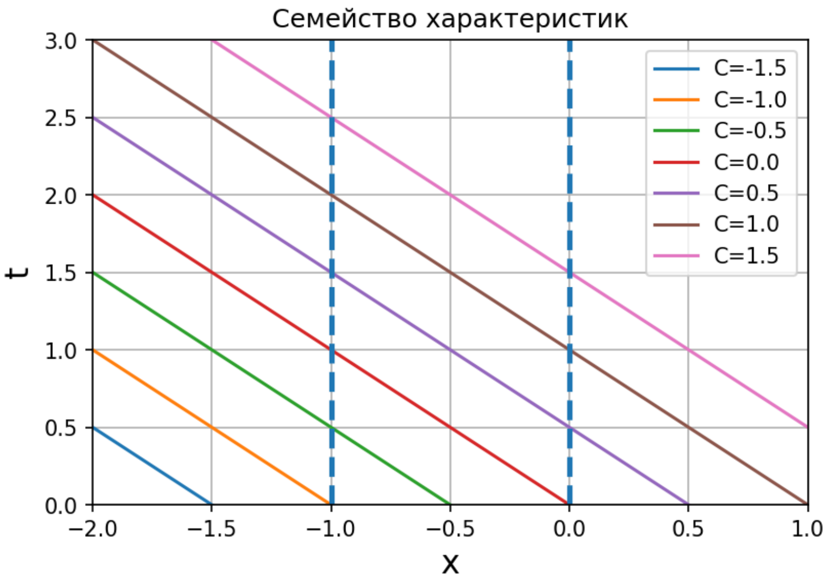

Рис. 1: График семейства характеристик $x + t = C$

Для решения этого дифференциального уравнения можно сделать замену, например, такую:

$$y = e^{t+U} \;\Rightarrow\; dU = d\left(-t + ln\left(y\right)\right) = -dt + \frac{dy}{y} \tag{5}$$

Тогда получим:

$$\begin{gathered} dt = \frac{-dt + \frac{dy}{y}}{1-y} \;\Rightarrow \\ \Rightarrow\; dt = \frac{-dt}{1-y} + \frac{dy}{y\left(1-y\right)} \end{gathered} \tag{6}$$

После преобразований:

$$dt = \frac{dy}{y\left(2-y\right)} = dy\left(\frac{1}{y} + \frac{1}{2-y}\right)\frac{1}{2} \tag{7}$$

Проинтегрировав, приходим к выражению:

$$\begin{gathered} 2t + C_2 = ln(y) - ln(2-y) \;\Rightarrow\; \frac{y}{2-y} = 2t + C_2 \;\Rightarrow \\ \Rightarrow\; y = \frac{2C_2e^{2t}}{1 + C_2e^{2t}} = e^{t+U} \;\Rightarrow\; U = ln\left(\frac{2C_2e^{t}}{1 + C_2e^{2t}}\right) \end{gathered} \tag{8}$$

Запишем граничные условия для характеристик:

$$\begin{cases} t = C_1 > 0 \\ U = ln\left(\dfrac{2C_2e^{C_1}}{1+C_2e^{2C_1}}\right) = ln\left(\dfrac{2}{e^{C_1} - 2C_1e^{-C_1}}\right) \end{cases} \tag{9}$$

Таким образом, при $C_1 > 0$ из нижнего соотношения получаем:

$$\frac{C_2}{1+C_2e^{C_1}} = \frac{1}{e^{2C_1} - 2C_1e^{C_1}} \;\Rightarrow\; C_2 = -\frac{1}{2C_1} \tag{10}$$

Аналогично запишем начальные условия:

$$\begin{cases} x = C_1 \in [-1, 0] \\ U = ln\left(\dfrac{2C_2}{1+C_2}\right) = ln\left(\dfrac{2}{1-2C_1}\right) \end{cases} \tag{11}$$

При $C_1 \in [-1, 0]$ из нижнего соотношения получаем:

$$\frac{C_2}{1+C_2} = \frac{1}{1-2C_1} \;\Rightarrow\; C_2 = -\frac{1}{2C_1} \tag{12}$$

Таким образом, можно сделать вывод, что:

$$C_2 = -\frac{1}{2C_1},\;\; C_1 \in [-1,\; +\infty) \tag{13}$$

то есть во всей рассматриваемой области. Поскольку в самой области верно, что $C_1 = x + t$, то аналитическое решение поставленной задачи имеет вид:

$$U = ln\left(\frac{-2\frac{1}{2(x+t)}e^t}{1 - \frac{1}{2(x+t)}e^{2t}}\right),\;\; -1 \le x < 0,\; t > 0 \tag{14}$$

Или это выражение можно записать в виде, из которого сразу можно видеть соответсвие данного решения граничным и начальным условиям задачи:

$$U = ln\left(\frac{2}{e^t - 2(x+t)e^{-t}}\right),\;\; -1 \le x < 0,\; t > 0 \tag{15}$$

Численное решение

1. Методика численного решения

Введем равномерную сетку в исследуемой области:

$$\omega = \left\{x_n = nh, t_k = k\tau \ , n = \overline{0, N_x} \ k = \overline{0, N_t}\right\} \tag{16}$$

где:

$$h = \frac{1}{N_x} \ ; \ \tau = \frac{t_2}{N_t} \text{ , где } t_2 \text{ - конечное время в нашей области} \tag{17}$$

Введем следующие обозначения:

$$\begin{aligned} y_n &= U\left(x_n, t_k\right) \\ \hat{y}_n &= U\left(x_n, t_{k+1}\right) \end{aligned} \tag{18}$$

Заданное уравнение переноса является дифференциальным уравнением в частных производных первого порядка. Поэтому для его решения необходим шаблон перехода, в котором задействованы два слоя по координате и два слоя по времени. Существует несколько вариантов выбора такого шаблона. В данной работе был использован шаблон, представленный на рис.2. Запишем конечно-разностную аппроксимацию заданного уравнения переноса с использованием выбранного шаблона:

$$\frac{\hat{y}_{n+1} - y_{n+1}}{\tau} - \frac{\hat{y}_{n+1} - \hat{y}_n}{h} = 1 - e^{t_{k+1}} + \hat{y}_{n+1} \ , \ n = \overline{0, N_x - 1} \ , k = \overline{0, N_t - 1} \tag{19}$$

Таким образом, мы приходим к разностной схеме:

$$\begin{cases} \dfrac{\hat{y}_{n+1} - y_{n+1}}{\tau} - \dfrac{\hat{y}_{n+1} - \hat{y}_n}{h} = 1 - e^{t_{k+1}} + \hat{y}_{n+1} \ , \ n = \overline{0, N_x - 1} \ , k = \overline{0, N_t - 1} \\ y_n^0 = ln\left(\dfrac{2}{1 - 2x_n}\right) \ n = \overline{0, N_x} \text{ , НУ} \\ y_0^k = ln\left(\dfrac{2}{e^{t_k} - 2t_k \ e^{-t_k}}\right) \ k = \overline{0, N_t} \text{ , ГУ} \end{cases} \tag{20}$$

Для решения данного уравнения будет использоваться метод Ньютона. Суть этого метода можно изложить в следующем виде.

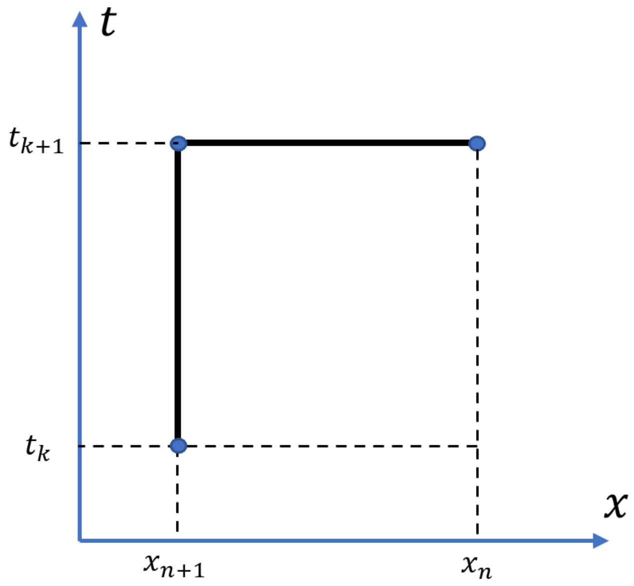

Рис. 2: Шаблон, примененный в данной задаче. Обратите внимание, что шаблон записан для отрицательных x, которые и рассматриваются в нашей задаче.

Запишем наше уравнение в виде: $f(\hat{y}_{n+1}) = 0$. Решение происходит итерационно. На самом первом шаге нам известно некоторое начальное значение (приближение) y^0. Далее по этому значению вычисляются значения всех следующих приближений как:

$$y^{m+1} = y^m - \frac{f\left(y^m\right)}{f'\left(y^m\right)}\ m = 0, 1, ... \tag{21}$$

Условие прекращения итерационного процесса выбирается самим исследователем. Можно выбрать такое условие:

$$\left|y^{m+1} - y^m\right| < \varepsilon\ ; m = 0, 1, ... \tag{22}$$

где ε - значение допустимой ошибки вычислений. Распишем подробнее функции $f(\hat{y}_n)$ и $f'(\hat{y}_n)$ в виде, в котором они применены в этой задаче:

$$\begin{aligned} f(\hat{y}_{n+1}) &= \frac{\hat{y}_{n+1} - y_{n+1}}{\tau} - \frac{\hat{y}_{n+1} - \hat{y}_n}{h} + e^{t_{k+1}} + \hat{y}_{n+1} - 1 = 0\ , \\ f'(\hat{y}_{n+1}) &= \frac{1}{\tau} - \frac{1}{h} + e^{t_{k+1}} + \hat{y}_{n+1} \end{aligned} \tag{23}$$

Таким образом, в данной задаче в качестве y^m выбираются значения $\hat{y}_{n+1}$ и схема бегущего счёта в данной задаче состоит в последовательном вычислении значении функции y во всех узлах сетки.

При этом можно заметить, что в написанных выше уравнениях нам известны y_{n+1} и $\hat{y}_n$ как значения на предыдущем шаге или как начальные и граничные условия. Таким образом, происходит расчёт значений y во всех узлах сетки.

2. Порядок аппроксимации

Найдем порядки аппроксимации по τ, h. Рассмотрим действие оператора производной по t на непрерывную функцию $U(x)$:

$$\begin{aligned} L_\tau U(x,t) &= \frac{U(x,t+\tau) - U(x,t)}{\tau} = \\ &= \frac{1}{\tau}\left(U(x,t) + \tau\frac{\partial U(x,t)}{\partial t} + \frac{\tau^2}{2}\frac{\partial^2 U(x,t)}{\partial t^2} - U(x,t)\right) = \\ &= \frac{\partial U(x,t)}{\partial t} + O(\tau) \end{aligned} \tag{24}$$

Абсолютно аналогичные выкладки можно проделать и для оператора производной по x. Таким образом, схема имеет первый порядок аппроксимации по координате x и времени t.

3. Исследование на устойчивость

Спектральным методом Неймана исследуем условие устойчивости схемы 20. Будем рассматривать разностную задачу с однородным уравнением (заменим y на V_n^k для удобства):

$$\begin{cases} \dfrac{V_{n+1}^{k+1} - V_{n+1}^{k}}{\tau} - \dfrac{V_{n+1}^{k+1} - V_{n}^{k+1}}{h} = 0\ ,\ n = \overline{0, N_x - 1}\ , k = \overline{0, N_t - 1} \\ V_n^0 = e^{i\omega n}\ n = \overline{0, N_x} \end{cases} \tag{25}$$

Будем искать решение этой системы в виде:

$$V_n^k = \lambda^k e^{i\omega n} \tag{26}$$

Такой вид решения подставляется в разностное уравнение, далее это уравнение сокращается на V_n^k и экспоненты. Приходим к уравнению:

$$\frac{\lambda - 1}{\tau} - \lambda\frac{1 - e^{-i\omega}}{h} = 0 \tag{27}$$

Отсюда можно получить, что:

$$\lambda = \frac{1}{1 - \frac{\tau}{h}\left(1 - e^{-i\omega}\right)} \tag{28}$$

Вычислим модуль этой величины

$$\begin{aligned} |\lambda| &= \frac{1}{\left|1 - \frac{\tau}{h}\left(1 - e^{-i\omega}\right)\right|} = \frac{1}{\left|1 - \frac{\tau}{h}\left(1 - cos\left(\omega\right) + isin\left(\omega\right)\right)\right|} = \\ &= \frac{1}{\left|1 - \frac{\tau}{h}\Big(1 - cos\left(\omega\right)\Big) - i\frac{\tau}{h}sin\left(\omega\right)\Big)\right|} = \\ &= \frac{1}{\sqrt{\left(\frac{\tau}{h}\right)^2 sin^2\left(\omega\right) + \left(1 - \frac{\tau}{h}\left(1 - cos\left(\omega\right)\right)\right)^2}} = \\ &= \frac{1}{\sqrt{\left(\frac{\tau}{h}\right)^2 sin^2\left(\omega\right) + \left(1 + \frac{\tau}{|h|}\left(1 - cos\left(\omega\right)\right)\right)^2}} \end{aligned} \tag{29}$$

Проанализируем полученное соотношение. Можно заметить, что:

$$\left(1 - cos\left(\omega\right)\right) \geq 0 \;\; \forall\omega \tag{30}$$

И поэтому:

$$|\lambda| \leq 1 \tag{31}$$

Таким образом, наша схема является безусловно устойчивой.
Также существует геометрическая интерпретация устойчивости схемы бегущего счёта. Суть этой интерпретации может быть изложена следующим образом.
Рассмотрим уравнение переноса, которое мы решаем:

$$\frac{\partial U}{\partial t} - \frac{\partial U}{\partial x} = 1 - e^{t+U}\;,\; -1 \leq x < 0,\; t > 0 \tag{32}$$

Характеристики данного уравнения:

$$x + t = C \tag{33}$$

Схема будет устойчивой, если характеристика, проведенная через точку, в которой решение ещё не известно (точка $(n+1, k+1)$) будет пересекать прямую, соединяющую те точки, в которых решение

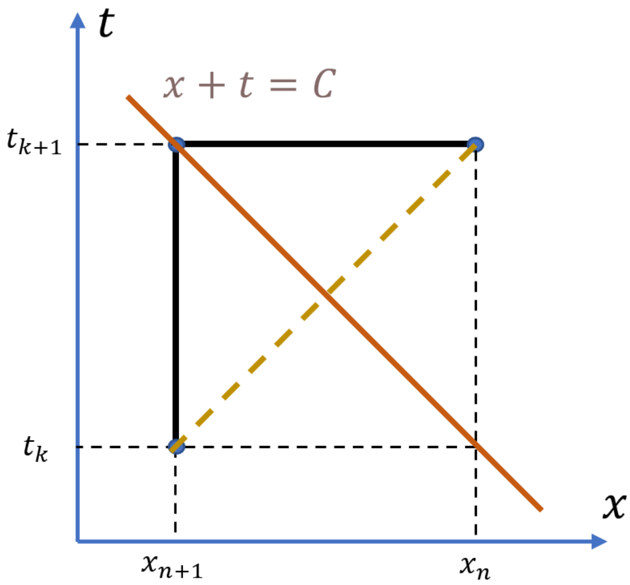

Рис. 3: Геометрическая интерпретация безусловной усточивости. Показано пересечение характеристики и прямой, соединяющей два узла с известными значениями сеточной функции.

уже известно (точки $(n, k+1)$ и $(n+1, k)$). Это схематически показано на рис.3.
Видно, что характеристика пересекает соединительную прямую между узлами с известными значениями сеточной функции. Поэтому схема является устойчивой.

Графики

1. Графики численного и аналитического решений, погрешности численного решения

2. Зависимость погрешности от шага сетки в выбранном узле

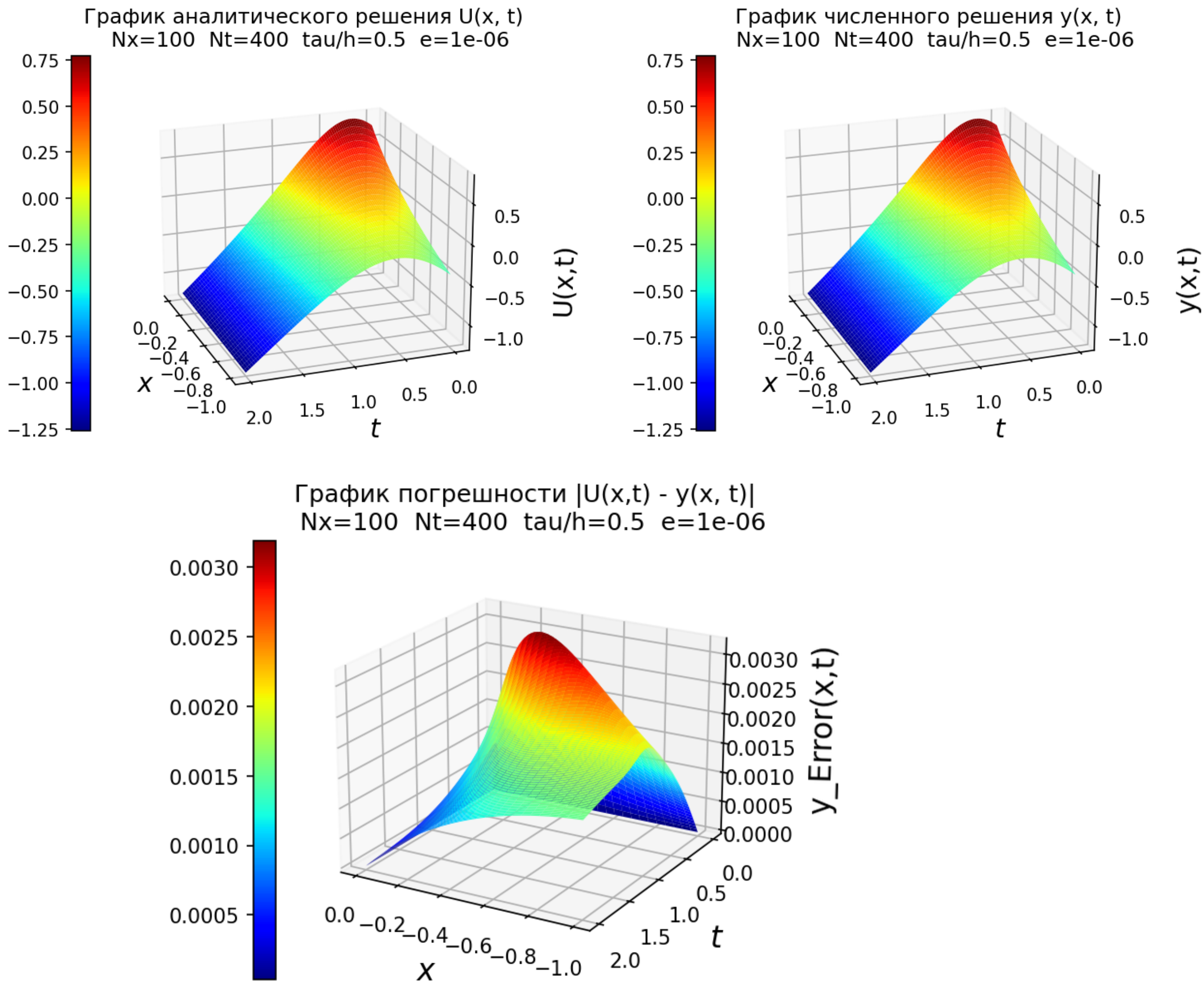

Рис. 4: Графики аналитического и численного решений, погрешности.

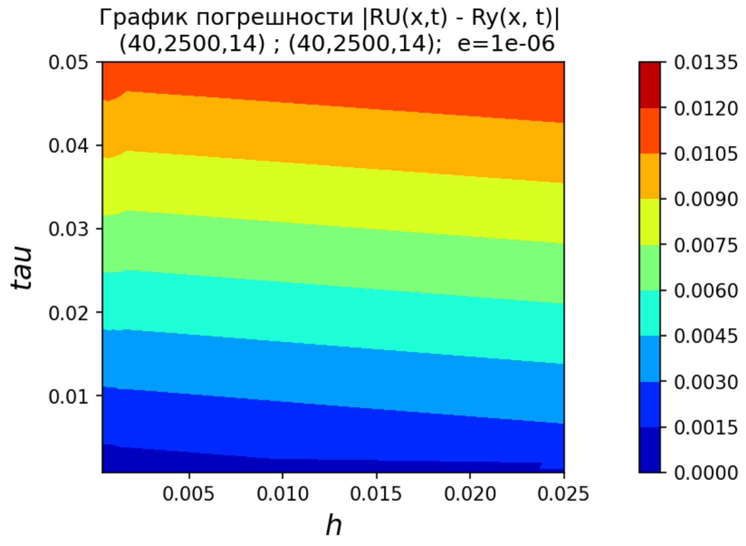

Рис. 5: Зависимость погрешности в узле (Nx-1,Nt-1) от величин шагов по времени и координате.

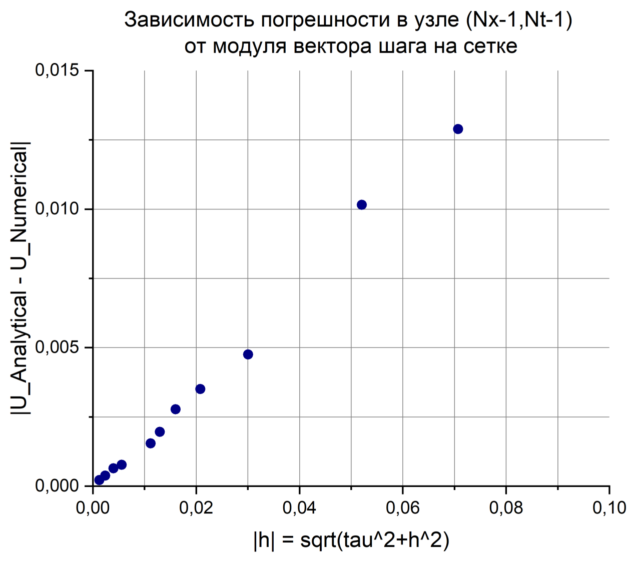

Рис. 6: Зависимость погрешности в узле (Nx-1,Nt-1) от модуля вектора шага сетки

Код программы

```
# -*- coding: utf-8 -*-
"""
Created on Sat Apr 10 22:47:19 2021

@author: PolМихайлов( Павел 316 группа, задача 2 вариант 16)


"""

from matplotlib import cm
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np
import pylab



"1. Построение характеристик для задачи"
x1, x2 = -2, 1
t1, t2 = 0, 2
NC=2


plt.ylim(t1,t2)
plt.xlim(x1,x2)
plt.ylabel('t', size = 15)
plt.xlabel('x', size = 15)
plt.grid(True)
plt.title('Семейство характеристик')
for C in range(-NC-1,NC+2,1):
    plt.plot([x1,x2], [-x1 + C/NC, -x2 + C/NC], label='C='+str(round(C/NC,2)))
plt.axvline(x=-1.0, linewidth = 2.5, linestyle = '--')
plt.axvline(x=0.0, linewidth = 2.5, linestyle = '--')
plt.legend()
mpl.rcParams['figure.dpi'] = 150
plt.show()



"2. Зададим число шагов по временной и пространственной координатам"
x1, x2 = -1, 0
t1, t2 = 0, 2
Nx = 100
Nt = 400
e = 0.000001 #Погрешность для метода Ньютона

h = -(x2 - x1)/Nx
tau = (t2 - t1)/Nt
x = np.linspace(x2, x1, Nx)
t = np.linspace(t1, t2, Nt)


```

```

"3. Зададим массив для нашей сеточной функции, определим граничныеначальные/ условия"
y = np.zeros((Nx, Nt))
for n in range(0,Nx): #Начальное условие
    y[n][0] = np.log(2/(1 - 2*x[n]))

for k in range(0,Nt): #Граничное условие
    y[0][k] = np.log(2/(np.exp(t[k]) - 2*t[k]*np.exp(-t[k])))



"4. Задание функций f и f'  "
def f(n,k):
    return 1/tau * (y[n+1][k+1] - y[n+1][k]) - 1/h * (y[n+1][k+1] - y[n][k+1]) + np.exp(t[k+1] + y[n+1][k+1]) - 1

def fd(n,k):
    return 1/tau - 1/h + np.exp(t[k+1] + y[n+1][k+1])



"5. Реализация метода Ньютона"
for n in range(0,Nx-1):
    for k in range(0,Nt-1):
        y[n+1][k+1] = 0.5*(y[n+1][k] + y[n][k+1]) #Приближение для следующего шага
        while(np.abs(f(n,k) / fd(n,k)) > e):
            y[n+1][k+1] = y[n+1][k+1] - f(n,k) / fd(n,k)



"6. Аналитическое решение"
U = np.zeros((Nx,Nt))
for n in range(0,Nx):
    for k in range(0,Nt):
        U[n][k] = np.log(2/(np.exp(t[k]) - 2*(x[n]+t[k])*np.exp(-t[k])))



"7. Построение аналитического решения"
T,X = np.meshgrid(t,x)
fig = plt.figure()
axes = fig.add_subplot(111, projection='3d')

surf = axes.plot_surface(X, T, U[:,:], rstride=4, cstride=4, cmap = cm.jet)
pylab.xlabel('$x$', size=15, labelpad = 5)
plt.xticks(rotation = 0)
pylab.ylabel('$t$', size=15, labelpad = 5)
plt.yticks(rotation = 0)

axes.set_zlabel('U(x,t)', size = 15, labelpad = 10)

axes.view_init(20, 160) #поворот графика
```

```
fig.colorbar(surf, ax = [axes], pad = 0.05, location='left')
plt.title('График аналитического решения U(x, t) \n Nx=' +str(Nx) + ' Nt='
          +str(Nt)+ ' tau/h=' +str(np.abs(tau/h))+ ' e=' +str(e))
plt.show()



"8. Построение численного решения"
T,X = np.meshgrid(t,x)
fig = plt.figure()
axes = fig.add_subplot(111, projection='3d')

surf = axes.plot_surface(X, T, y[:][:], rstride=4, cstride=4, cmap = cm.jet)
pylab.xlabel('$x$', size=15, labelpad = 5)
plt.xticks(rotation = 0)
pylab.ylabel('$t$', size=15, labelpad = 5)
plt.yticks(rotation = 0)

axes.set_zlabel('y(x,t)', size = 15, labelpad = 10)


axes.view_init(20, 155) #поворот графика
fig.colorbar(surf, ax = [axes], pad = 0.05, location='left')
plt.title('График численного решения y(x, t) \n Nx=' +str(Nx) + ' Nt='
          +str(Nt)+ ' tau/h=' +str(np.abs(tau/h))+ ' e=' +str(e))
plt.show()



"9. Построение погрешности решения"
T,X = np.meshgrid(t,x)
fig = plt.figure()
axes = fig.add_subplot(111, projection='3d')

surf = axes.plot_surface(X, T, np.abs(U[:][:] - y[:][:]), rstride=4, cstride=4, cmap = cm.jet)
pylab.xlabel('$x$', size=15, labelpad = 5)
plt.xticks(rotation = 0)
pylab.ylabel('$t$', size=15, labelpad = 5)
plt.yticks(rotation = 0)

axes.set_zlabel('y_Error(x,t)', size = 15, labelpad = 10)



axes.view_init(20, 120) #поворот графика
plt.colorbar(surf, ax = [axes], pad = 0.05, location='left')
plt.title('График погрешности |U(x,t) - y(x, t)| \n Nx=' +str(Nx) + ' Nt='
          +str(Nt)+ ' tau/h=' +str(np.abs(tau/h))+ ' e=' +str(e))
plt.show()
print(np.abs(U[Nx-1][Nt-1] - y[Nx-1][Nt-1]))
print(np.sqrt(tau*tau+h*h))
```

```


"Построение зависимости прогрешности в узле от величины шагов сетки"
Nx_1 = 40
Nx_2 = 2500
Nx_N = 14

Nt_1 = 40
Nt_2 = 2500
Nt_N = 14

Nx = np.linspace(Nx_1,Nx_2,Nx_N)
Nt = np.linspace(Nt_1,Nt_2,Nt_N)
h = np.zeros((Nx_N))
tau = np.zeros((Nt_N))

for a in range(0,Nx_N):
    h[a] = (x2 - x1)/Nx[a]
for b in range(0,Nt_N):
    tau[b] = (t2 - t1)/Nt[b]

Ry = np.zeros((Nx_N,Nt_N))
rU = np.zeros((Nx_N,Nt_N))

for a in range(0,Nx_N):
    for b in range(0,Nt_N):
        Ex = np.linspace(x2,x1,int(Nx[a]))
        Et = np.linspace(t1,t2,int(Nt[b]))
        Ey = np.zeros((int(Nx[a]), int(Nt[b])))
        for n in range(0,int(Nx[a])): #Начальное условие
            Ey[n][0] = np.log(2/(1 - 2*Ex[n]))

        for k in range(0,int(Nt[b])): #Граничное условие
            Ey[0][k] = np.log(2/(np.exp(Et[k]) - 2*Et[k]*np.exp(-Et[k])))


        "Задание функций f и f'  "
        def Ef(n,k,a,b):
            return 1/tau[b]*(Ey[n+1][k+1] - Ey[n+1][k]) + 1/h[a]*(Ey[n+1][k+1] - Ey[n][k+1]) + np.exp(Et[k+1] + Ey[n+1][k+1]) - 1

        def Efd(n,k,a,b):
            return 1/tau[b] + 1/h[a] + np.exp(Et[k+1] + Ey[n+1][k+1])


        "Реализация метода Ньютона"
        for n in range(0,int(Nx[a])-1):
            for k in range(0,int(Nt[b])-1):
                Ey[n+1][k+1] = 0.5*(Ey[n+1][k]+Ey[n][k+1]) # Приближение для следующего шага
                while(np.abs(Ef(n,k,a,b) / Efd(n,k,a,b)) > e):
                    Ey[n+1][k+1] = Ey[n+1][k+1] - Ef(n,k,a,b) / Efd(n,k,a,b)

```

```

        "Аналитическое решение"
        eU = np.zeros((int(Nx[a]),int(Nt[b])))
        for n in range(0,int(Nx[a])):
            for k in range(0,int(Nt[b])):
                eU[n][k] = np.log(2/(np.exp(Et[k]) - 2*(Ex[n]+Et[k])*np.exp(-Et[k])))

        "Выбираем узел, близкий к самой далекой точке от ГУ и НУ"
        Ry[a][b] = Ey[int(Nx[a]) - 1][int(Nt[b]) - 1]
        rU[a][b] = eU[int(Nx[a]) - 1][int(Nt[b]) - 1]


"9. Построение ошибки узла"
NT,NX = np.meshgrid(Nt,Nx)
fig = plt.figure()
axes = fig.add_subplot(111, projection='3d')
surf = axes.plot_surface(NX, NT, np.abs(rU[:][:] - Ry[:][:]), rstride=1, cstride=1, cmap = cm.jet)
pylab.xlabel('$h$', size=15, labelpad = 5)
plt.xticks(rotation = 0)
pylab.ylabel('$tau$', size=15, labelpad = 5)
plt.yticks(rotation = 0)

axes.set_zlabel('|RU - Ry|', size = 15, labelpad = 10)
axes.view_init(20, 130) #поворот графика
plt.colorbar(surf, ax = [axes], pad = 0.05, location='left')
plt.title('График погрешности |RU(x,t) - Ry(x, t)| в узле \n (' +str(Nx_1) + ','
          +str(Nx_2)+','+str(Nx_N)+') ; (' +str(Nt_1)+ ',' +str(Nt_2)+ ','
          +str(Nt_N)+');  e=' +str(e))
plt.show()


TAU,H = np.meshgrid(tau,h)
plt.contourf(H, TAU, np.abs(rU[:][:] - Ry[:][:]), cmap = cm.jet)
plt.xlabel('$h$', size=15, labelpad = 5)
plt.xticks(rotation = 0)
plt.ylabel('$tau$', size=15, labelpad = 5)
plt.yticks(rotation = 0)

axes.set_zlabel('|RU - Ry|', size = 15, labelpad = 10)
plt.colorbar(pad = 0.12)
plt.title('График погрешности |RU(x,t) - Ry(x, t)| \n (' +str(Nx_1) + ','
          +str(Nx_2)+','+str(Nx_N)+') ; (' +str(Nt_1)+ ',' +str(Nt_2)+ ','
          +str(Nt_N)+');  e=' +str(e))
plt.show()
```

Список литературы

[1] Численные методы, Калиткин Н.Н., М.: Издательский центр «Академия», 2013 г.

[2] Курс лекций «ОСНОВЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ», Тихонов Н.А., Токмачев М.Г., Москва, 2012